Machine Translation (2020) 34:251–286
https://doi.org/10.1007/s10590-020-09253-x

Transfer learning and subword sampling for asymmetric-resource one-to-many neural translation

Stig-Arne Grönroos[1] · **Sami Virpioja**[2,3] · **Mikko Kurimo**[1]

Received: 25 February 2020 / Accepted: 27 November 2020 / Published online: 30 January 2021

Abstract

There are several approaches for improving neural machine translation for low-resource languages: monolingual data can be exploited via pretraining or data augmentation; parallel corpora on related language pairs can be used via parameter sharing or transfer learning in multilingual models; subword segmentation and regularization techniques can be applied to ensure high coverage of the vocabulary. We review these approaches in the context of an asymmetric-resource one-to-many translation task, in which the pair of target languages are related, with one being a very low-resource and the other a higher-resource language. We test various methods on three artificially restricted translation tasks—English to Estonian (low-resource) and Finnish (high-resource), English to Slovak and Czech, English to Danish and Swedish—and one real-world task, Norwegian to North Sámi and Finnish. The experiments show positive effects especially for scheduled multi-task learning, denoising autoencoder, and subword sampling.

Keywords Low-resource languages · Multilingual machine translation · Transfer learning · Multi-task learning · Denoising sequence autoencoder · Subword segmentation

✉ Stig-Arne Grönroos
stig-arne.gronroos@aalto.fi

Sami Virpioja
sami.virpioja@helsinki.fi

Mikko Kurimo
mikko.kurimo@aalto.fi

[1] Department of Signal Processing and Acoustics, Aalto University, Espoo, Finland

[2] Department of Digital Humanities, University of Helsinki, Helsinki, Finland

[3] Utopia Analytics, Helsinki, Finland

1 Introduction

Machine translation (MT) has become an important application for natural language processing (NLP), enabling increased access to the wealth of digital information collected on-line, and new business opportunities in multilingual markets. MT has made rapid advances following the adoption of deep neural networks in the last decade, with variants of the sequence-to-sequence (seq2seq, (Kalchbrenner and Blunsom 2013; Sutskever et al. 2014)) architecture currently holding the state of the art in neural machine translation (NMT). However, the recent success has not applied to all languages equally. Current state-of-the-art methods require very large amounts of data: Seq2seq methods have been shown to work well in large data scenarios, but are less effective for low-resource languages. The rapid digitalization of society has increased the availability of suitable parallel training corpora, but the growth has not distributed evenly across languages.

The amount of data needed to reach acceptable quality can also vary based on language characteristics. Rich, productive morphology leads to a combinatorial explosion in the number of word forms. Therefore, a larger corpus is required to reach the same coverage of word forms. Often the two challenges coincide, with morphologically complex languages that are also relatively low on resources.

Three distinct types of resources may be available for MT training: parallel data, monolingual data, and data in related languages. In the low-resource translation setting, it is primarily the parallel data that is scarce. Monolingual data is easier to acquire and typically more abundant. In addition, there may be related languages with much more abundant resources.

In this work, we consider machine translation *into* a low-resource morphologically rich language by means of *transfer learning* from a related high-resource target language, by exploiting available *monolingual corpora*, and by exploring the methods and parameters for *vocabulary construction*. Figure 1 illustrates an overview of the known techniques for low-resource multilingual NMT; most of them are considered in our experiments.

Our task is a *one-to-many* setting in multilingual neural machine translation (MNMT), as opposed to *many-to-one* and *many-to-many* settings (Luong 2016).

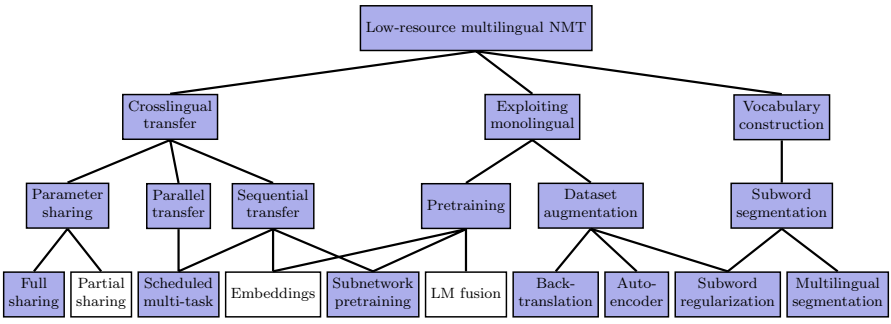

Fig. 1 Overview of techniques for improving low-resource multilingual NMT. Techniques highlighted with blue are used in this work (color figure online)

Table 1 Example from NMT system overfitted to the language modeling task

Estonian	Source	Laktoosi puhul see nii ju ongi!
English	Overfit translation	I've been thinking about it.
English	Reference	That's the case with lactose!

As we consider target languages that have different amounts of training resources available, we call this an *asymmetric-resource one-to-many* translation task. It has three major challenges:

Sparsity Translating into a low-resource is challenging, especially in the case of a morphologically rich language, due to a combination of small data and a large target vocabulary. The resulting data sparsity makes it difficult to estimate statistics for all but the most frequent items. Even though continuous-space representations allow neural methods to generalize well, they learn poorly from low-count events. Methods like subword segmentation (Virpioja et al. 2007; Sennrich et al. 2015) can reshape the frequency distribution of the basic units to reduce sparsity, and yield a more balanced class distribution in the generator. Suitable subwords are also beneficial for exploiting transfer from related high-resource languages (Grönroos et al. 2018), and from monolingual data.

Data imbalance In multilingual machine translation, it is very common to have an imbalance between the languages in the training data. The data can vary in quantity, quality and appropriateness of domain. Typically all three challenges affect the low-resource languages: when data is hard to come by, even noisy and out-of-domain data must be used. The data imbalance is typically addressed by oversampling the low-resource data. One way to choose the oversampling weights is using a temperature-based approach to interpolate between sampling from the true distribution and sampling uniformly (Arivazhagan et al. 2019). An alternative to oversampling the data is to adapt the gradient scale or learning rate individually for each task (Chen et al. 2018).

Task imbalance An NMT system is a conditional language model. The training signal for the language model is much stronger than for conditioning on the source. The conditioning requires training the natural language understanding encoder and the cross-lingually aligning attention mechanism, which are both difficult tasks. High fluency is a known property of NMT (Toral and Sánchez-Cartagena 2017; Koponen et al. 2019). When a vanilla NMT system is trained in a low-resource setting, the learning signal may be sufficient to train the language model, but insufficient for the conditioning (Östling and Tiedemann 2017). In this case, the MT system degenerates into a fancy language model, with the output resembling generated nonsense, with possibly high fluency but little relation to the source text. As an example, Table 1 shows an output from an Estonian–English translation system trained from parallel data of only 18k sentence pairs. Mueller et al. (2020) observe this language model overfitting phenomenon in a massively multilingual but low-resource setting using Bible translations as the corpus.

Given these challenges, our research questions include:

1. On cross-lingual transfer, is it better to use sequential (pretraining followed by fine-tuning) or parallel (all tasks at the same time) transfer, or something in between?
2. On exploiting monolingual data:
   (a) For which languages should one add monolingual auxiliary tasks? Is it useful to have a target-language autoencoder in addition to the back-translation strategy, where synthetic training data is generated by a target-to-source translation model?
   (b) What kind of noise models are most useful for the denoising sequence autoencoder task?
3. On vocabulary construction:
   (a) What is a suitable granularity of subword segmentation for the low-resource task?
   (b) Does it matter what data-driven segmentation method is used?
   (c) Does subword regularization (sampling different segmentations for the same word forms) help?
4. On available data and languages:
   (a) When data is very scarce, is it better to train a small model on the low-resource data, or a larger model using also the auxiliary data?
   (b) Is cross-lingual transfer more useful than transfer from monolingual tasks?
   (c) How does the amount of the data available for the low-resource language affect the translation quality?
   (d) How important is language relatedness for the cross-lingual transfer?

As methodological contributions for NMT, we formulate a scheduled multi-task learning technique for asymmetric-resource cross-lingual transfer, propose our recently introduced Morfessor EM+Prune method (Grönroos et al. 2020) for learning the subword vocabulary, and introduce a taboo sampling task for improving the modeling of segmentation ambiguity. We include experiments using three diverse language families, with Estonian, Slovak and Danish as simulated low-resource target languages. We also contribute a Norwegian bokmål to North Sámi translation system, the first NMT system for this target language, to the best of our knowledge.

In the next three sections, we will discuss the different techniques for cross-lingual transfer, exploiting monolingual data, and vocabulary construction. Then we will describe our experimental setup and discuss the results for four different groups of languages, and finally summarize our findings.

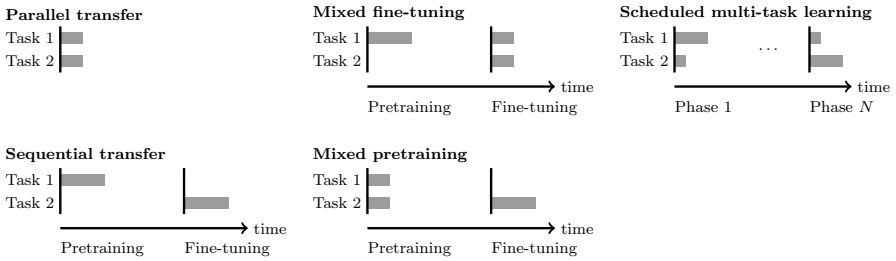

Fig. 2 Task mixing strategies for transfer learning

2 Cross-lingual transfer

Multilingual training allows exploiting cross-lingual transfer between related languages by training a single model to translate between multiple language pairs. This is a form of *multi-task learning* (Caruana 1998), in which each language pair in the training data can be seen as a separate learning task (Luong et al. 2015). The low-resource language is the main task, and at least one related high-resource language is used as an auxiliary task. The cardinality of the multilingual translation has an effect: cross-lingual transfer is easier in the many-to-one setting compared to one-to-many (Arivazhagan et al. 2019). For a general survey on multilingual translation, see Dabre et al. (2020).

2.1 Sequential and parallel transfer

In transfer learning, knowledge gained while learning one task is transferred to another. The tasks can either be trained sequentially or in parallel. Transfer is essential in *asymmetric-resource* settings, in which the amount of training examples for the target task very small, requiring the learner to rapidly generalize. *Sequential transfer* is a form of adaptation. In sequential transfer learning, the *pretraining* on a high-resource parent task is used to initialize and constrain the *fine-tuning* training on the low-resource child task. Zoph et al. (2016) apply sequential transfer learning to low-resource neural machine translation. Sequential transfer carries the risk of catastrophic forgetting (McCloskey and Cohen 1989; Goodfellow et al. 2014), in which the knowledge gained from the first task fades away completely. Some parameters can be frozen between the two training phases. This reduces the number of parameters trained from the small data, which may delay overfitting.

When training tasks in parallel, called *multi-task learning*, catastrophic forgetting does not occur. If the amount of data for different tasks is highly asymmetrical, careful tuning of the task mixture weights is critical to avoid overfitting on the small task. Sequential transfer does not require the same tuning, as convergence can be determined for each task separately.

It is also possible to combine sequential and parallel transfer. Figure 2 shows some possible ways of achieving this by mixing the tasks. One strategy—*mixed*

fine-tuning—involves first pretraining only on the large task, and then fine-tuning with a mixture of tasks. Chu et al. (2017) apply this strategy to domain adaptation. Kocmi (2019) try the inverse setting—*mixed pretraining*—pretraining on a mixture of tasks and fine-tuning only on the child task.

Kiperwasser and Ballesteros (2018) propose generalizing these strategies into *scheduled multi-task learning*, in which training examples from different tasks are selected according to a mixing distribution. The mixing distribution changes during training according to the task-mix schedule. They experiment with three schedules: constant, exponential and sigmoidal. We propose a new partwise constant task-mix schedule suitable for an asymmetric-resource setting with multiple auxiliary tasks. The task-mix schedule can have an arbitrary number of steps, any of which can be mixing multiple tasks. All the other strategies can be recovered by using particular schedules with scheduled multi-task learning.

2.2 Parameter sharing

In neural networks, multilingual models are implemented through parameter sharing. It is possible to share all neural network parameters, or select a subset for sharing allowing the remaining ones to be language-specific. Parameter sharing can be either hard or soft. In hard parameter sharing the exact same parameter matrix is used for several languages. In soft parameter sharing, each language has its own parameter matrix, but a dependence is constructed between the corresponding parameters for different languages.

The *target language token* Johnson et al. (2017) and *language embedding* (Conneau and Lample 2019) approaches use hard sharing of all parameters. In the former, the model architecture is the same as in a language-pair-specific model. The target language is indicated by a preprocessing step that prepends to the input a special target language token, e.g. ⟨TO_FI⟩ to indicate that the target language is Finnish. The approach can be scaled to more languages by increasing the capacity of the model, primarily by increasing the depth in layers (Arivazhagan et al. 2019). The latter can be described as a factored representation, with the language embedding factor marking the language of each word on the target side.

In contrast to full parameter sharing, it is also possible to divide the model parameters into shared and *language-specific subnetworks*, e.g. sharing all parameters of the encoder, while letting each target language have its own decoder. Parameter sharing can even be controlled on a more fine-grained level (Sachan and Neubig 2018). Shared attention (Firat et al. 2016) uses language-specific encoders and decoders with a shared attention, while language-specific attention (Blackwood et al. 2018) does the opposite by sharing only the feedforward sublayers of the decoder, while using language-specific parameters for the attention mechanisms.

The *contextual parameter generator* (Platanios et al. 2018) meta-learns a soft dependency between parameters for different tasks. It does this by using one neural network (the parameter generator) to generate from some contextual variables the weights of another network (the model). Gu et al. (2018) apply meta-learning to find initializations that can very rapidly adapt to a new low-resource source language.

3 Exploiting monolingual data

While parallel data is the primary type of data used for training MT models, methods for effectively exploiting the more abundant monolingual data can greatly increase the number of available examples to learn from. Use of monolingual data can be viewed as semi-supervised learning: both labeled (parallel) and unlabeled (monolingual) data are used. There are two main approaches to exploiting monolingual data in MT: transfer learning and dataset augmentation.

3.1 Transfer learning: monolingual pretraining

In monolingual pretraining, some of the parameters of the final translation model are pretrained on a task using monolingual data, possibly using a different loss than the one used during NMT training. There are several ways to use pretraining: Pretrain word (or subword) *embeddings* for the encoder, decoder, or both. Pretrain a separate *language model* for the target language, and combine it with the predictions of the translation model. Or, finally, pretrain an entire *subnetwork*—encoder or decoder—of the translation model.

3.1.1 Embeddings

Source and target embeddings can be pretrained on monolingual data from the source and target languages, respectively (Di Gangi and Federico 2017). Alternatively, joint cross-lingual embeddings can be trained on both (Artetxe et al. 2018). As the embeddings are trained for e.g. a generic contextual prediction task, this is a form of transfer learning. The pretrained embeddings can either be frozen or fine-tuned, by respectively omitting or including them as trainable parameters during NMT training. Thompson et al. (2018) investigate the effects of freezing various subnetwork parameters—including embeddings—on domain adaptation. In addition to using monolingual data, pretrained embeddings can contribute to cross-lingual transfer in the case of a shared multilingual embedding space (Artetxe et al. 2018). The shared embedding spaces are typically on a word level.

3.1.2 Language model fusion

The predictions of a strong language model can be combined with the predictions of the translation model, either using a separate rescoring step, or by combining the predictions during decoding, using *model fusion*. This approach is used in statistical machine translation, where one or more target language models are combined with a statistical translation model. The approach can also be applied in neural machine translation, through shallow fusion, deep fusion (Gulcehre et al. 2015), cold fusion (Sriram et al. 2017), or PostNorm (Stahlberg et al. 2018). As a neural machine translation system is already a conditional language model, it may be preferable to find a way to train the parameters of the NMT system using the monolingual data.

3.1.3 Subnetwork pretraining

In subnetwork pretraining, the intent is to pretrain entire network components—the encoder or the decoder—with knowledge about the structure of language. One way to achieve this using unlabeled data is to apply a language modeling loss during pretraining. The loss function can either be the traditional next token prediction, or a masked language model. Alternatively an autoencoder loss can be used.

Domhan and Hieber (2017) modify the NMT architecture by adding an auxiliary language model loss in the internal layers of the decoder, before attending to the source. This loss allows the first layers of the decoder to be trained on monolingual data. They find no benefit of adding the language model loss unless additional monolingual data is used. Adding monolingual data gives a benefit, but does not outperform back-translation. Ramachandran et al. (2017) pretrain the encoder and decoder with source and target language modeling tasks, respectively. To prevent overfitting, they use task-mix fine-tuning: the translation and language modeling objectives are trained jointly (with equally weighted tasks). Skorokhodov et al. (2018) use both pretraining (on both source and target side) and gated shallow fusion (on the target side) to transfer knowledge from pretrained language models. Some of the experiments are performed on low-resource data going down to 10 k sentence pairs.

3.2 Dataset augmentation

The easiest way to improve generalization is to train on more data. As natural training data is limited, a practical way to acquire more is to generate additional synthetic data for augmentation. The main benefit of dataset augmentation is as regularization to prevent overfitting to non-robust properties of small data.

Simple ways to generate synthetic data include using a single dummy token on the source side (Sennrich et al. 2016), and copying the target to source (Currey et al. 2017). The latter can be interpreted as a target-side autoencoder task without noise. The largest factor in determining the effectiveness of using synthetic data is how much the synthetic data deviates from the true data distribution. To avoid confusing the encoder with synthetic data from a different distribution than the natural data, it may be beneficial to use a special tag to identify the synthetic data (Caswell et al. 2019).

3.2.1 Back-translation

Synthetic data can be self-generated by the model being trained, or a related model. In machine translation, the best known example of synthetic data is *back-translation* (BT) (Sennrich et al. 2016). The process of back-translation begins with the training of a preliminary MT model in the reverse direction, from target to source. The target language monolingual data is translated using this model, producing a synthetic, pseudo-parallel data set with the potentially noisy MT output on the source side. Because the quality of the translation system used for the back-translation affects

the noisiness of the synthetic data, the procedure can be improved by iterating with alternating translation direction (Lample et al. 2018b). Edunov et al. (2018) propose adding noise to the back-translation output. The benefit of noisy back-translation is further analyzed by Graça et al. (2019), who recommend turning off label smoothing in the reverse model when combined with sampling decoding. As a related strategy, Karakanta et al. (2018) convert parallel data from a high-resource language pair into synthetic data for a related low-resource pair using transliteration. Zhang and Zong (2016) exploit monolingual data in two ways: through self-learning by "forward-translating" the monolingual source data to create synthetic parallel data, and by applying a reordering auxiliary task: the input is the natural source text, while the output is the source text reordered using rules to match the target word order.

3.2.2 Subword regularization

Subword regularization is a technique proposed by Kudo (2018) for applying a probabilistic subword segmentation model to generate more variability in the input text. Each time a word token is used during training, a new segmentation is sampled for it. It can be seen as treating the subword segmentation as a latent variable. While marginalizing over the latent variable exactly is intractable, the subword regularization procedure approximates it through sampling.

3.2.3 Denoising sequence autoencoder

Back-translation is a slow method due to the additional training of the reverse translation model. A computationally cheaper way to turn monolingual data into synthetic parallel data is to use a denoising autoencoder as an auxiliary task. Target language text, corrupted by a noise model, is fed in as a pseudo-source. Different noise models can be used, e.g. applying reordering, deletions, or substitutions to the input tokens. The desired reconstruction output is the original noise-free target language text.

An autoencoder Bourlard and Kamp (1988) is a neural network that is trained to copy its input to its output. It applies an encoder mapping from input to a hidden representation, i.e. code $\boldsymbol{h} = f(\boldsymbol{x})$, and decoder mapping from code to a reconstruction of the input $\hat{\boldsymbol{x}} = g(\boldsymbol{h})$. To force the autoencoder to extract patterns in the data instead of finding the trivial identity function $\hat{\boldsymbol{x}} = \mathbf{1}(\mathbf{1}(\boldsymbol{x}))$, the capacity of the code must be restricted somehow. In the undercomplete autoencoder, the restriction is in the form of a bottleneck layer with small dimension. For example, in the original sequence autoencoder (Dai and Le 2015), the entire sequence is compressed into a single vector.

In a modern sequence-to-sequence architecture, the attention mechanism ensures a very large bandwidth between encoder and decoder. When used as an autoencoder, the network is thus highly overcomplete. In this case, the capacity of the code has to be controlled by regularization. Robustness to noise is used as the regularizer in the *denoising autoencoder* (Vincent et al. 2008). Instead of feeding in the clean example $\boldsymbol{x}$, a corrupted copy of the input is sampled from a noise model $\mathrm{C}(\tilde{\boldsymbol{x}} \mid \boldsymbol{x})$. The denoising autoencoder must then learn to reverse the corruption to reconstruct

the clean example. The use of noise as regularization is a successful technique used e.g. in Dropout (Srivastava et al. 2014), label smoothing (Szegedy et al. 2016), and SwitchOut (Wang et al. 2018). Also multi-task learning acts as regularization by claiming some of the capacity of the model. Belinkov and Bisk (2017) apply both natural and synthetic noises for NMT evaluation, finding that standard character-based NMT models are not robust to these types of noise.

There are multiple ways of adding the autoencoder loss to the NMT training. The simplest one treats the autoencoder task as if it was another language pair for multilingual training, and involves no changes to the architecture. When using this type of autoencoder task on target language sentences, the task cardinality changes into a many-to-one problem: the model must simultaneously learn a mapping from source to target and from corrupted target to clean target. In both tasks the target language is the same. As the decoder is a conditional language model, this task strengthens the modeling of the target language. When using source language sentences, the model must simultaneously learn a one-to-many mapping from source to target and from corrupted source to clean source. Thus the decoder must learn to output both languages. The task may strengthen the encoder, by increasing its robustness to noise, and by preventing the encoding from becoming too specific to the target language. Luong et al. (2015) and Luong (2016) experiment with various auxiliary tasks, including this type of autoencoder setup. They see a benefit of using the autoencoder task, as long as it has a low enough weight in the task mix. This setup is used also in our experiments.

There are also more complex NMT autoencoder setups. In *dual learning*, the autoencoder is built from source-to-target and target-to-source translation models. He et al. (2016) combine source-to-target and target-to-source translations in a closed loop which can be trained jointly, using two additional language modeling tasks (for source and target respectively), and reinforcement learning with policy gradient. Cheng et al. (2016) use a dual learning setup to exploit monolingual corpora in both source and target languages. Their loss consists of four parts: translation likelihoods in both directions, source autoencoder, and target autoencoder. Tu et al. (2017) simplify the dual learning setup into an encoder–decoder–reconstructor network. The reconstructor attends to the final hidden states of the decoder and thus does not need a separate encoder. Their aim is to improve adequacy by penalizing undertranslation: the reconstructor is not able to generate any parts of the sentence omitted by the decoder.

3.2.4 Noise models for text

To apply a denoising autoencoder to text, a suitable noise model for text is needed. In domains such as image and speech, there are very intuitive noises, including rotating, scaling, and mirroring for images; and reverberation, time-scale stretching, and pitch shifting for speech. As text is a sequence of discrete symbols, where even a small change can have a drastic effect on meaning, suitable noise models are less intuitive. It is not feasible to guarantee the noise does not change the correct translation of the input.

Local reordering. Lample et al. (2018a) perform a local reordering operation σ that they call *slightly shuffling* the sentence. The reordering is achieved by adding to the index i of each token a random offset drawn from the uniform distribution from 0 to a maximum distance k. The tokens are then sorted according to the offset indices. This maintains the condition $\forall i \in \{1, n\}, |\sigma(i) - i| \leq k$.

Token deletion Randomly dropping tokens is perhaps the most commonly used noise. It is the central idea in *word dropout* (Iyyer et al. 2015). In word dropout, each token is dropped according to a Bernoulli distribution parameterized by a tunable dropout probability.

Token insertion Randomly selected tokens can also be inserted into the sentence. The tokens can be sampled from the entire vocabulary, or from a particular class of tokens. E.g. Vaibhav et al. (2019) insert three classes of tokens: stop words, expletives, and emoticons.

Token substitution SwitchOut (Wang et al. 2018) applies random substitutions to tokens both in the source and the target sentence. One benefit of SwitchOut is that it can easily and efficiently be applied late in the data processing pipeline, even to a numericalized and padded minibatch. Any noises that affect the length of the sequence are best applied before numericalization.

Token masking Masked language models (Devlin et al. 2019; Song et al. 2019; Lewis et al. 2019; Joshi et al. 2020) apply a special case of token substitution, randomly substituting tokens or spans of tokens with a mask symbol.

Word boundary noise In a special case of token substitution, the substituted token is selected deterministically as the token with a word boundary marker either added or removed. E.g. *"kielinen"* would be substituted by *"_kielinen"* and vice versa. This might improve robustness to compounding mistakes such as *"*suomen kielinen"* (Finnish speaker).

Taboo sampling In addition to training the translation model, the idea of subword regularization (Kudo 2018) can be used in the autoencoder. Here, we propose taboo sampling as a special form of subword regularization for monolingual data. The method takes a single word sequence as input, and outputs two different segmentations for it. The two segmentations consist of different subwords, whenever possible. Only single character morphs are allowed to be reused on the other side, to avoid failure if no alternative exists. e.g. *"unreasonable"* could be segmented into *"un"*$+\!\!+$*"reasonable"* on the source side and *"unreason"*$+\!\!+$*"able"* on the target side. When converted into numerical indices into the lexicon, these two representations are completely different. The task aims to teach the model to associate with each other the multiple ambiguous ways to segment a word, by using a segmentation-invariant internal representation.

For each word, one segmentation is sampled in the usual way, after which another segmentation is sampled using taboo sampling. During taboo sampling, all multi-character subwords used in the first segmentation have their emission probability temporarily set to zero. To avoid introducing a bias from having all the taboo sampled segmentations on the same side, the sides are mixed by uniformly sampling a binary mask of the same length as the sentence from the set of masks with half the bits 1. All words for which the mask bit is set have the source and target segmentations swapped.

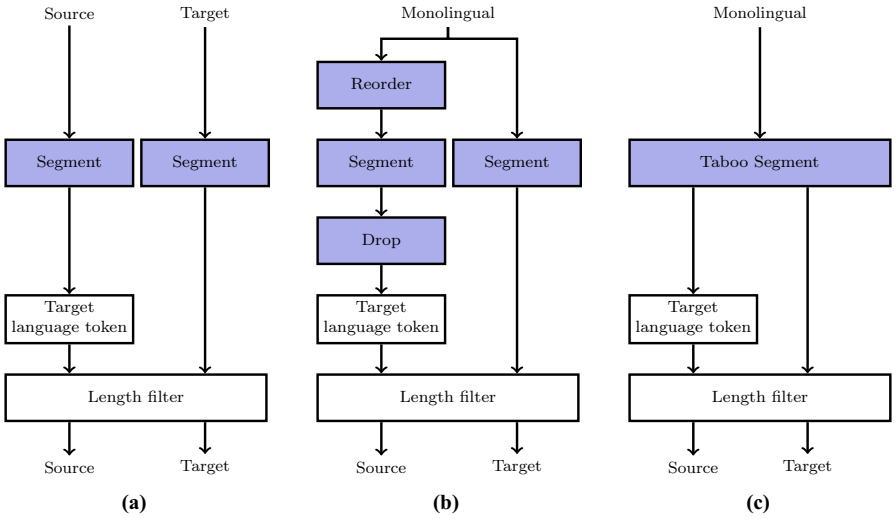

Fig. 3 Transformations applied to data at training time. Steps with blue background are part of the stochastic noise model. Steps with white background are the deterministic target language token prefixing and length filtering. Length filtering must be applied after segmentation, which may make the sequence longer (color figure online)

Proposed noise model combinations Our proposed noise model combination is depicted in Fig. 3. It consists of three pipelines: The pipeline for parallel data (a) consists of only sampling segmentation. The primary pipeline for monolingual data (b) is a concatenation of multiple noise models: local reordering, segmentation, and token deletion. A secondary pipeline for monolingual data (c) uses taboo segmentation. In all cases the output consists of a pair of source and target sequences.

Observe that the transformations are applied in the data loader at training time, not as an off-line preprocessing stage. This allows the noise to be resampled for each parameter update, which is critical when training continues for multiple epochs of a small dataset. As a minor downside, the NMT software needs to be modified to accommodate the heavier data loader, while preprocessing generally requires no modifications to the software.

4 Vocabulary construction

The vocabulary or lexicon of a translation model is the set of basic units or building blocks the text is decomposed into. In phrase-based machine translation, the standard approach is to use a word lexicon. Segmentation into subword units has been proposed mostly for morphologically rich languages, for which a word lexicon leads to very high out-of-vocabulary (OOV) rates (Lee 2004; Oflazer and El-Kahlout 2007; Virpioja et al. 2007), and character segmentation for closely related languages (Tiedemann 2009). However, the change of paradigm to neural machine translation has changed also the practice in vocabulary construction: With the exception of

unsupervised translation based on pretrained word embeddings (Artetxe et al. 2018; Yang et al. 2018), the standard approach for models is segmentation into subword units (Sennrich et al. 2015). Some studies aim even to the other extreme, characters (Chung et al. 2016; Costa-jussà and Fonollosa 2016) or bytes (Costa-jussà et al. 2017).

A specific task in subword segmentation is the morphological surface segmentation. There the aim is to split words into morphs, the surface forms of meaning-bearing sub-word units, morphemes. The concatenation of the morphs is the word, for example

$$capability \mapsto cap +\!\!+ abil +\!\!+ ity.$$

Unsupervised morphological segmentation, dating back to Harris (1955), was an active research topic in 2000s and early 2010s (Goldsmith 2001; Creutz and Lagus 2007; Hammarström and Borin 2011), and the methods have been evaluated in various NLP applications (Kurimo et al. 2010; Virpioja et al. 2011). However, in applications based on neural network models, such as NMT, the correspondence of the subwords to linguistic morphemes is not of high importance, as the encoders are able to determine the meaning of the units in context. Therefore the subword segmentation is typically tuned using other criteria, such as the size of subword lexicon or the frequency distribution of the units. Desirable characteristics for a vocabulary to be used in multilingual NMT include:

1. **high coverage** of the training data, without imbalance between languages,
2. **a tractable size** for training, and
3. the right **level of granularity** for cross-lingual transfer.

Without a high coverage, some parts of the training data are impossible to represent using the vocabulary. The unrepresentable parts may be replaced with a special "unknown" token. If the proportion of unknown tokens increases, translation quality deteriorates. In a multilingual setting, a common approach is to use a shared subword vocabulary between the multiple source or target languages. In this case, training the segmentation model with a balanced data distribution is important to provide high coverage also for the less resourced languages.

Vocabulary size affects both the memory complexity via the number of network parameters and the computational cost via the length of the sequences and the size of the softmax layer. When using large vocabularies, e.g. words, the sequences are short, but vocabularies may grow intractably large, particularly for morphologically complex languages. When using small vocabularies, e.g. characters, memory requirements are low, but long sequences make training slow, particularly for recurrent networks.

The granularity of the segmentation affects both coverage and size of the lexicon: finer granularity typically means better coverage and smaller lexicon size. However, within the reasonable limits set by the coverage and size, it is much harder to determine the best possible level of granularity. Recent research (Cherry et al.

2018; Kreutzer and Sokolov 2018; Arivazhagan et al. 2019) indicates that smaller subwords are particularly useful for cross-lingual transfer to low-resource languages in supervised settings. Exploiting similarity of related languages by increasing the consistency of the segmentation between similar words of the source and target language can also be useful (Grönroos et al. 2018). In unsupervised NMT (Artetxe et al. 2018), cross-lingual transfer requires basic units to be aligned between languages without use of parallel data. When starting with pretrained embeddings, longer units are typically used, as they carry more meaning than short units. It is therefore an open question how the optimal segmentation granularity varies with the amount of resources available.

Next, we consider different data-driven segmentation methods proposed for machine translation. This study focuses on segmentation methods applying a *unigram language model*. In the unigram language model, it is assumed that the morphs in a word occur independently of each other. Given the parameters $\boldsymbol{\theta}$ of the segmentation model, the probability of a sequence of morphs $\boldsymbol{s}$ decomposes into the product of the probabilities of the morphs m of which it consists:

$$P_{\boldsymbol{\theta}}(\boldsymbol{s}) = \prod_{i=1}^{N} P_{\boldsymbol{\theta}}(m_i) \tag{1}$$

4.1 Byte pair encoding

The most popular method for subword segmentation in the field of NMT is currently the Byte Pair Encoding (BPE) compression algorithm (Gage 1994). The BPE algorithm iteratively replaces the most frequent pair of bytes in the data with a single unused byte. In NMT, the algorithm is typically used on characters, and the merging of characters is stopped when the given vocabulary size is reached (Sennrich et al. 2015). While BPE is not a probabilistic model, the coding resembles unigram language models in that every subword m_i is encoded individually. As a bottom-up algorithm, BPE is reasonable to use in multilingual settings just by concatenating the corpora before training; this approach is called *joint* segmentation (Sennrich et al. 2015). If the data is balanced over the languages, the frequent words will be constructed in the early steps of the algorithm for all languages.

4.2 SentencePiece

SentencePiece (Kudo 2018; Kudo and Richardson 2018) is another segmentation method proposed especially for NMT. In contrast to BPE, it defines a proper statistical model for the unigram model in Eq. 1, and tries to find the model parameters that maximize likelihood of the data given a constraint on the vocabulary size.

For training the model, SentencePiece applies the Expectation Maximization (EM) algorithm (Dempster et al. 1977). The EM algorithm only updates the expected frequencies of the current units; it is not able to add or remove subwords from the vocabulary. Thus to use EM for the segmentation problem, two other things

are needed: a *seed lexicon* and a *pruning phase*. The seed lexicon initializes the vocabulary with useful candidate units, and pruning phase removes the least probable units from the model. Prior to SentencePiece, a similar approach has been proposed by Varjokallio et al. (2013) for application in automatic speech recognition.

In SentencePiece, the seed lexicon is constructed from the most frequent substrings in the training data. After initializing the seed lexicon, SentencePiece alternates between the EM phase and the pruning phases until the desired vocabulary size is reached. In the pruning phase, the subwords are sorted by the reduction in the likelihood function if the subword was removed. A certain proportion (e.g. 25%) of the multi-character subwords are pruned at a time, followed by the next EM phase.

4.3 Morfessor EM+Prune

Morfessor is a family of generative models for unsupervised morphology induction (Creutz and Lagus 2007). Here, consider the Morfessor Baseline method (Creutz and Lagus 2002; Virpioja et al. 2013) and its recent Morfessor EM+Prune variant (Grönroos et al. 2020).

4.3.1 Model and cost function

Morfessor Baseline is applies the unigram language model (Eq. 1). In contrast to SentencePiece, Morfessor finds a point estimate for the model parameters $\hat{\boldsymbol{\theta}}$ using Maximum a Posteriori (MAP) estimation. The MAP estimate yields a two-part cost function, consisting of a prior (the lexicon cost) and likelihood (the corpus cost). The Morfessor prior, inspired by the Minimum Description Length (MDL) principle (Rissanen 1989), favors lexicons containing fewer, shorter morphs.

For tuning the model, Kohonen et al. (2010) propose weighting the likelihood with a hyper-parameter α:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \{ -\log \overbrace{P(\boldsymbol{\theta})}^{\text{prior}} - \alpha \log \overbrace{P(\boldsymbol{D} \mid \boldsymbol{\theta})}^{\text{likelihood}} \} \tag{2}$$

This parameter controls the granularity of the segmentation. High values increase the weight of each emitted morph in the corpus (less segmentation), and low values give a relatively larger weight to a small lexicon (more segmentation).

Similar to SentencePiece, Morfessor can be used in subword regularization (Kudo 2018). Alternative segmentations can be sampled from the full data distribution using the forward-filtering backward-sampling algorithm (Scott 2002) or approximatively from an n-best list.

4.3.2 Training algorithm

The original training algorithm of the Morfessor Baseline method, described in more detail by Creutz and Lagus (2005) and Virpioja et al. (2013), is a local greedy search. The lexicon is initialized by whole words, and the segmentation proceeds

recursively top-down, finding an optimal segmentation into two parts for the current word or subword unit. Our preliminary studies have indicated that this algorithm does not find as good local optima as the EM algorithm especially for the small lexicons useful in NMT. As a solution, we have developed a new variant of the method called Morfessor EM+Prune (Grönroos et al. 2020).[1] It supports the MAP estimation and MDL-based prior of the Baseline model, but implements a new training algorithm based on the EM algorithm and lexicon pruning inspired by SentencePiece.

The training algorithm starts with a seed lexicon and alternates the EM and lexicon pruning steps similarly to SentencePiece. The prior of the Morfessor model must be slightly modified for use with the EM algorithm, but the standard prior is used during pruning. While SentencePiece aims for a predetermined lexicon size, in Morfessor, the final lexicon size is controlled by the hyper-parameter α (Eq. 2). To reach a subword lexicon of a predetermined size while using the prior, Morfessor EM+Prune implements an automatic tuning procedure. When the estimated change in prior and likelihood are computed separately for each subword, the value of α that gives exactly the desired size of lexicon after the pruning can be calculated.

In earlier work (Grönroos et al. 2020), we have shown that the EM+Prune algorithm reduces search error during training, resulting in models with lower costs for the optimization criterion. Moreover, lower costs lead to improved accuracy when segmentation output is compared to linguistic morphological segmentation. In the present study, we test it for the first time in NMT.

5 Experiments

In the experiments, we study how to best exploit the additional monolingual and cross-lingual resources for improving machine translation into low-resource morphologically rich languages. We compare various methods for three major aspects affecting the translation quality: using cross-lingual transfer, exploiting monolingual data and applying subword segmentation. The main focus lies on a noise model incorporating the subword segmentation.

We target a one-to-many multilingual setting with related, morphologically rich languages on the target side. The related languages include both high- and low-resource languages. This setting provides a good opportunity for cross-lingual learning, as the amount of data is highly asymmetric. Our aim is not to achieve an interlingual representation, so allowing the encoder to specialize for target languages is acceptable if it improves performance.

[1] Software available at https://github.com/Waino/morfessor-emprune.

Table 2 Parallel corpora

		Europarl	OpenSubtitles	Other parallel
ENG	CZE			CzEng
ENG	SLO	✓	✓	
ENG	FIN	✓	✓	Rapid2016, Paracrawl
ENG	EST	✓	✓	
ENG	SWE	✓	✓	
ENG	DAN	✓	✓	
NOB	FIN		✓	
NOB	SME			UiT freecorpus

5.1 Data sets

We perform experiments on four translation tasks, each consisting of a language triple: source language (SRC), high-resource target language (HRL) and low-resource target language (LRL). We only show SRC-LRL translation results, as the goal is to improve this particular translation direction.

The four tasks (LRL in boldface) are:

1. English (ENG) to Finnish (FIN) and **Estonian** (EST),
2. English to Czech (CZE) and **Slovak** (SLO),
3. English to Swedish (SWE) and **Danish** (DAN),
4. Norwegian bokmål (NOB) to Finnish (FIN) and **North Sámi** (SME).

In each task the two target languages are related. The target languages belong to three different language families: Germanic, Balto-Slavic and Uralic. All target languages are morphologically complex.

We use as parallel corpora Europarl (Koehn 2005), and OpenSubtitles v2018 (Lison and Tiedemann 2016), when available. In addition, we use the eu, news, and subtitle domains of CzEng v1.7 (Bojar et al. 2016), and the UiT freecorpus.[2] The corpora used for each language pair are shown in Table 2. The domains for the training data are parliamentary debate, movie subtitles, news and web, with the exception of North Sámi which contains a mix of many domains.

Our main source of monolingual data is WMT news text.[3] In addition, we use the following monolingual corpora: skTenTen[4] and Categorized News Corpus[5] for Slovak, Riksdagens protokoll[6] for Swedish, News 2012[7] for Danish, Aviskorpus[8] for Norwegian, and Wikipedia[9] for North Sámi.

[2] https://victorio.uit.no/freecorpus/.
[3] http://www.statmt.org/wmt18/translation-task.html.
[4] http://hdl.handle.net/11858/00-097C-0000-0001-CCDB-0.
[5] Technical University of Kosice, 2014
[6] https://spraakbanken.gu.se/eng/resource/rd-prot.
[7] http://hdl.handle.net/11022/0000-0000-2238-B.
[8] https://www.nb.no/sprakbanken/show?serial=oai%3Anb.no%3Asbr-4&lang=en.
[9] sewiki-20191201 dump.

Table 3 Data set sizes after cleaning

SRC	HRL	LRL	Parallel			Monolingual		
			SRC-HRL (M)	SRC-LRL	BT	SRC	HRL	LRL
ENG	CZE	SLO	24.7	(18k)	1M	44.3M	13.6M	27.8M
ENG	FIN	EST	19.4	(18k)	1M	44.3M	6.3M	3.6M
ENG	SWE	DAN	11.5	(18k)	750k	44.3M	10.7M	950k
NOB	FIN	SME	4.9	152k	150k	40.1M	6.3M	181k

Table 4 Specifications for the NMT system

Encoder	8 Transformer layers	Label smoothing	0.1
Decoder	8 Transformer layers	Precision	16-bit floating point
Hidden size	1024	Minibatch size	9200 subword tokens
Filter size	4096	Gradient accumulation	4 minibatches
Attention heads	16	Effective minibatch size	36800 subword tokens
Adam beta2	0.997	Training time	100k steps
Warmup	Noam, 16k steps	Beam size	8
Dropout weight	0.1	Heuristic penalties	None

For each of the low-resource languages, we select a subset of 18k sentence pairs. For ENG-EST, we also perform an experiment where the low-resource subset is repeatedly subsampled down to 3k sentence pairs. To avoid introducing a domain imbalance in the sampled subset, the pairs are sampled such that an equal number of sentences are selected uniformly at random from each cleaned corpus. The training data sizes after cleaning and subsampling are shown in Table 3.

As test sets we use the WMT newstest2018 (Bojar et al. 2018) for ENG-EST, the WMT test2011 extended to Slovak by Galuščáková and Bojar (2012) for ENG-SLO. For ENG-DAN we use 2k sentence pairs sampled from the JRC-Acquis corpus (Steinberger et al. 2006). For NOB-SME we use the Apertium story "Where is James?", a 48-sentence text with simple language, used as an initial development set for Apertium rule based MT systems (Forcada et al. 2011).

5.2 Evaluation measures

When selecting the evaluation measures, the morphologically rich target languages must be taken into account. Therefore, we use Character-F_1 (Popović 2015) in addition to BLEU[10] (Papineni et al. 2002). To evaluate the performance of systems on rare words, we use word unigram F_1 score computed over words occurring less than 5 times in the parallel training data (Sennrich et al. 2015).

[10] mteval-v13a.pl

5.3 Training details

We use the Transformer NMT architecture (Vaswani et al. 2017). Model hyperparameters are shown in Table 4. Training takes approximatively 96 h on a single V100 GPU, with the data loader in a separate process. When using scheduled multi-task learning, the mixing distribution is changed after 40k steps. In all experiments, we apply full parameter sharing using a target language token. We tune our models towards the best product of the three evaluation measures (charF_1, BLEU, rare word F_1) on a development set.

Back-translation was performed with essentially the same system, but with sources and targets swapped to achieve a many-to-one configuration. We mark the back-translation data as synthetic using a special token.

When using subword regularization or denoising autoencoder, the training data is not simply loaded from disk, but new random segmentations and noises are sampled each time a training example is used. To alleviate slowdown, we moved the data-loader and preprocessing pipeline into a separate process, which communicates the numericalized and padded minibatches to the training process via a multiprocessing queue. Our data loader is implemented as a fork of OpenNMT-py[11] (Klein et al. 2017).

With multilingual training, autoencoders and back-translation, our setting involves a large number of different tasks. The tasks can be divided by language (HRL, LRL) and by type (translation, autoencoder). Nearly all runs, with the exception of our vanilla baseline, use a mix of tasks in some or all phases.

5.4 Results

In this section, we present the results of ten experiments, each exploring a separate aspect of asymmetric-resource one-to-many NMT. We have detailed results for English–Estonian, and verify the central findings on two additional language triples. Finally, we present some results on the actual low-resource pair Norwegian–North Sámi.

Unless otherwise stated, the compared models are trained using joint Morfessor EM+Prune segmentation with 16k subword vocabulary, cross-lingual scheduled multi-task learning, autoencoder with full noise model, and subword regularization for the translation task. Our initial results are using autoencoder tasks for all three languages (SRC + HRL + LRL). Later some of the results were rerun with the better SRC + LRL configuration, which omits the high-resource target language autoencoder.

[11] Software available at https://github.com/Waino/OpenNMT-py/tree/dynamicdata. Later, the dataloader of OpenNMT-py version 2.0 was redesigned to incorporate our proposals.

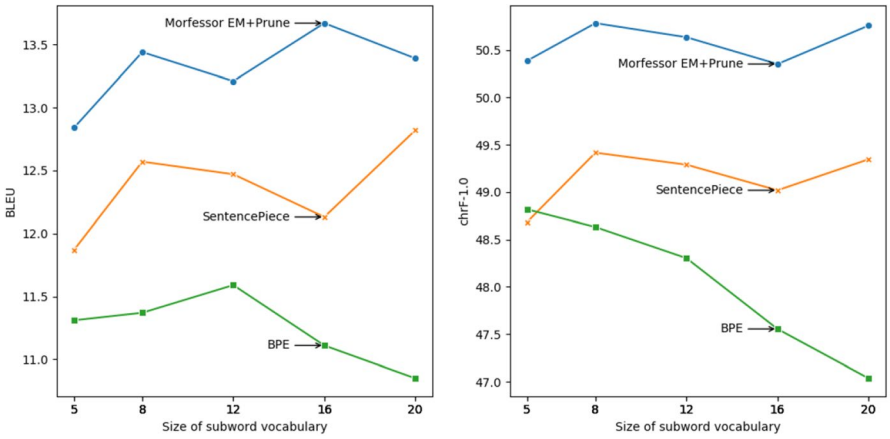

Fig. 4 Varying the subword vocabulary. Multilingual models, with SRC + HRL + LRL autoencoder and full noise model, except for BPE which are multilingual models without autoencoder or noise. Results on English → Estonian newsdev2018

5.4.1 Subword segmentation

For subword segmentation, we compare Morfessor EM+Prune to SentencePiece on various vocabulary sizes. The results are shown in Fig. 4. There is no clear optimal vocabulary size: in particular for the Character F_1 measure the performance remains nearly constant. On the test set, Morfessor EM+Prune is +0.6 BLEU better than SentencePiece. The difference is smaller than the +1.48 BLEU difference on the development set, but consistent. The difference between Morfessor EM+Prune and SentencePiece is similar for the ENG-DAN and ENG-SLO translation directions. In preliminary experiments BPE gave 0.65 BLEU worse results than EM+Prune already without subword regularization. We decided against further experiments using BPE, as it is incompatible with subword regularization.

5.4.2 Cross-lingual transfer

Table 5 shows the effect of multilingual training, with and without the autoencoder task. The cross-lingual transfer from the high-resource language yields the largest single improvement in our experiments. The multilingual model without autoencoder performs between + 10.26 and + 12.7 BLEU better than the vanilla model using only LRL parallel data. Adding an autoencoder loss results in a smaller gain, between + 4.97 and + 5.55 BLEU. The gains are partly cumulative for an additional gain of +0.05 to +1.14 BLEU.

The results for the vanilla model use a smaller configuration, with 4 encoder and 4 decoder layers, and batch size reduced to 2048. For the vanilla model the small network performed better than the large one, but when adding either multilingual training or autoencoder, the large network is superior.

Table 5 Results for cross-lingual transfer

Method	ML	BT	Autoencoder			ENG–EST			ENG–DAN			ENG–SLO		
			SRC	HRL	LRL	chrF1	BLEU	Rare	chrF1	BLEU	Rare	chrF1	BLEU	Rare
Both	✓		✓		✓	51.71	14.04	34.79	50.06	13.92	54.58	50.19	14.02	69.94
Only ML	✓					50.09	12.90	33.20	49.57	13.13	54.21	49.83	13.97	68.79
Only AE			✓		✓	42.65	8.19	21.59	42.26	7.60	44.48	38.97	6.25	62.51
Neither (vanilla)						29.46	2.64	6.22	31.95	2.63	30.40	23.76	1.27	36.80

ML multilingual, *BT* back-translation, *AE* autoencoder

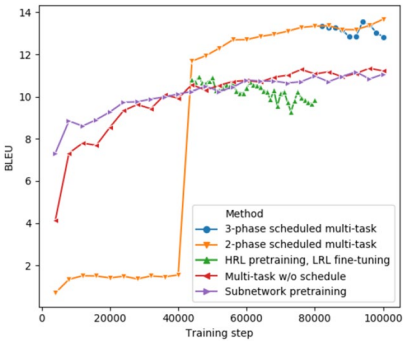

Fig. 5 Learning curves on LRL English → Estonian development set. Multilingual models, with SRC + HRL + LRL autoencoder and full noise model. Note that up to 40 k training steps, the model using scheduled multi-task learning has not seen any LRL data

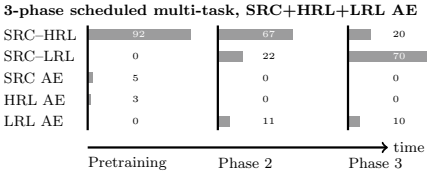

Fig. 6 The task mix schedule used in the 3-phase scheduled multi-task learning experiment. The 2-phase schedule is the same, except it omits the third phase, continuing the second phase until the end of training

5.4.3 Scheduled multi-task learning

Figure 5 shows the learning curves on the development set and Table 6 the evaluations on the test set for different configurations of transfer learning.

Multi-task without schedule is trained with a constant task mixing distribution. The result marked *HRL pretraining, LRL fine-tuning* uses a mix of HRL translation and autoencoder tasks for pretraining, and only a single task—LRL translation—for fine-tuning, and is thus fully sequential in terms of languages. It quickly overfits in the fine-tuning phase.

The models using scheduled multi-task learning combine sequential and parallel transfer. In *2-phase scheduled multi-task*, LRL tasks are not used in the pretraining phase, but a mix of tasks is used for fine-tuning. It gives a benefit of + 2.4 BLEU compared to the model fine-tuning on only LRL tasks, and +1.77 BLEU compared to training with a constant mixing distribution. The *3-phase scheduled multi-task* adds a third phase training mostly on LRL tasks. A small proportion of HRL translation is included to delay overfitting. The model again overfits in the final phase, but does reach a higher score before doing so. The 3-phase task mixing schedule is shown in Fig. 6.

Torrey and Shavlik (2009) describe three ways in which transfer learning can benefit training: (1) higher performance at the very beginning of learning, (2) steeper learning curve, and (3) higher asymptotic performance. When pretraining the encoder and decoder on source and target autoencoder tasks respectively, we see the first of these, but not the other two: for ENG–EST NMT training at first improves faster than with random initialization, but converges to a worse final model. As the

Table 6 Results for scheduled multi-task learning

Method	ML	BT	Autoencoder			ENG–EST			ENG–DAN			ENG–SLO		
			SRC	HRL	LRL	chrF1	BLEU	Rare	chrF1	BLEU	Rare	chrF1	BLEU	Rare
3-phase scheduled multi-task	✓		✓	✓	✓	51.71	13.94	33.96	50.1	13.7	54.6	50.1	14.1	69.7
2-phase scheduled multi-task	✓		✓	✓	✓	51.42	13.75	33.83	49.8	13.5	55.3	50.2	14.0	69.7
Multi-task w/o schedule	✓		✓	✓	✓	48.62	11.98	29.16	48.0	12.2	52.5	48.3	12.6	68.8
HRL pretraining, LRL fine-tuning	✓		✓	✓	✓	48.15	11.35	29.88	47.8	11.6	49.9	47.0	11.4	66.3
Subnetwork pretraining	✓		✓	✓	✓	47.74	11.17	26.93						

Table 7 Results with subword regularization (SWR)

Method	ML	BT	Autoencoder			ENG–EST			ENG–DAN			ENG–SLO		
			SRC	HRL	LRL	chrF1	BLEU	Rare	chrF1	BLEU	Rare	chrF1	BLEU	Rare
SWR	✓					50.09	12.90	33.20	49.57	13.13	54.21	49.83	13.97	68.79
no SWR	✓					49.77	12.57	31.14	49.27	13.05	53.66	49.27	13.42	69.07

Table 8 Ablation results for noise model

Method	ML	BT	Autoencoder			ENG–EST		
			SRC	HRL	LRL	chrF-1.0	BLEU	Rare
+ Word boundary noise	✓		✓	✓	✓	51.56	13.95	33.20
+ Taboo sampling	✓		✓	✓	✓	51.23	13.84	33.81
No drop	✓		✓	✓	✓	51.48	13.79	33.89
Full noise	✓		✓	✓	✓	51.42	13.75	33.83
+ Insertion	✓		✓	✓	✓	50.88	13.74	33.51
Only switchout	✓		✓	✓	✓	50.78	13.49	32.21
No SWR	✓		✓	✓	✓	50.71	13.46	32.18
Only SWR	✓		✓	✓	✓	50.96	13.43	32.85
No reorder	✓		✓	✓	✓	50.90	13.39	33.03

Ordered by decreasing BLEU

Table 9 Autoencoder language tasks

Method	ML	BT	Autoencoder			ENG–EST		
			SRC	HRL	LRL	chrF-1.0	BLEU	Rare
SRC+LRL AE	✓		✓		✓	51.71	14.04	34.79
LRL AE	✓				✓	51.41	13.93	33.57
SRC+HRL+LRL AE	✓		✓	✓	✓	51.42	13.75	33.83
No AE	✓					50.09	12.90	33.20

approach was clearly inferior, we did not use it for the other language pairs. However, we have not tested pretraining on a next token prediction or masked language modeling task.

5.4.4 Dataset augmentation: subword regularization

Table 7 shows an improvement between + 0.08 and + 0.55 BLEU from using subword regularization as the only noise model, without the use of an autoencoder.

5.4.5 Dataset augmentation: autoencoder

Table 8 shows an ablation experiment for the noise model. When compared against only using the subword regularization, the additional noises give between + 0.2 and + 0.5 BLEU. All parts of the noise model are individually ablated: the most important is local reordering, which when omitted causes a decrease of − 0.36 BLEU. The full noise model includes subword regularization. When subword regularization is ablated, we turn it entirely off, both for the parallel data and the autoencoder. Word boundary noise, taboo sampling, and insertions are not included in our full

Table 10 Results using back-translation

Method	ML	BT	Autoencoder			ENG–EST			ENG–DAN			ENG–SLO		
			SRC	HRL	LRL	chrF1	BLEU	Rare	chrF1	BLEU	Rare	chrF1	BLEU	Rare
Full BT	✓	✓	✓		✓	56.45	18.05	41.13	51.27	14.80	56.63	52.80	16.87	70.97
No AE, full BT	✓	✓				56.33	18.15	40.85	51.20	15.00	57.39	52.65	16.63	70.82
AE, no BT	✓		✓		✓	51.71	14.04	34.79	50.06	13.92	54.58	50.19	14.02	69.94
Vanilla BT		✓ †				36.12	5.51	13.25						

✓ †Indicates the use of a low-quality back-translation made with a non-multilingual non-autoencoder vanilla BT model

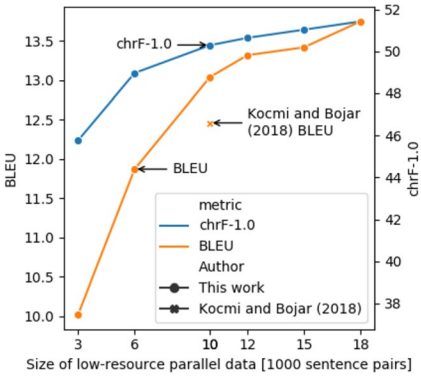

Fig. 7 Varying the amount of low-resource data. Multilingual models, with SRC + HRL + LRL autoencoder and full noise model. Results on English → Estonian newstest2018

noise model, as they did not show a benefit on the development set. However, word boundary noise gives + 0.2 BLEU and taboo sampling + 0.09 BLEU on the test set.

We also consider for which languages an autoencoder task should be added. Table 9 shows variants starting from no autoencoder, adding autoencoders one by one first for the low-resource target language, then for the source language and finally for the high-resource target language. The best combination uses source and LRL, with the SRC autoencoder giving a gain of + 0.11 BLEU over only using the LRL. The HRL autoencoder is detrimental, and leaving it out gives + 0.29 BLEU.

5.4.6 Dataset augmentation: back-translation

Table 10 shows the improvements gained using back-translated synthetic data. We weight the natural and synthetic LRL data equally. Back-translation is generally effective, giving a benefit between + 1.31 and + 4.46 BLEU. When using back-translated data, the autoencoder task is less effective, with small improvements to Character F_1 but inconsistent results for the other measures. Note that back-translation is not a silver bullet. The *Vanilla BT* system uses only back-translation, but not multilingual training or autoencoder: the back-translation is performed with a weak model trained only on the low-resource parallel data, and then a forward model is trained augmented only by this low-quality back-translation. The performance when using only back-translation is very low: only +2.87 BLEU better than the vanilla model without back-translation. The high-quality back-translation together with multilingual training gives an + 12.7 BLEU increase over the vanilla back-translation.

5.4.7 Amount of low-resource language data

Figure 7 shows how the performance degrades when the low-resource parallel data is reduced. Each set is subsampled from the previous larger set. All models use multilingual training with scheduled multi-task learning, and SRC + HRL + LRL autoencoders. Down to 10 k parallel sentences the performance stays reasonable, after which it rapidly deteriorates.

Table 11 HRL language relatedness

Method	ML	BT	Autoencoder			ENG–EST		
			SRC	HRL	LRL	chrF-1.0	BLEU	Rare
Within family	FIN		✓		✓	51.71	14.04	34.79
Cross family	CZE		✓		✓	50.20	13.12	30.69

Table 12 Results on Norwegian Bokmål–North Sámi Apertium story

Method	ML	BT	Autoencoder			NOB–SME		
			SRC	HRL	LRL	chrF-1.0	BLEU	Rare
ML, AE, BT	✓	✓	✓		✓	57.27	24.40	35.62
ML, AE	✓		✓		✓	54.86	21.07	21.54
Vanilla						45.97	15.64	21.05

Also plotted is a 10 k sentence pair baseline by Kocmi and Bojar (2018), reaching 12.46 BLEU in a similar setting on the same test set. Our result at 10 k is 13.04 BLEU, or + 0.68.

5.4.8 Relatedness of the target languages

Table 11 shows the results of using an unrelated but larger HRL (Czech). The results favor transfer from the related HRL (Finnish), by +0.92 BLEU. The difference in favor of the related HRL is largest for the rare words.

Previously, Zoph et al. (2016) and Dabre et al. (2017) find that related parent languages result in better transfer. However, Kocmi and Bojar (2018) find in the case of Estonian that a bigger parent (Czech) gave better results than a more related parent (Finnish). Our results contradict Kocmi and Bojar (2018) and agree with the prior literature.

5.4.9 Norwegian bokmål → Finnish + North Sámi

We apply the findings of the previous experiments to the low-resource pair Norwegian bokmål to North Sámi. We use a larger task mix weight for the LRL task (40 SRC-HRL/30 SRC-LRL/30 BT) to account for the larger LRL parallel data. Table 12 shows the results to be similar to the results of the other languages, with benefit from multilingual training, autoencoder task and back-translation.

5.5 Discussion

In our experiments for four asymmetric-resource one-to-many translation tasks, we find that the largest gains come from cross-lingual transfer (up to + 12.7 BLEU), back-translation (up to + 4.46 BLEU), and scheduled multi-task learning (up to + 2.4 BLEU). To sum up our findings related to the questions asked in the introduction:

On cross-lingual transfer, we find that applying scheduled multi-task learning is superior to both fully sequential and fully parallel transfer. In scheduled multi-task learning, the model is first pretrained on a mix of only high-resource tasks and then fine-tuned using a mix of both high- and low-resource tasks. A second fine-tuning phase only on the low-resource tasks is prone to overfitting.

On exploiting monolingual data, a low-resource target-language autoencoder is beneficial, even when using multilingual training, but inconclusive together with back-translation. A source-language autoencoder is also helpful, to a lesser degree, but a high-resource target autoencoder is not. A noise model including subword regularization, reordering, and deletion is beneficial. The results for substitutions and the proposed taboo sampling method are inconclusive.

On vocabulary construction, Morfessor EM+Prune is superior to SentencePiece in this translation setting, for a gain of + 0.6 BLEU. As the methods use the same training algorithm, it indicates that the prior used in Morfessor is beneficial in finding efficient subword lexicons. The vocabulary size has less effect (up to 0.5 BLEU for sizes between 8 k and 20 k) on the results. Subword lexicon size has been considered an important parameter to tune (Sennrich and Zhang 2019; Salesky et al. 2020). Also our preliminary experiments of low-resource NMT without subword regularization suggested a more substantial effect for the lexicon size. It seems that the subword sampling procedure (and perhaps the autoencoder task) lessens the impact of the subword vocabulary size.

Regarding available data and languages, larger low-resource parallel data give better results, but diminishing returns are already reached after 10 k sentences. We find language relatedness to be more important than parent language size in highly asymmetrical transfer. Sennrich and Zhang (2019) find that smaller models and batch sizes work better in low-resource settings. We find that large models are better whenever auxiliary multilingual or monolingual data is used. While in the vanilla setting, the smaller model is better, it still falls far behind the models using additional data.

Among the translation tasks, we get the lowest scores in the English–Danish translation. While Danish has the smallest LRL monolingual corpus, as the same order is observed also for the models not using monolingual data, the reason must lie elsewhere, possibly in the difficulty of the JRC-Acquis corpus. The autoencoder task has the largest benefit for English–Estonian. In the Norwegian–North Sámi experiment the size of the low-resource parallel data is an order of magnitude larger than in the other experiments, but the results remain similar. Due to the small size of the test set, we include the entire translation output in Online Resource 1.

The three evaluation measures—BLEU, Character F_1, and rare words F_1—generally agree. Some exceptions include ablation of the subword regularization and

using SwitchOut as the sole noise model, which hurt in particular the rare words more than BLEU. Turning off the autoencoder has the least effect on rare words, even giving a slight improvement for ENG–DAN when using back-translation.

Our results again underscore the need to gather parallel data for low-resource language pairs. This may be possible to accomplish at reasonable cost, as 10 k sentence pairs already goes a long way. Monolingual corpora of high quality and quantity are also of great importance as auxiliary data for MT.

6 Conclusion

When training a neural translation model for low-resource languages with limited parallel training data, it is important to make use of efficient methods for cross-lingual learning, data augmentation, and subword segmentation. Our experiments in asymmetric-resourced one-to-many translation show that the largest individual improvements come from any cross-lingual transfer learning and augmenting the training data with back-translation. However, considerable benefits are gained also by less common approaches: scheduled multi-task learning, subword regularization, and a denoising autoencoder with multiple noise models. For this reason, we strongly recommend that NMT frameworks should include a dataloader with the ability to (a) sample noisy minibatches for training and (b) use a schedule for controlling the mixing of different tasks. Subword sampling requires a probabilistic segmentation model such as SentencePiece or Morfessor, making them preferable to the more common BPE method. Both our data loader implementation for the OpenNMT-py system and the Morfessor EM+Prune software are available with non-restrictive licenses.

Supplementary Information The online version containssupplementary material available at https://doi.org/10.1007/s10590-020-09253-x.

Acknowledgements This study has been supported by the MeMAD project, funded by the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 780069), and the FoTran project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 771113). Computer resources within the Aalto University School of Science "Science-IT" project were used.

References

Arivazhagan N, Bapna A, Firat O, Lepikhin D, Johnson M, Krikun M, Chen MX, Cao Y, Foster G, Cherry C, Macherey W, Chen Z, Wu Y (2019) Massively multilingual neural machine translation in the wild: Findings and challenges. arXiv:1907.05019 [cs.CL]

Artetxe M, Labaka G, Agirre E, Cho K (2018) Unsupervised neural machine translation. In: Proceedings of the 6th international conference on learning representations (ICLR), http://arxiv.org/abs/1710.11041

Belinkov Y, Bisk Y (2017) Synthetic and natural noise both break neural machine translation. arXiv:1711.02173 [cs.CL]

Blackwood G, Ballesteros M, Ward T (2018) Multilingual neural machine translation with task-specific attention. In: Proceedings of the 27th international conference on computational linguistics, pp 3112–3122

Bojar O, Dušek O, Kocmi T, Libovický J, Novák M, Popel M, Sudarikov R, Variš D (2016) CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In: Sojka P, Horák A, Kopeček I, Pala K (eds) Text, speech, and dialogue: 19th international conference, TSD 2016, Masaryk University, Springer International Publishing, Cham/Heidelberg/New York/Dordrecht/London, no. 9924 in Lecture Notes in Artificial Intelligence, pp 231–238

Bojar O, Federmann C, Fishel M, Graham Y, Haddow B, Huck M, Koehn P, Monz C (2018) Findings of the 2018 conference on machine translation (wmt18). In: Proceedings of the third conference on machine translation, volume 2: shared task papers, association for computational linguistics, Belgium, Brussels, pp 272–307. http://www.aclweb.org/anthology/W18-6401

Bourlard H, Kamp Y (1988) Auto-association by multilayer perceptrons and singular value decomposition. Biol Cybern 59(4–5):291–294

Caruana R (1998) Multitask learning. Learning to learn. Springer, Berlin, pp 95–133

Caswell I, Chelba C, Grangier D (2019) Tagged back-translation. In: Proceedings of the fourth conference on machine translation (WMT) (Volume 1: Research Papers), pp 53–63

Chen Z, Badrinarayanan V, Lee CY, Rabinovich A (2018) Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: Proceedings of the international conference on machine learning (ICML), pp 794–803

Cheng Y, Xu W, He Z, He W, Wu H, Sun M, Liu Y (2016) Semi-supervised learning for neural machine translation. In: Proceedings of the 54th annual meeting of the association for computational linguistics (acl) (volume 1: long papers), pp 1965–1974, http://arxiv.org/abs/1606.04596

Cherry C, Foster G, Bapna A, Firat O, Macherey W (2018) Revisiting character-based neural machine translation with capacity and compression. In: Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP), Association for Computational Linguistics, Brussels, Belgium, pp 4295–4305, https://doi.org/10.18653/v1/D18-1461, https://www.aclweb.org/anthology/D18-1461

Chu C, Dabre R, Kurohashi S (2017) An empirical comparison of domain adaptation methods for neural machine translation. In: Proceedings of the 55th annual meeting of the association for computational linguistics (ACL) (Volume 2: Short Papers), Association for Computational Linguistics, Vancouver, pp 385–391, https://doi.org/10.18653/v1/P17-2061, https://www.aclweb.org/anthology/P17-2061

Chung J, Cho K, Bengio Y (2016) A character-level decoder without explicit segmentation for neural machine translation. In: Proceedings of the 54th annual meeting of the association for computational linguistics (ACL) (Volume 1: Long Papers), pp 1693–1703, http://arxiv.org/abs/1603.06147

Conneau A, Lample G (2019) Cross-lingual language model pretraining. In: Proceedings of advances in neural information processing systems (NIPS), pp 7059–7069, http://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining

Costa-jussà MR, Fonollosa JAR (2016) Character-based neural machine translation. In: Proceedings of the 54th annual meeting of the association for computational linguistics (ACL) (Volume 2: Short Papers), Association for Computational Linguistics, Berlin, Germany, pp 357–361, https://doi.org/10.18653/v1/P16-2058, https://www.aclweb.org/anthology/P16-2058

Costa-jussà MR, Escolano C, Fonollosa JAR (2017) Byte-based neural machine translation. In: Proceedings of the first workshop on subword and character level models in NLP, Association for Computational Linguistics, Copenhagen, Denmark, pp 154–158, https://doi.org/10.18653/v1/W17-4123, https://www.aclweb.org/anthology/W17-4123

Creutz M, Lagus K (2002) Unsupervised discovery of morphemes. In: Proceedings of the ACL-02 workshop on morphological and phonological learning (MPL), Association for Computational Linguistics, Philadelphia, Pennsylvania, vol 6, pp 21–30, https://doi.org/10.3115/1118647.1118650, http://portal.acm.org/citation.cfm?doid=1118647.1118650

Creutz M, Lagus K (2005) Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Technical Report A81, Publications in Computer and Information Science, Publications in Computer and Information Science, Helsinki University of Technology

Creutz M, Lagus K (2007) Unsupervised models for morpheme segmentation and morphology learning. ACM Trans Speech Lang Process 4(1):405

Currey A, Miceli-Barone AV, Heafield K (2017) Copied monolingual data improves low-resource neural machine translation. In: Proceedings of the second conference on machine translation (WMT), pp 148–156

Dabre R, Nakagawa T, Kazawa H (2017) An empirical study of language relatedness for transfer learning in neural machine translation. In: Proceedings of the 31st Pacific Asia conference on language, information and computation, pp 282–286

Dabre R, Chu C, Kunchukuttan A (2020) A comprehensive survey of multilingual neural machine translation. ArXiv:2001.01115 [cs.CL], arXiv:2001.01115

Dai AM, Le QV (2015) Semi-supervised sequence learning. In: Proceedings of advances in neural information processing systems (NIPS), pp 3079–3087

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc 39(1):1–38

Devlin J, Chang MW, Lee K, Toutanova K (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), pp 4171–4186, http://arxiv.org/abs/1810.04805

Di Gangi MA, Federico M (2017) Monolingual embeddings for low resourced neural machine translation. In: Proceedings of the 14th international workshop on spoken language translation (IWSLT'17), pp 97–104

Domhan T, Hieber F (2017) Using target-side monolingual data for neural machine translation through multi-task learning. In: Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP), pp 1500–1505

Edunov S, Ott M, Auli M, Grangier D (2018) Understanding back-translation at scale. In: Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP), pp 489–500

Firat O, Cho K, Bengio Y (2016) Multi-way, multilingual neural machine translation with a shared attention mechanism. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 866–875, http://arxiv.org/abs/1601.01073

Forcada ML, Ginestí-Rosell M, Nordfalk J, O'Regan J, Ortiz-Rojas S, Pérez-Ortiz JA, Sánchez-Martínez F, Ramírez-Sánchez G, Tyers FM (2011) Apertium: a free/open-source platform for rule-based machine translation. Mach Transl 25(2):127–144

Gage P (1994) A new algorithm for data compression. C Users J 12(2):23–38

Galuščáková P, Bojar O (2012) WMT 2011 testing set. http://hdl.handle.net/11858/00-097C-0000-0006-AADA-9, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University

Goldsmith J (2001) Unsupervised learning of the morphology of a natural language. Comput Linguist 27(2):153–198

Goodfellow IJ, Mirza M, Xiao D, Courville A, Bengio Y (2014) An empirical investigation of catastrophic forgetting in gradient-based neural networks. In: Proceedings of international conference on learning representations (ICLR), Citeseer, https://arxiv.org/abs/1312.6211

Graça M, Kim Y, Schamper J, Khadivi S, Ney H (2019) Generalizing back-translation in neural machine translation. In: Proceedings of the fourth conference on machine translation (volume 1: research papers), pp 45–52, https://arxiv.org/abs/1906.07286

Grönroos SA, Virpioja S, Kurimo M (2018) Cognate-aware morphological segmentation for multilingual neural translation. In: Proceedings of the third conference on machine translation, Association for Computational Linguistics, Brussels

Grönroos SA, Virpioja S, Kurimo M (2020) Morfessor EM+Prune: improved subword segmentation with expectation maximization and pruning. In: Proceedings of the 12th language resources and evaluation conference, ELRA, Marseilles

Gu J, Wang Y, Chen Y, Cho K, Li VOK (2018) Meta-learning for low-resource neural machine translation. In: Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP), pp 3622–3631, http://arxiv.org/abs/1808.08437

Gulcehre C, Firat O, Xu K, Cho K, Barrault L, Lin HC, Bougares F, Schwenk H, Bengio Y (2015) On using monolingual corpora in neural machine translation. http://arxiv.org/abs/1503.03535

Hammarström H, Borin L (2011) Unsupervised learning of morphology. Comput Linguist 37(2):309–350

Harris ZS (1955) From phoneme to morpheme. Language 31(2):190–222

He D, Xia Y, Qin T, Wang L, Yu N, Liu TY, Ma WY (2016) Dual learning for machine translation. In: Proceedings of advances in neural information processing systems (NIPS), pp 820–828, http://arxiv.org/abs/1611.00179

Iyyer M, Manjunatha V, Boyd-Graber J, Daumé III H (2015) Deep unordered composition rivals syntactic methods for text classification. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (ACL-IJCNLP) (Volume 1: Long Papers), pp 1681–1691

Johnson M, Schuster M, Le QV, Krikun M, Wu Y, Chen Z, Thorat N, Viégas F, Wattenberg M, Corrado G, Hughes M, Dean J (2017) Google's multilingual neural machine translation system: enabling zero-shot translation. Trans Assoc Comput Linguist 5:339–351

Joshi M, Chen D, Liu Y, Weld DS, Zettlemoyer L, Levy O (2020) Spanbert: Improving pre-training by representing and predicting spans. Trans Assoc Comput Linguist 8:64–77

Kalchbrenner N, Blunsom P (2013) Recurrent continuous translation models. In: Proceedings of the 2013 conference on empirical methods in natural language processing (EMNLP), pp 1700–1709

Karakanta A, Dehdari J, van Genabith J (2018) Neural machine translation for low-resource languages without parallel corpora. Mach Transl 32(1–2):167–189

Kiperwasser E, Ballesteros M (2018) Scheduled multi-task learning: from syntax to translation. Trans Assoc Comput Linguist 6:225–240

Klein G, Kim Y, Deng Y, Senellart J, Rush AM (2017) OpenNMT: Open-source toolkit for neural machine translation. In: Proceedings of the annual meeting of the association for computational linguistics (ACL). https://doi.org/10.18653/v1/P17-4012,arXiv: 1701.02810

Kocmi T (2019) Exploring benefits of transfer learning in neural machine translation. PhD Thesis, Charles University

Kocmi T, Bojar O (2018) Trivial transfer learning for low-resource neural machine translation. In: Proceedings of the third conference on machine translation (WMT): research papers, pp 244–252

Koehn P (2005) Europarl: a parallel corpus for statistical machine translation. MT Summit 5:79–86

Kohonen O, Virpioja S, Lagus K (2010) Semi-supervised learning of concatenative morphology. In: Proceedings of the 11th meeting of the ACL special interest group on computational morphology and phonology, association for computational linguistics, Uppsala, Sweden, pp 78–86, http://www.aclweb.org/anthology/W10-2210

Koponen M, Salmi L, Nikulin M (2019) A product and process analysis of post-editor corrections on neural, statistical and rule-based machine translation output. Mach Transl 33(1–2):61–90

Kreutzer J, Sokolov A (2018) Learning to segment inputs for NMT favors character-level processing. In: Proceedings of the 15th international workshop on spoken language translation (IWSLT), https://arxiv.org/abs/1810.01480

Kudo T (2018) Subword regularization: Improving neural network translation models with multiple subword candidates. In: Proceedings of the 56th annual meeting of the association for computational linguistics (ACL) (Volume 1: Long Papers), pp 66–75, http://arxiv.org/abs/1804.10959

Kudo T, Richardson J (2018) SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations, association for computational linguistics, Brussels, Belgium, pp 66–71, https://doi.org/10.18653/v1/D18-2012, https://www.aclweb.org/anthology/D18-2012

Kurimo M, Virpioja S, Turunen V, Lagus K (2010) Morpho challenge 2005-2010: Evaluations and results. In: Heinz J, Cahill L, Wicentowski R (eds) Proceedings of the 11th meeting of the ACL special interest group on computational morphology and phonology, association for computational linguistics, Uppsala, Sweden, pp 87–95

Lample G, Conneau A, Denoyer L, Ranzato M (2018a) Unsupervised machine translation using monolingual corpora only. In: International conference on learning representations (ICLR), http://arxiv.org/abs/1711.00043

Lample G, Ott M, Conneau A, Denoyer L, Ranzato M (2018b) Phrase-based & neural unsupervised machine translation. In: Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP), pp 5039–5049, https://www.aclweb.org/anthology/D18-1549.pdf

Lee YS (2004) Morphological analysis for statistical machine translation. In: Proceedings of HLT-NAACL 2004: short papers, Association for Computational Linguistics, pp 57–60

Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L (2019) BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ArXiv:1910.13461 [cs.CL], arXiv:1910.13461

Lison P, Tiedemann J (2016) OpenSubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In: Proceedings of the 10th international conference on language resources and evaluation (LREC 2016), European Language Resources Association

Luong MT (2016) Neural machine translation. PhD Thesis, Stanford University

Luong MT, Le QV, Sutskever I, Vinyals O, Kaiser L (2015) Multi-task sequence to sequence learning. In: Proceedings of international conference on learning representations (ICLR), http://arxiv.org/abs/1511.06114

McCloskey M, Cohen NJ (1989) Catastrophic interference in connectionist networks: the sequential learning problem. In: Psychology of learning and motivation, vol 24, Elsevier, pp 109–165

Mueller A, Nicolai G, McCarthy AD, Lewis D, Wu W, Yarowsky D (2020) An analysis of massively multilingual neural machine translation for low-resource languages. In: Proceedings of The 12th language resources and evaluation conference, pp 3710–3718

Oflazer K, El-Kahlout ID (2007) Exploring different representational units in English-to-Turkish statistical machine translation. In: Proceedings of the second workshop on statistical machine translation, Association for Computational Linguistics, pp 25–32, https://doi.org/10.3115/1626355.1626359, http://portal.acm.org/citation.cfm?doid=1626355.1626359

Östling R, Tiedemann J (2017) Neural machine translation for low-resource languages. ArXiv:1708.05729 [cs.CL], arXiv:1708.05729

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: 40th annual meeting of the association for computational linguistics, Association for Computational Linguistics, Philadelphia, pp 311–318, https://doi.org/10.3115/1073083.1073135, http://portal.acm.org/citation.cfm?doid=1073083.1073135

Platanios EA, Sachan M, Neubig G, Mitchell T (2018) Contextual parameter generation for universal neural machine translation. In: Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP), pp 425–435, https://www.aclweb.org/anthology/D18-1039

Popović M (2015) chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the 10th workshop on statistical machine translation (WMT), association for computational linguistics, pp 392–395, https://doi.org/10.18653/v1/W15-3049, http://aclweb.org/anthology/W15-3049

Ramachandran P, Liu PJ, Le Q (2017) Unsupervised pretraining for sequence to sequence learning. In: Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP), pp 383–391

Rissanen J (1989) Stochastic complexity in statistical inquiry, vol 15. World Scientific Series in Computer Science, Singapore

Sachan DS, Neubig G (2018) Parameter sharing methods for multilingual self-attentional translation models. In: Proceedings of the third conference on machine translation (WMT): research papers, pp 261–271, https://www.aclweb.org/anthology/W18-6327

Salesky E, Runge A, Coda A, Niehues J, Neubig G (2020) Optimizing segmentation granularity for neural machine translation. Mach Transl pp 1–19

Scott SL (2002) Bayesian methods for hidden Markov models: recursive computing in the 21st century. J Am Stat Assoc 97(457):337–351

Sennrich R, Zhang B (2019) Revisiting low-resource neural machine translation: a case study. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 211–221

Sennrich R, Haddow B, Birch A (2015) Neural machine translation of rare words with subword units. In: Proceedings of the annual meeting of the association for computational linguistics (ACL), http://arxiv.org/abs/1508.07909

Sennrich R, Haddow B, Birch A (2016) 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), pp 86–96

Skorokhodov I, Rykachevskiy A, Emelyanenko D, Slotin S, Ponkratov A (2018) Semi-supervised neural machine translation with language models. In: Proceedings of the AMTA 2018 workshop on technologies for MT of low resource languages (LoResMT 2018), pp 37–44

Song K, Tan X, Qin T, Lu J, Liu TY (2019) Mass: Masked sequence to sequence pre-training for language generation. In: Proceedings of the international conference on machine learning (ICML), pp 5926–5936

Sriram A, Jun H, Satheesh S, Coates A (2017) Cold fusion: Training seq2seq models together with language models. In: Proceedings of the Interspeech 2018, https://arxiv.org/abs/1708.06426

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Stahlberg F, Cross J, Stoyanov V (2018) Simple fusion: Return of the language model. In: Proceedings of the third conference on machine translation (WMT): research papers, pp 204–211

Steinberger R, Pouliquen B, Widiger A, Ignat C, Erjavec T, Tufiş D, Varga D (2006) The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In: Proceedings of the 5th international conference on language resources and evaluation (LREC'2006), Genoa, Italy

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of advances in neural information processing systems (NIPS), pp 3104–3112

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826, http://arxiv.org/abs/1512.00567

Thompson B, Khayrallah H, Anastasopoulos A, McCarthy AD, Duh K, Marvin R, McNamee P, Gwinnup J, Anderson T, Koehn P (2018) Freezing subnetworks to analyze domain adaptation in neural machine translation. In: Proceedings of the third conference on machine translation (WMT): research papers, pp 124–132

Tiedemann J (2009) Character-based PSMT for closely related languages. In: Proceedings of the 13th conference of the european association for machine translation (EAMT 2009), pp 12–19

Toral A, Sánchez-Cartagena VM (2017) A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics (EACL): Volume 1, Long Papers, pp 1063–1073, https://www.aclweb.org/anthology/E17-1100

Torrey L, Shavlik J (2009) Transfer learning. In: Olivas ES (ed) Handbook of research on machine learning applications and trends: algorithms, methods, and techniques: algorithms, methods, and techniques, IGI Global, pp 242–264

Tu Z, Liu Y, Shang L, Liu X, Li H (2017) Neural machine translation with reconstruction. In: Thirty-first AAAI conference on artificial intelligence, http://arxiv.org/abs/1611.01874

Vaibhav V, Singh S, Stewart C, Neubig G (2019) Improving robustness of machine translation with synthetic noise. 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), pp 1916–1920, https://www.aclweb.org/anthology/N19-1190/

Varjokallio M, Kurimo M, Virpioja S (2013) Learning a subword vocabulary based on unigram likelihood. In: Proceedings of the 2013 IEEE workshop on automatic speech recognition and understanding (ASRU), IEEE, Olomouc, Czech Republic, pp 7–12, https://doi.org/10.1109/ASRU.2013.6707697, http://ieeexplore.ieee.org/document/6707697/

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 6000–6010, http://arxiv.org/abs/1706.03762

Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on machine learning (ICML), pp 1096–1103

Virpioja S, Väyrynen JJ, Creutz M, Sadeniemi M (2007) Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. Machine Translation Summit XI, Copenhagen, Denmark 2007:491–498

Virpioja S, Turunen VT, Spiegler S, Kohonen O, Kurimo M (2011) Empirical comparison of evaluation methods for unsupervised learning of morphology. Traitement Automatique des Langues 52(2):45–90, http://www.atala.org/Empirical-Comparison-of-Evaluation

Virpioja S, Smit P, Grönroos SA, Kurimo M (2013) Morfessor 2.0: Python implementation and extensions for Morfessor Baseline. Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, Department of Signal Processing and Acoustics, Aalto University, Helsinki, Finland

Wang X, Pham H, Dai Z, Neubig G (2018) Switchout: an efficient data augmentation algorithm for neural machine translation. In: Proceedings of the 2018 conference on empirical methods in natural language processing (EMNLP), pp 856–861, https://www.aclweb.org/anthology/D18-1100

Yang Z, Chen W, Wang F, Xu B (2018) Unsupervised neural machine translation with weight sharing. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 46–55

Zhang J, Zong C (2016) Exploiting source-side monolingual data in neural machine translation. In: Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP), pp 1535–1545

Zoph B, Yuret D, May J, Knight K (2016) Transfer learning for low-resource neural machine translation. In: Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP), pp 1568–1575