Towards Effective Machine Translation For a Low-Resource Agglutinative Language: Karachay-Balkar

Enora Rice
Advised by Jonathan Washington

December 2021

1 Abstract

Neural machine translation (NMT) is often heralded as the most effective approach to machine translation due to its success on language pairs with large parallel corpora. However, neural methods produce less than ideal results on low-resource languages when their performance is evaluated using accuracy metrics like the Bilingual Evaluation Understudy (BLEU) score. One alternative to NMT is rule-based machine-translation (RBMT), but it too has drawbacks. Furthermore, little research has been done to compare the two approaches on criteria beyond their respective accuracies. This thesis evaluates RBMT and NMT systems holistically based on efficacy, ethicality, and utility to low-resource language communities. Using the language Karachay-Balkar as a case-study, the latter half of this thesis investigates how two free and open-source machine translation packages, Apertium (rule-based) and JoeyNMT (neural), might support community-driven machine translation development. While neither platform is found to be ideal, this thesis finds that the Apertium is more conducive to a community driven machine translation development process than JoeyNMT when evaluated on the criteria of efficiency, accessibility, ease of deployment, and interpretability.

2 Acknowledgements

This thesis would not have been possible without the guidance of my advisor Jonathan Washington and my second-reader Alvin Grissom. In particular I would like to thank Jonathan Washington for contributing critical improvements to the Apertium based Karachay-Balkar translator, more than doubling the BLEU score of the system.

Contents

3 Introduction

Karachay-Balkar (krc) is a Turkic language spoken by two populations living in the North Caucasus: the Karachays and the Balkars. According to the 2010 Russian census, there are 310,000 native speakers of Karachay-Balkar living in Russia, and additional speakers can be found living in Turkey, and the USA. The language is not supported by formal institutions but is still used widely in the community and the home (Eberhard et al., 2021). There have been some efforts towards developing machine translation tools for Karachay-Balkar, but the language lacks a widely available, high-performing translation system. My primary reason for choosing Karachay-Balkar as a language of focus is that Karachay-Balkar/English is a low-resource language pair. The nebulous definition of the term low-resource is discussed at length in section 3.1, but narrowly defined, the designation signifies that a language pair has a small parallel corpus.[a] Karachay-Balkar is also an interesting case study because it is agglutinative[b], which can be challenging to handle in machine translation systems.

This thesis provides a survey of the state of low-resource machine translation research with a particular emphasis on technology that is useful to the translation of agglutinative languages. I evaluate two approaches in this thesis: rule-based machine translation (RBMT) and neural machine translation (NMT). At this time, NMT is a far more popular approach to machine translation than RBMT because it consistently achieves desirable results on high-resource language pairs. However, NMT still underperforms on low-resource language pairs, so it is worthwhile to consider the alternatives.

Though this thesis focuses on machine translation of Karachay-Balkar, it is intended to be relevant to other low-resource languages. I have chosen to put a particular emphasis on methods that support community driven efforts because of the complicated, often extractive relationship between global tech giants and Indigenous language communities, which is discussed in further detail in section 3.4.

In the latter half of this thesis, I build systems for translating Karachay-Balkar to English using two free, and open-source machine translation packages, JoeyNMT and Apertium, which are neural and rule-based respectively. I evaluate the models not only on their accuracy, but on how well their development process could support a community driven effort. In addition to considering quantitative metrics such as the BiLingual Evaluation Understudy (BLEU)[1] score, this thesis evaluates the platforms on the criteria of efficiency, accessibility, interpretability, and ease of deployment. In considering all of these factors, I find that rule-based methods are still viable and necessary for the development of low-resource machine translation, and have not been made obsolete by neural methods.

4 Background

4.1 Defining Low-Resource

For the purposes of this thesis, I define low-resource, as it is defined by the NMT research community, to refer to languages pairs that have small parallel corpora. There is no agreed upon cutoff for exactly how small a parallel corpus must be before a language pair is considered low resource, but state-of-the-art NMT systems are typically trained on tens or hundreds of millions of parallel sentences. Even a parallel corpus of 100k sentences is often considered low-resource. However, outside of the scope of this thesis, the term "low-resource" carries connotations that go beyond the size of a parallel corpus. "Low-resource" tends to be misused as a synonym for "endangered," but is important to differentiate the two terms (Hämäläinen, 2021). According to Cieri et al. (2016), "Endangered refers to languages that are at risk of losing their native speakers through a combination of death and shift to other languages." However, this definition is incomplete because it does not emphasize the fact that in order for speaker death to impact the the vitality of a language, there has to be interrupted intergenerational transmission, which comes as a direct result of oppression. We must not minimize the complexity of status. Ethnologue estimates that there are 3,018 endangered languages in the world today (Eberhard et al., 2021).

Whether or not a language is endangered has little to do with the size of its parallel corpora. A small parallel corpus for a given language pair does not mean either language that pair is endangered. Karachay-Balkar/English is one of many language pairs that is low-resource, but neither English nor Karachay-Balkar is officially considered endangered. The Expanded Graded Intergenerational Disruption Scale (EGIDS) is a metric that is used to denote the degree to which a given language is endangered

[a] A parallel corpus consists of a source language corpus that is aligned to a translation corpus in a target language

[b] An agglutinative language is one that makes heavy use of agglutination in its morphology. Agglutination is the morphological process where words are formed by stringing together morphemes that each correspond to a single syntactic structure.

(Lewis & Simons 2010). The scale ranges from 1, signifying that the language is used internationally, to 9, signifying that the language is dormant. A score of 6b (threatened) indicates that the language is used for face-to-face communication intergenerationally but is actively losing speakers. Languages that rank 6b and below are typically considered to be endangered. In 2010, Karachay-Balkar had a score of 6a (vigorous), so it barely missed the threshold (this may have changed in recent years). The only difference between designations of 6b and 6a is that the latter score indicates that the state of the language has been deemed sustainable.

A minority language is a language that is spoken by a minority of speakers in a region, and whose speakers operate in social and political contexts with a majority language. Often the term minority language implies that a language is less politically and materially resourced (Lackaff and Moner 2016). In some regions where Karachay-Balkar is spoken, it is a minority language.

To varying degrees, minority languages are subject to the complex sociological pressures that threaten endangered languages and can themselves be considered "at risk." Minority languages face political and social persecution at the hands of majority language speakers. Many minority language speakers are punished for speaking their native languages in schools, driven to speak majority languages for career success, and cut off from their communities due to political unrest. Communication technology and social media pose another threat that is increasingly important to consider. While social media provides an opportunity for members of minority language communities to interact with one another online regardless of their geographic dispersion, it also makes majority languages more accessible, which can "encroach on the cultural prevalence of a minority language in certain contexts" (Lackaff and Moner 2016). Lackaff and Moner give the example of the Irish language, which was able to thrive in pockets of rural Ireland throughout centuries of English colonialism precisely because the social and economic isolation of those regions restricted speakers' interaction with the English language. In the modern day, it is increasingly rare for a language to remain isolated, and social media plays an critical role in this.

Because of the complex interdependence between the terms "low-resource," "endangered," and "minority", I address the technological challenges faced by all three designations in developing machine translation tools. I place a specific emphasis on "low-resource" languages only because that is a term preferred by the NMT field due to its quantifiability and apoliticism.

4.2 Overview of Neural Networks for Machine Translation

Neural machine translation falls into the domain of sequence-to-sequence (seq2seq) learning, which is a specific class of machine learning that involves converting sequences from one domain into sequences in another domain. Broadly speaking, a seq2seq model consists of two components: an encoder and a decoder. In machine translation, a seq2seq model is trained on a parallel corpus of sentences in the source and target languages. During the training process, a seq2seq model gradually learns the translation task, by updating hidden parameters within itself. Once trained, the model will output a sequence in the target language when fed a sequence in the source language. This is called prediction. During prediction, the the encoder converts an input sequence into a vector representation called a sentence embedding[2] based on the parameters it learned during training. Then, the decoder takes that vector and converts it to text in the target language.

There are a few different ways to construct encoder-decoder models, but in order to understand the mechanisms underlying any of the models, it is important to first understand neural networks. All encoder-decoder models rely on neural networks to some extent. A neural network[3] is a machine learning model that learns to classify a set of inputs into a set of output categories or classes. When used in conjunction with other methods, neural networks serve as building blocks of NMT. Neural networks are different from many of the machine learning models that preceded them, because instead of being fed carefully defined input features on which to base classification, neural networks can often figure out features as they learn. This fact is enormously useful for NLP applications because it is sometimes hard to manually assess which linguistic features are most relevant to a given task.

The most basic kind of neural network is called a feed-forward neural network[4], but in practice, there is a specific kind of neural network favored by NLP researchers called a recurrent neural network (RNN).[5] RNNs differ from feed-forward neural networks in that they have an internal memory mechanism which allows them to perform much better than feed-forward models on sequential data. In the context of NMT, this means that when parsing an sentence word by word, an RNN can more effectively remember the context of the previous words. Unfortunately, RNNs struggle to correctly parse long input sentences due to a problem known as the vanishing gradient problem[6]. Researchers have addressed this issue by developing an RNN adaptation called long short-term memory (LSTM)[7] (Hochreiter et al. 1997).

LSTMs are RNNs that are explicitly designed to be capable of learning long-term dependencies. Though LSTMs are more successful than vanilla RNNs, LSTMs are still limited in their ability to handle long input sequences because they are forced to compress input into a fixed length internal vector which can result in data loss. Both LSTMs and RNNs can be used as encoders and decoders for seq2seq machine translation.[8] During training the encoder LSTM learns how to encode the input text as a large fixed-dimensional vector representation and the decoder learns to decode this encoding to output words in the target language. (Sutskever et al. 2014).

Bahdanau et al. (2015) introduced a mechanism called attention[9] that improved performance even further in seq2seq models. At a high level, attention allows the decoder to map the most important content in the input sentence. By retaining information from the encoder, the decoder gains broader access to the context of the input sentence and can effectively search for parts of the source sentence relevant to predicting a target word. Attention has dramatically increased the performance of seq2seq models (Bahdanau et al. 2015). Attention can be used with RNNs, but the more effective use of attention is in the current state-of-the art seq2seq model: the Transformer[10] (Vaswani et al., 2017). In Transformers, the encoder and decoder blocks are actually stacks of multiple encoders and decoders. Encoders feed into one-another until the final encoder in the stack, which passes its output to the decoder stack. Each encoder and decoder layer employ a form of attention called self-attention, which allows them to better understand the context of their input. Over the past few years, Transformer based NMT has only continued to increase in popularity and accuracy. The NMT system developed in this thesis is Transformer based and built with JoeyNMT, which is a minimalist PyTorch[c] based neural machine translation toolkit that was designed to be accessible to novices (Kreutzer et al. 2019).

4.3 Overview of Rule-Based Machine Translation

The RBMT system developed in this thesis is based on the Apertium machine translation platform (Forcada et al. 2010). The Apertium translation engine pipeline consists of nine modules: a deformatter, morphological analyzer, a lexical disambiguator, a lexical transfer module, a lexical selection module, structural transfer module, a morphological generator, a post-generator finite state transducer (FST), and a reformatter. To give a high-level overview of how a string is processed, it is first deformatted so that it is interpretable to the system and then it is passed through a morphological analyzer that takes the surface form of each word and produces its lemma along with tags that correspond to the part of speech and any relevant morphological information. In cases where the meaning or form of the word in the source language is unclear, there is a module that disambiguates and returns the correct form based on context. Each lemma and its associated tags are passed through the lexical transfer module which outputs the corresponding lemma and tags in the target-language. If there are multiple possible outputs, a lexical selection module returns the correct form based on context from the input string. Structural transfer rules then dictate how the grammatical structures in the source language map to structures in the target language and the tags and word order are altered appropriately. The resulting set of lemmas and tags is then passed into a morphological generator that applies the appropriate inflection based on the tags. The translated string is then processed to fix any minor orthographic issues and reformatted for output. Every stage of the pipeline is hand specified by linguistic rules written in the appropriate formalism.

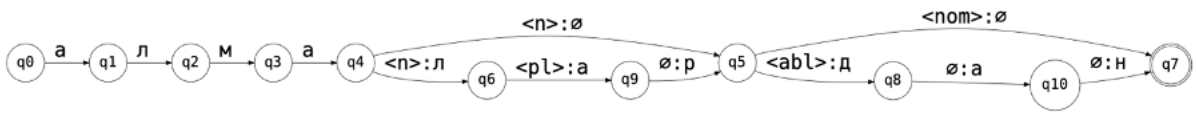

Figure 1: A morphological transducer containing form-to-analysis mappings of алмалардан 'from the apples' (Washington et al. 2021)

The workhorse of the Apertium RBMT pipeline is the morphological transducer, which is the technology underlying the morphological analyzer and generator (figure 1). Morphological transducers are based on Helsinki Finite State Technology (Linden et al., 2011), a free and open-source re-implementation

[c] An open-source machine learning Python library

of the Xerox finite-state tool-chain. Morphological transducers map the surface form of a word to its morphemes and morphological tags and vice versa.

5 Evaluation of the State of the Field

The overwhelming majority of contemporary machine translation research is NMT oriented. One might argue that the issues faced by NMT systems are more pressing to address because of NMT's prevalence. However, I have attempted to make this section a holistic review, mentioning RBMT as a point of comparison to NMT where possible. Section 3.1 covers how varying levels of resource availability are relevant to the performance of NMT systems, briefly touching on how RBMT compares. Section 3.2 discusses ethics and bias in NMT. Section 3.3 broadly discusses the perils vulnerable language communities face when developing language technology of any kind. Section 3.4 evaluates alternatives to NMT and their promises and pitfalls. Finally, section 3.5 serves as a literature review of advances in low-resource NMT for agglutinative languages, specifically through the lens of what might be relevant to Karachay-Balkar.

5.1 Levels of Resource Availability

The success of NMT is dependent on the availability of a large amount of parallel training data (Wang et al. 2021). Thousands of the world's languages do not have large corpora readily available to researchers and language technology developers. As a result, these languages are left behind by the NMT research community at large (Joshi et al. 2020).

Joshi et al. (2020) provide a useful taxonomy for discussing degrees of resource availability. The study divides languages into six classes based on availability of both labeled and unlabeled language data. Category 0, "The Left-Behinds," includes languages with exceptionally limited corpora, like Dahalo and Warlpiri, whereas category 5, "The Winners," includes languages with a dominant online presence, like English and French. Categories 0, 1, and 2, which account for around 94% of the world's languages, are considered to be low-resource, and are at an enormous disadvantage for developing natural language processing (NLP) tools (Joshi et al. 2020). The same study looks at the WALS data (Dryer and Haspelmath, 2013), which contains typological features for 2679 languages, in order to evaluate the distribution of typological features across language resource categories. In the WALS data, there are a total of 192 typological features with an average of 5.93 categories per features. Joshi et al. find that there are 549 unique typological categories represented in the categories 0, 1, and 2 (low-resource languages) that are not represented in 3, 4, and 5 (high resource languages). This finding is extremely relevant to current low-resource NMT research, as much of it relies on a mechanism called transfer learning, in which a machine learning model can store information it learned from solving one problem and apply it to another. NMT models can use transfer learning to learn features from high-resource languages to make translation coverage more robust in low-resource settings (effectively filling in the gaps in linguistic understanding). However, when a language has typological features that are not represented in any high-resource language, transfer learning may not be particularly useful. It is possible that employing transfer learning with two languages that are typologically dissimilar could even negatively impact the accuracy of the system.

As for RBMT, levels of resource availability have less of a direct impact on the performance of the system– at least when it comes to parallel corpora. Because RBMT is not a corpus-based approach, it makes no difference whether you have a parallel corpus of one sentence or one million sentences. If a community wishes to develop a RBMT system, the resources that matter are access to grammatical documentation, linguistic expertise, and time. These resources are harder to quantify and assess so there is no helpful taxonomy for resource levels on that front.

5.2 Against Extractive Language Technology

In evaluating machine translation systems, researchers tend to place a high value on the system's accuracy, specifically its BLEU score. Even in this thesis, I return to BLEU repeatedly because it is an easy and standardized metric for comparing MT systems to one another. However, it is important not to lose sight of the fact that evaluating language technology goes beyond evaluating its accuracy. The ethical nuance of dealing with vulnerable language communities is a significant issue that is present regardless of one's approach to machine translation. Though, as discussed earlier in this thesis, "low-resource," "endangered," and "minority" are not synonymous, the three statuses often come hand in hand. So, in developing low-resource language technology, it is important to consider how endangered and minority

languages communities might receive your work. Often, the process of developing language technology can alienate language communities. Decontextualizing language from its speakers and commodifying it in the form of data can reinforce colonialist ideologies as it creates "technology [that] does not address the social injustices that underly [sic] language endangerment" (Bird 2020).

There are many documented cases where tech companies, purportedly with altruistic motives, destructively exert their power over low-resource language communities during the development of language technology. Take, for example, the conflict between Mayan K'ichee' speakers and Microsoft when Microsoft attempted to develop a K'ichee' version of Windows without engaging with the appropriate Indigenous institutions (Romero 2016). The situation was complicated because K'ichee' is not a single language at all, but rather a collection of many different regional varieties tied closely to ethnic identity. No single variety of K'ichee' considered to be "standard," so matters relating to standardization and revitalization are sensitive and require cultural expertise. The Maya-run "Academy of the Mayan Languages of Guatemala" (ALMG) is an institution that exists to manage just such affairs. For Microsoft to neglect to consult with ALMG, instead hiring individual contractors and carelessly selecting a single dialect to represent standard K'ichee', undermined K'ichee' linguistic sovereignty. The language technology that was intended to support language revitalization efforts did little more than sow discontent.

So how can one develop language technology ethically? Many researchers and activists propose throwing out traditional, extractive methods for developing language technology, and adopting a community-driven, collaborative approach that prioritizes the goals of the language community (Bird 2020). Bird (2020) calls for decolonizing practices in language work, emphasizing the importance of Indigenous peoples' right to self-determination, and stating that the role of outside experts should be to support an approach the community has already settled on rather than to impose outside "expertise". The first step in supporting an Indigenous community's right to self-determination as an outsider is to identify community goals, which may not align with an outside language ideology. Any good evaluation of a machine translation system should include an assessment of its value to the community and alignment with their goals.

5.3 Ethics and Bias in NMT

Neural models can pick up on subtle biases in data and reproduce harmful ideologies (Bender et al. 2021). Prates et al. (2020) presents a straightforward example of this occurring in a machine translation context. The study shows that when asked to translate sentences like "He/she is a(n) [profession]" from languages which do not signal gender in their third person pronouns (such as Hungarian, Chinese, and Yoruba) to languages that do, Google Translate exhibits a strong tendency towards male defaults for professions where there is a male stereotype (Prates et al. 2020). In fact, the gender distribution across Google Translate outputs fails to replicate the real-world distribution of women in those fields. Instead, the translations yield male defaults more frequently than one would expect based on demographic information alone, thus reinforcing and amplifying gender bias (Prates et al. 2020).[d]

Although most studies of bias, including Prates et al. (2020), only address high-resource languages, there is no reason why low-resource languages should be immune to the machine bias challenges that their high-resource counterparts face. Available training data for low-resource languages is never truly representative, as it often comes from only a handful of sources across very limited domains (Haddow et al. 2021). For example, the parallel corpus available for Karachay-Balkar is essentially limited to the New Testament (Mirzakhalov et al. 2021b). Not only does this make for poor performance (Haddow et al. 2021), but neural machine translators can only learn the specific dialects and language patterns that are reflected in the sources they are trained on, so whatever biases are present in the training data will be over-represented in a model's output.

Beyond enforcing hegemonic worldviews and stereotypes through biased outputs, machine translation, when compared to human translation, can offer a poor reflection of the lexical richness and diversity in the target language. Translations produced by NMT models over-represent common words in the language and under-represent uncommon words. The ability of NMT models to generalize is seen as an asset because it allows translators to translate beyond the exact contexts represented in their training data. But, in fact, overgeneralization might be the cause underlying the loss of lexical richness and could potentially even exacerbate translation biases (Vanmassenhove et al. 2019). Given these findings, it is important to consider the possibility that releasing a poor-quality machine translation system into a vulnerable language community could adversely affect the longevity of the language by decreasing

[d] It is important to note that RBMT systems are subject to gender bias as well, but the phenomenon has been less thoroughly studied (Savoldi et al. 2021).

language learners' awareness of its lexical diversity, contributing to the demise of language variants that are not represented in the translation system, and fostering a culture of linguistic purism.

5.4 Alternatives to NMT

5.4.1 Statistical Methods

Before the rise of NMT, statistical machine translation (SMT) reigned supreme. SMT is an approach to machine translation where translations are generated based on statistical models learned from a parallel corpus. Generally, SMT systems require less data to train than NMT systems, which is why they are often brought up in discussions of low-resource language translation. Koehn and Knowles (2017) simulated varying corpus sizes by partitioning a parallel English-Spanish corpus in order to compare the performance of NMT to SMT. For their NMT model, they trained an attention based RNN encoder-decoder and for their SMT model, they used a phrase-based system. They found that SMT outperformed NMT for corpora under 15 million words.

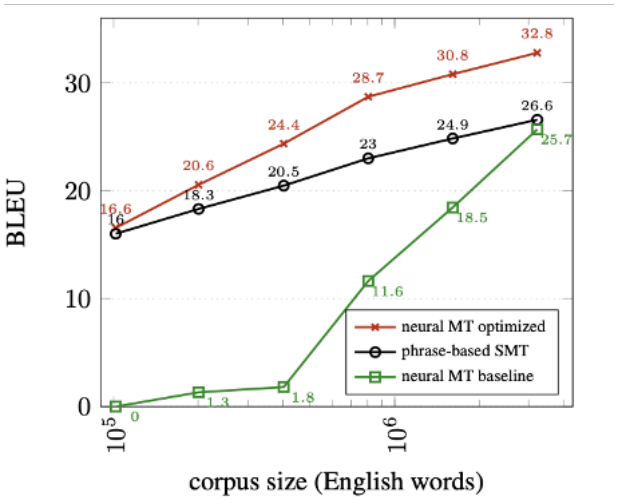

Figure 2: Performance of NMT versus SMT (Sennrich & Zhang, 2019)

However, the validity of these results has been questioned. According to Sennrich and Zhang (2019), Koehn and Knowles misrepresent the NMT by training their systems with hyperparameters suited for high-resource settings rather than fine-tuning for low-resource conditions. By altering several hyperparameters, Sennrich and Zhang improve the BLEU score 9.4 points in the ultra-low resource setting (100k words) and beat SMT across the board (figure 2). SMT is not discussed at length in this thesis because, as a corpus-based method, it suffers from many of the same pitfalls as NMT, but NMT models perform better overall.

5.4.2 Rule-Based Methods

Of the three methods mentioned in this thesis, RBMT gets the least attention. There are several reasons for this:

- **RBMT requires linguistic expertise.** RBMT systems rely on handwritten, built-in rules that dictate how two languages correspond to one another on a lexical, syntactic, and morphophonological level. To construct a RBMT system, one must have deep linguistic comprehension of both the source and target language or have access to robust documentation.
- **RBMT systems are comparatively slower to develop.** Encoding the rules of a language by hand takes a lot longer than making a machine do it for you.
- **RBMT systems cannot generalize over unknown inputs.** A blessing and a curse of NMT models is that they can generalize what they learn from one context to new unfamiliar contexts. A RBMT translation system is not going to learn how to translate novel linguistic information unless that linguistic information is explicitly encoded.

However, there are a lot of advantages to RBMT, the main advantage being that the accuracy of an RBMT system is not constrained by the size of the parallel corpora. Another major asset of RBMT is that errors in rule-based systems can be identified and fixed directly, whereas it is almost impossible to target specific errors in NMT systems. This comparison is drawn more explicitly in section 6.

5.5 Advances in NMT for Low-Resource Agglutinative languages

Having considered the pitfalls of neural machine translation in depth as well as its alternatives, it is worthwhile to discuss promising developments in neural methods given their dominance in the field. In this thesis I consider the relevant literature from two angles: general improvements to low-resource translation, and novel approaches to agglutinative language translation.

5.5.1 General Advances in Low-Resource NMT

Because parallel sentence data is limited, many recent advances in low-resource machine translation focus on leveraging data other than parallel corpora. Monolingual data is a largely under-utilized resource for low-resource machine translation because while low-resource language pairs might not have a large parallel corpus, some do have abundant monolingual text data. For example, there is a roughly 400,000-word monolingual Karachay-Balkar corpus available online, which does not quite size up to the over 160TB of monolingual English data available through Common Crawl[e] but is a step up from the around 133,000 words in the parallel corpus.

This thesis reviews two main approaches that leverage monolingual data: back translation, and unsupervised NMT[f]. Back translation is the process of re-translating content from the target language back into the source language. Sennrich et al. (2016) propose back translation as a method of utilizing target side monolingual data to improve NMT performance. Back translation is used to augment training data by adding synthetic parallel text generated by automatic translation of target side monolingual data. As a comparison, Sennrich et al. also experiment with treating monlingual target side training examples as parallel example with an empty source side.[11] Following the NMT architecture by Bahdanau et al. (2015), Sennrich et al. evaluate English-German and Turkish-English translators with and without back-translation. The study finds that mixing non-synthetic training data with synthetic data improves over the baseline by +2.1-3.4 BLEU. Using empty source side data also improved the performance of the Turkish-English model (albeit less so) at +0.6 BLEU on average.

Artetxe et al. (2018) present a novel approach to NMT system that relies on nothing but monolingual corpora and is trained using back translation. The study exploits the dual nature of machine translation to handle both translation directions at once, source-target and target-source, by making use of a shared encoder. The goal of a shared encoder is to produce a vector representation of the input text that is language independent, leaving the decoder to handle transforming the vector into the appropriate language. Artetxe et al. (2018) enhanced their model by inserting random noise into input strings in order to train the system to reconstruct the original input through a process called denoising. Through denoising the model learns the structure and word order of the input language. The system also employs the back translation technique introduced in Sennrich et al. (2016) to generate pseudo-parallel sentence pairs and train the model to predict the original sentence from the synthetic translation. The unsupervised model achieves strong results (considering that it was trained without parallel data) at 15.13 BLEU, but does not approach the state-of-the-art supervised model for the high resource language pair English-French at 38.95 BLEU. Nevertheless, the approach presented by Artetxe et al. is promising for low-resource languages that have a large amount of monolingual data.

Data from auxiliary languages has also proven to be useful in low-resource NMT thanks to transfer learning. Transfer learning could be very useful in developing an effective translator for Karachay-Balkar because there are Turkic languages that have far larger parallel corpora than Karachay-Balkar and English, such as English-Turkish which has a parallel corpus of 39.9 million sentences (Mirzakhalov et al. 2021b). The key idea behind transfer learning for NMT is to initialize and constrain the training of a low-resource pair (child-model) by reusing learned parameters from a model trained on a high-resource language pair (parent-model). When transfer learning is not employed, weight vectors are initialized randomly, but with transfer learning, the weights in the child model are initialized with the weights from the parent model. Zoph et al. (2016) is a seminal study on transfer learning that employs transfer learning with French-English as the parent model and four child source languages: Hausa, Turkish, Uzbek, and

[e] https://commoncrawl.org/

[f] An unsupervised model is a model that is trained on unlabeled data (e.g. monolingual data without translations)

Urdu. Using transfer learning, Zoph et al. improved the baseline NMT models by an average of 5.6 BLEU on the four low-resource language pairs. Zoph et al. also present evidence that choice of parent language is important and using a parent language more similar to the child language improves overall quality. They run the same experiment with French as a parent language and a synthetic child language called French' (which is exactly like French except that the vocabulary is shuffled randomly). Compared to an unrelated parent-child pair (French-Uzbek), the study finds a greater BLEU improvement with a related parent (French-French').

Kim et al. (2019) improved upon the transfer learning paradigm presented by Zoph et al. (2016) by proposing three adjustments to make it less dependent on the similarity between parent and child more widely applicable to various languages. One of the main flaws with vanilla transfer learning is that regardless of how similar the parent and the child language-pairs are, natural languages are discrete and have unique vocabularies. Because the mapping between the different vocabularies is arbitrary, when we feed child language inputs into the parent model, the pre-trained encoder weights do not correspond well with the new source language. Kim et al. propose we circumvent this problem by generating cross-lingual word embeddings.[12] Cross-lingual word embeddings keep the vocabularies separate but share their embedding spaces by mapping monolingual embeddings of the parent and child languages to one another. Initializing the child model with cross-lingual embeddings improved over vanilla transfer learning by an average of 3.3 BLEU across five language pairs. Kim et al. also address issues of overfitting[13] and data scarcity in transfer learning by introducing two new techniques. The former technique is to force the parent model not to over optimize to the parent source language by inserting random noise into the training data. The latter technique is to reuse parallel data already used for training the parent model by retaining only the tokens that exist in the child vocabulary and replacing all other tokens with a predefined <unk> token. The study finds 1:2 to be the ideal ratio between non-synthetic and synthetic data. The two techniques improved BLEU over vanilla transfer learning by 0.8 BLEU and 1.5 BLEU respectively.

Massively multilingual models take transfer learning to the next level by training a single model on multiple source languages and multiple target languages. A single multilingual models can translate multiple languages in multiple directions. Training on multiple languages at once increases the amount of information that can be shared between different translation tasks and thus allows the model to leverage data from other languages when learning a low-resource language pair. Aharoni et al. (2019) investigate the possibility of a "universal" NMT which supports up to 102 languages. The study evaluates several multilingual model arrangements trained on 58 languages to and from English and tests on Azerbaijani, Belarusian, Galician and Slovak, all of which are extremely low-resource. Many-to-many models[g] perform most successfully with a 1.82 BLEU improvement averaged across the four language pairs.

5.5.2 Approaches Specific to Morphologically Rich Languages

Translation of agglutinative languages poses novel challenges that are not present for languages with simpler morphology. NMT systems typically translate at the word level and are agnostic to words' internal structures. Because stems and affixes can be combined in numerous and varied ways, the vocabulary size of an agglutinative language is considerable, which can cause memory problems for large language models as well as data sparsity issues. To reduce vocabulary size and create models with a more robust handling of morphology, much of the research on NMT for agglutinative languages focuses on the morpheme segmentation, tokenization, and labeling.

Chimalamarri et al. (2020) focus on the language pair Kannada-Telugu, which is a low-resource pair where both languages feature agglutination. In this study, they divide the vocabulary into classes of words using a Trigrams'n'Tags based part of speech (POS) tagger[14] and then segment morphemes using a statistical morpheme segmentation tool called Morfessor (Virpioja et al. 2013) to identify frequently occurring suffixes. Using the suffixes identified by Morfessor, Chimalamarri et al. run a stemming script to segment out stems and suffixes in a parallel corpus and build a morpheme corpus for both source and target languages by replacing every morphologically complex word with its constituent morphemes. They then train a seq2seq model with a bidirectional[15] encoder LSTM and decoder with attention on the morpheme corpus, using both words and POS tags as features. This model improves 7.52 BLEU over baseline and 7.19 BLEU over Byte Pair Encoding (BPE)[16] tokenization, showing great promise for linguistically informed morpheme segmentation methods.

[g] Many-to-many NMT models are multilingual models that translate from many languages to many other language as opposed to one-to-many or many-to-one models which translate from one language to many and from many languages to one respectively.

One method that has shown promise in agglutinative language translation is the use of a multi-task neural model that jointly learns to perform bi-directional translation and agglutinative language stemming. Pan et al. (2020) trains a stemming task on monolingual source-side data that can be applied jointly with any NMT structure with an encoder-decoder framework.[17] For the language pair Turkish-English, the proposed approach outperformed the baseline NMT model where experimental data was segmented by a morphological segmentation method presented in Pan et al. 2020b, wherein words are segmented into stem unit and a sequence of suffix units, and BPE is applied on the stem unit. Future research will likely involve more creative combinations of approaches as more of the techniques outlined in this thesis become standard. However, it will probably be a while before cutting-edge low-resource NMT techniques are made both accessible and approachable to a broader audience through ample documentation and streamlined open-source toolkits. Furthermore, there are not currently any low-resource NMT approaches that can rival the performance of state-of-the-art high-resource NMT systems. With that in mind, the research outlined in the section is of little relevance to language communities presently seeking to develop effective machine translators for their low-resource languages, but hopefully it will pave the way for greater advances in the future. An ideal NMT solution to Karachay-English translation might involve some combination of the techniques outlined in this section: leveraging monolingual data, transfer learning, and linguistically informed morpheme segmentation.

5.6 Prior Research on Karachay-Balkar Machine Translation

Some basic first steps have already been taken to develop machine translation systems for Karachay-Balkar. Mirzakhalov et al. (2021a) present a large-scale study of Turkic-language machine translation in which they built and evaluated translators for 22 Turkic languages, including Karachay-Balkar. Their Karachay-Balkar-to-English translator (the Mirzakhalov model) was a Transformer model trained using the JoeyNMT framework with 256-dimensional embeddings and hidden layers. The model was tokenized using BPE with a joint vocabulary size of 4k and optimized with the Adam optimizer (Kingma and Ba, 2015). The model was trained using cross-entropy loss and used perplexity[18] as an early stopping metric[19] with a patience of 5 epochs. The model also employed dropout[20](Srivastava et al., 2014) probability of 0.3 in both the encoder and the decoder. The accuracy of the Mirzakhalov model serves as a useful benchmark for evaluating the success of both the neural and rule-based models developed in this thesis, as it leaves plenty of room for improvement. The Mirzakhalov model obtained a BLEU score of just 11.57. Furthermore, the researchers have made their parallel corpora publicly available, which is used as the foundational training data for this thesis.

Karachay-Balkar also has an existing Apertium rule-based morphological transducer with around 90% coverage (Washington et al. 2021), which lays the groundwork for developing a machine translator. No structural transfer rules have been written for Karachay-Balkar specifically, but rules have been written for Kyrgyz, a similar Turkic language, which provide a good starting point (Washington et al. 2021).

6 Methodology

Having surveyed the research relevant to the development of community-driven machine translation, this thesis pivots to evaluate the implementation process and efficacy of two free, and open-source platforms for developing machine translation systems: JoeyNMT and Apertium. I selected JoeyNMT because it was designed specifically with novices in mind, but it achieves performance on standard benchmarks that is comparable to more complex frameworks (Kreutzer et al. 2019). I selected Apertium because the existing Karachay-Balkar morphological transducer provides a strong foundation on which to build an RBMT system without requiring as much effort as starting from scratch.

I compare both platforms not only on their respective BLEU scores, but also on their utility to low-resource language communities. The goal of this thesis is not simply to assess which platform produces a translation system with higher accuracy, but to evaluate how either platform might be used practically to support community efforts. In this thesis, I propose four criteria beyond accuracy on which to evaluate machine translation platforms: efficiency, accessibility, ease of deployment, and interpretability.

6.1 Data

In this thesis I use the eng-krc parallel corpus compiled by Mirzakhalov et al. (2021b), which contains an aligned translation of the New Testament. The corpus is split into 8220 training samples, 250 development samples and 250 test samples. 27 additional sentences of out-of-domain (OOD) testing data were gathered

from riddles, proverbs, and sample text in Seegmiller (2016). I tested OOD data because of the limited domain training data for the NMT model. OOD data is useful to evaluate the success of a model more generally by giving it inputs it has not seen during training.

6.2 Neural Model Specification

Prior to training, I tokenized the data with Sacremoses and applied BPE with a joint vocabulary size of 5k. I trained a transformer-based NMT model using the JoeyNMT framework and fine-tuned the hyperparameters based on Araabi and Monz (2021), with 512-dimensional embeddings and hidden layers. I optimized the model with the Adam optimizer (Kingma and Ba, 2015) and trained over cross-entropy loss[21]. I used used perplexity as an early stopping metric with a patience of 5 epochs and trained the model with a feed-forward dimension of 1024, 2 attention heads, 5 layers, and a batch size of 4096. I also employed activation dropout with probability of 0.3 in the encoder, and 0.1 in the decoder.

All models were trained with GPUs freely available on Google Colab[h], and Apex[i] was used to speed up training.

6.3 Rule-Based Model Specification

We[j] bootstrapped the krc-en language pair using Apertium and the Karachay-Balkar morphological transducer developed in Washington et al. (2021). Because of the linguistic similarity between Kyrgyz and Karachay-Balkar, we adapted structural transfer rules developed for a Kyrgyz to English Apertium translator (Washington et al. 2021). We populated the dictionary with entries from the lexicon of the morphological transducer that had corresponding English translations commented in the code and translations from the glossary of Seegmiller (1996). In total, we added 671 entries to the dictionary. POS tags were pulled naively from Seegmiller (1996) as well, and may or may not have corresponded with the POS tags already present in the transducer.

6.3.1 Limitations

The efforts presented in this thesis are not to be taken as representative of what is possible to achieve with a rule-based translation system. My collaborator and I were limited by the fact that neither of us were proficient in Karachay-Balkar and we did not have access to a native speaker. Working side-by-side with a native speaker would have likely been faster and yielded a more accurate system, but the process would have consisted of more overhead. An in depth discussion of Apertium development workflows is beyond the scope of this thesis, but is worth considering in future work.

6.4 Results

I evaluated the two translation models on 27 OOD sentences as well as a 250 sentence test corpus from the New Testament.

Model	New Testament (BLEU)	OOD (BLEU)
Transformer	10.73	1.14
Apertium	0.79	1.14

7 Discussion

7.1 Quantitative Analysis

Of the models developed in this thesis, the neural model performed better on both in-domain and out-of-domain data. With a difference of 9.59 BLEU, the neural model performed significantly worse on the out-of-domain data than in-domain data. The Apertium system performed better on the out-of-domain data than the in-domain data, but that is not particularly interesting. Out-of-domain is meaningless for RBMT because during development, vocabulary was pulled from the same grammar (Seegmiller 1997) as the out-of-domain test corpus out of necessity, so the out-of-domain data might well be considered in-domain for the Apertium system. Neither translation system is particularly useful in

[h] https://colab.research.google.com/

[i] https://github.com/NVIDIA/apex

[j] The Apertium system was developed in a joint e□ort with Jonathan Washington

its current state. Example 6.1 shows just how badly both models can fail. Whereas the JoeyNMT model overgeneralizes to predict an English translation that is far from the intended meaning, the Apertium system undergeneralizes and outputs untranslated text.

Ex 7.1. ***Source text:*** *Хар тюрлю дерт тутуу , къутуруу , ачыуланыу , къычырыу , аман сёлешиу эмда хар тюрлю аман ниет сизден къорасынла ;*
Translation: *Put out of your life all these things : bad feelings about other people , anger , temper , loud talk , bad talk which hurts other people , and bad feelings which hurt other people .*
JoeyNMT Prediction: *It will be bad for you . It will be bad for you . It will be bad for those who are not Jews . They will not be bad for you . They will be afraid of you .*
Apertium Prediction: *#All various *дерт takes, *къутуруу, *ачыуланыу, shouts, #bad speaks and #all various #bad *ниет from you *къорасынла;*

The BLEU score obtained by the Apertium model is not representative of what is possible with a mature Apertium system. For example, an Apertium Tartar-Kazakh translator achieved 91.79% naive coverage over the New Testament (Salimzyanov et al. 2013).The performance of the RBMT system developed in this thesis was primarily constrained by time and knowledge of the language.

On the other hand, the performance of the JoeyNMT is likely representative of what is possible with a vanilla transformer-based approach. The BLEU score (10.73) of my JoeyNMT model is very similar to the BLEU score (11.57) of the Mirzakhalov model. The modest difference can most likely be ascribed to the fact that I trained and tested my model on slightly different train and test sets than Mirzakhalov et al. (2021a), and with different hyperparameters. However, the hyperparameters I used in this thesis were pulled from a separate study on the optimal hyperparameters for low-resource machine translation (Araabi and Monz 2020), and thus should still be representative of the state-of-the-art.

7.2 Qualitative Analysis

In this thesis I evaluate the Apertium and JoeyNMT platforms on the basis of four additional factors beyond the respective BLEU scores of each model: efficiency, accessibility, ease of deployment, and interpretability. These factors were chosen based on their relevance to the utility of a machine translation tool for community-driven development.

7.2.1 E□ciency

For the purposes of this thesis, e□cient refers only to the human efficiency of the development process. Given two developers with equivalent expertise in NMT and RBMT, the NMT development process is undeniably more expedient. If one has access to a sufficiently large training corpus, developing an accurate RBMT system with Apertium takes more time than developing an accurate NMT system with JoeyNMT. For this reason, one might suggest that a community's time could be better spent annotating parallel corpora to support NMT than developing an RBMT system, but it is important to remember that the size of the parallel corpus required to train effective NMT models is enormous. Furthermore, annotating corpora itself requires specialist knowledge and fluency in at least two languages. Depending on manpower and funding available, RBMT development might still require less time than corpus annotation to achieve comparable results.

All things considered, it is very difficult to say which system is more efficient in the low-resource setting. While simply training an NMT system may be more efficient than developing an RBMT system from scratch, the efficiency of an NMT system is very much limited by the size of the training corpus. The process of developing RBMT is likely more efficient than the process of developing hand annotated corpora to enhance an NMT training set.

7.2.2 Accessibility

For a community to benefit from a tool, they have to be able to use it effectively, so accessibility is an important factor to consider. The intuitiveness of a platform impacts its accessibility. Both Apertium and JoeyNMT are relatively accessible to novices because of their comprehensive documentation. However, "relatively" is the key word here, as neither system is intuitve to the layperson. If one's goal is to train a standard model with default parameters, JoeyNMT requires less specialist knowledge than Apertium. However, fine-tuning a model properly requires knowledge of machine learning that is not outlined in the JoeyNMT documentation. All in all, the two packages are comparably novice-friendly, and ease-of-use is not a compelling reason to favor one over the other. In future work, it would be worthwhile to work

directly with a low-resource community to determine which system is preferable, and how documentation can be improved.

Another factor that impacts the accessibility of a platform is how expensive it is to develop. Because most people do not have access to high-powered computers, the computational intensity of a development process directly impacts its accessibility. Developing an RBMT system typically requires much less computational power than training a neural model. Depending on the size of the training corpus, neural models can take a prohibitively long time to train on CPU. Training on GPU speeds up the process considerably, but many computers are not capable of GPU computation. Though there are online services, such as Google Colab, that provide access to GPU computation, free access is limited. Solely on the basis of computational expense, Apertium is more accessible than JoeyNMT.

7.2.3 Ease of Deployment

In order for a machine translation system to be used by a community at large, it must first be deployed. Therefore, ease of deployment is a prohibiting factor in the usefulness of a system. Apertium has an open-source web framework that can be used to deploy Apertium systems relatively easily[k]. JoeyNMT does not have a similarly localised interface, so any attempt to deploy a machine translation system developed with JoeyNMT would require learning external web-development frameworks. Therefore, Apertium beats JoeyNMT on the ease of deployment criterion.

7.2.4 Interpretability

Interpretability is perhaps the most significant factor in a machine translation system's usefulness to low-resource language communities, because it determines the possibility of improvement. The inner workings of a trained NMT system are not interpretable, and it is currently not possible to open-up a neural model, tweak a few values, and correct a specific error. RBMT systems on the other hand are highly interpretable and allow for errors to be identified and corrected.

My limitations as a non-speaker of Karachay-Balkar are also very relevant here. In developing the RBMT system, I was disadvantaged by not having robust knowledge of the language, but a community driven effort would not be hampered by this issue. Even if a community were to hire an outside consultant with zero language specific knowledge to do the bulk of the Apertium development, the RBMT workflow lends itself to collaboration and a developer could easily crowd-source lexical and grammatical insights from native speakers to iteratively improve on the system.

8 Conclusion

Ultimately, the successes and failures of low-resoruce translation that I've explored in this thesis emphasize that there is still a long way to go. Neither RBMT nor NMT are ideal for community driven low-resource translation. NMT systems are quick to produce and they yield incredible results on large parallel corpora, but they underperform on small parallel corpora. RBMT is not corpus based, so it does not suffer from the same constraints, but it is labor intensive. Both approaches require a level of specialist knowledge that is prohibitive to novices. Assessed on the basis of not only performance, but the criteria of efficiency, accessibility, ease of deployment, and interpretabilty, it is clear that Apertium supports community driven machine translation more effectively than JoeyNMT.

[k] https://github.com/apertium/apertium-html-tools/

Notes

[1]The BiLingual Evaluation Understudy (BLEU) score (Papineni et al., 2002) is a metric for evaluating the quality of translations produced by machine translation systems. BLEU is a modified precision score that compares series of tokens in a candidate translation to series of tokens in the reference translation. The series of tokens is an called n-grams (where n = the number of tokens in the series). Most often n=4 is used for BLEU. BLEU counts how many n-grams in the candidate translation appear in the reference translation but puts a cap on the counts of each n-gram corresponding to the maximum number of times that n-gram appears in the reference translation. For example, if the n-gram "the cat" appears 4 times in the candidate but only two times in the reference, then only two repetitions of the n-gram "the cat" in the candidate will be counted as also appearing in the reference.

[2]A sentence embedding is a representation of text in a fixed-dimensional vector. Each entry or set of entries in the vector corresponds to a certain feature of the text. Therefore, sentences that are similar to one another should have similar embeddings. Though there is research on the subject, it is currently very di□cult to know exactly which features correspond to which vector indices. In NMT, sentence embeddings are generated by the encoder, which learns to to represent text as vectors during training.

[3]Neural networks were named as such because they are modeled after the neurons in brain. The simplest unit of a neural network is called a neuron or node. Neural networks consist of interconnected nodes arranged in layers called hidden layers. Nodes can be thought of as miniature logistic regression models that each come with a weight term, a bias term, and a non-linear activation function. The bias term is a constant used to shift the activation function slightly in accordance with your data. The weight term is a parameter that is continually updated by the model during training to optimize performance. A node processes an input by taking the product of that input and the weight and then adding the bias term. If the result is above a certain threshold, the node is activated, and sends information to the next set of nodes. The specific threshold is dependent on the activation function. Di□erent activation functions have di□erent advantages and disadvantages, but in practice ReLU is often the activation function of choice because empirically it seems to work well.

$$ReLU(x) = \begin{pmatrix} 0 & x & 0 \\ x & x & 0 \end{pmatrix} \quad (1)$$

A neural network takes an input in the form of a vector, passes information through its hidden layers, and is meant to output a probability distribution corresponding to the likelihood that that vector belongs to each of a set of predefined classes. Neural networks are trained in cycles called epochs. At the end of each epoch, the weights associated with each node are updated in such a way that the predicted probability distribution should move closer to the actual probability distribution of the training data. At test time, the model is fed an input vector and the class that is assigned the highest probability is taken to be the model's classification for that vector.

[4]In a feed-forward neural network, each node in each hidden layer is connect directly to each node in the following layer as seen in figure 3.

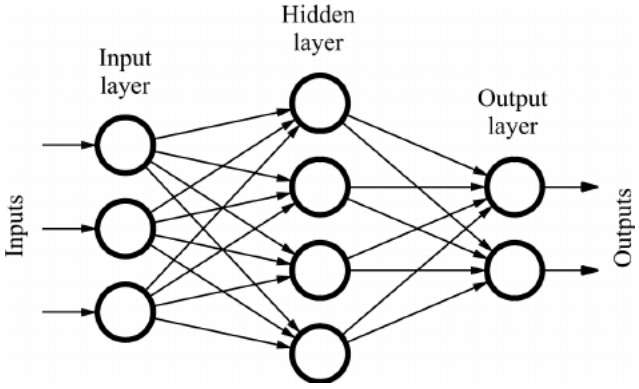

Figure 3: Feed-forward Neural Network (Quiza & Davim 2011: 4)

The equations below model a feed-forward neural network with two hidden layers in matrix notation. $W^{(n)}$ is the weight vector associated with layer n, $b^{(n)}$ is the bias vector associated with layer n, x is the input vector, and $f()$ is the activation function (see endnote 3 for more details).

Hidden layer 1:

$$h_1 = f(W^{(1)}x + b^{(1)}) \quad (2)$$

Hidden layer 2:

$$h_2 = f(W^{(2)}h_1 + b^{(2)}) \quad (3)$$

Output layer:

$$y = (W^{(o)} h_2 + b^{(o)}) \quad (4)$$

[5] RNNs are similar to feed-forward neural networks except they have a feedback loop that allows node outputs to be passed back in as input. As seen in figure 5, RNNs can be unrolled conceptually when we consider each pass through the feedback loop as a time-step forward. RNNs also di□er from feed-forward neural networks in that they make use of an additional input vector called the hidden state vector (h). At each time-step, an RNN takes an input vector x and a hidden state vector h, outputs a vector y, and updates the hidden state vector. The equations representing what occurs at each time-step are given below, where equation 5 represents how a hidden state is updated and equation 6 represents how the output is generated.

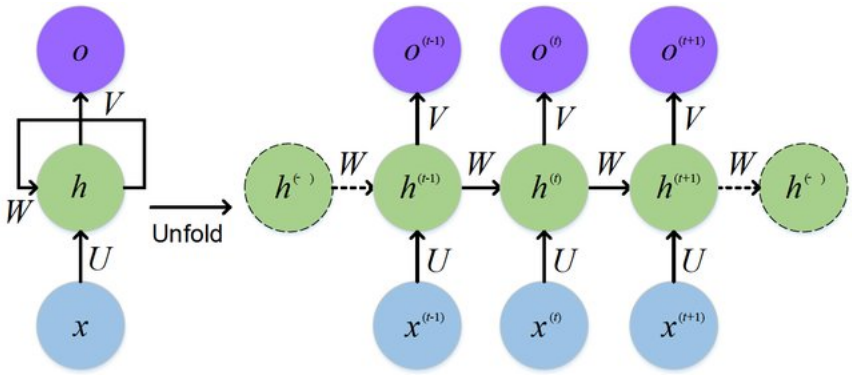

Figure 4: Unrolled RNN (Feng et al., 2017)

$$h_t = g(V x_t + U h_t \quad 1 + b_h) \tag{5}$$

$$y_t = O h_t \tag{6}$$

In equation 5, V and U are weight vectors that are updated during training. Generally speaking, U is representative of how interested the node is in the features of the input while v is representative of how interested the node is in the features of the hidden state. $g()$ is a non-linearity, and b_h is a bias term. In equation 6, O is yet another weight vector that is updated during training.

[6] The vanishing gradient problem is a byproduct of the RNN training process, specifically the backpropagation through time (BPTT) algorithm (Mozer, 1995; Robinson & Fallside, 1987; Werbos, 1988). Through BPTT, a model tries to iteratively approach the minimum of the loss function. This achieved by computing the gradient of the loss function with respect to each weight vector at the end of each training cycle (epoch) and adjusting the weight vectors accordingly. Taking the gradient of loss with respect to U (refer to endnote 5) can quickly get complicated because each hidden state vector relies on the hidden state before it. An example of the process for updating hidden states and calculating the gradient of the loss function with respect to V is given below.

$$h_1 = g(V x_1 + U h_0 + b_h)$$

$$h_2 = g(V x_1 + U h_1 + b_h)$$

$$h_3 = g(V x_3 + U h_2 + b_h)$$

$$\frac{@L_3}{@V} = \frac{@L_3}{@h_3}\frac{@h_3}{@V} + \frac{@L_3}{@h_3}\frac{@h_3}{@h_2}\frac{@h_2}{@V} + \frac{@L_3}{@h_3}\frac{@h_3}{@h_2}\frac{@h_2}{@h_1}\frac{@h_1}{@V}$$

The multiplicative nature of the chain-rule (which is used to calculate the gradient) means that, given a long input sequence, the hidden states associated with the earlier context in the sequence (h_1 for example) will have less of a contribution to the gradient (a lot of small fractional numbers multiplied by one another yield an even smaller number). Thus, the vanishing gradient problem presents a problem to RNN based machine translation models that are given long sentences to translate. There is also a similar problem with RNNs called the exploding gradient problem where basically the same issue occurs but the gradient increases exponentially because the a lot of large numbers are multiplied together during the process of finding the gradient. Both vanishing and exploding gradients inhibit learning.

[7] An LSTM is a kind of RNN that is capable of learning long-term dependencies. The goal of the LSTM is to avoid the vanishing/exploding gradient problem that RNNs face by replacing some of the multiplicative relationships in some hidden states with additive ones. The key di□erence between a vanilla RNN and an LSTM is that in addition to a hidden state (h) and input (x) vectors, an LSTM also have have a hidden state called textbfcell memory (c) and three gates: the input gate (i), the forget gate (f), and the output gate (o). The content of the cell memory is controlled by the gates. The input gate is controls how much the current input influences memory, the forget gate controls how much to let previous memory influence future memory, and the output gate controls what part of memory should go into the hidden state. The formulas representing each gate and their impact on c and h are given below, where is the sigmoid function (refer to endnote 5 for other variables).

$$f_t = (V_f x_t + U_f h_t \quad 1 + b_f) \tag{7}$$

$$i_t = (V_i x_t + U_i h_t \quad 1 + b_i) \tag{8}$$

$$o_t = (V_o x_t + U_o h_t \quad 1 + b_o) \tag{9}$$

$$C^0 = \tanh(V_c h_t \quad 1 + U_c x_t + b_c) \tag{10}$$

$$C_t = f_t \quad C_t \quad 1 + i_t \quad C^0_t \quad 1 \tag{11}$$

$$h_t = o_t \quad \tanh(C_t) \tag{12}$$

[8] In the encoder-decoder framework, the encoder RNN will read a sequence of input vectors $x = (x_1; :::; x_n)$ and produce an output vector. Equation 13 models the generation of hidden state vectors where h_t is a hidden state vector at time t, x_t is the input at time t, $h_t \quad 1$ is the hidden state vector at time $t \quad 1$, and f is a non-linear function.

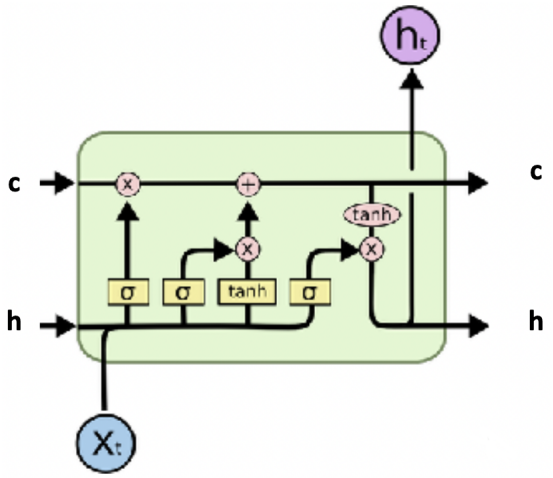

Figure 5: LSTM (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

$$h_t = f(x_t; h_{t-1}) \tag{13}$$

Equation 14 represents how hidden states are interpreted by the RNN to generate the vector c. The function q is a non-linear function and T_x is the total number of time steps.

$$c = q(h_1; h_n) \tag{14}$$

The decoder predicts each word in a sequence given the context vector c and all previously predicted words. The model defines a probability of a translation y as given in equation 14

$$p(y) = \prod_{t=1}^{T_y} p(y_t | y_1; \ldots; y_{t-1}; c) \tag{15}$$

where $y = (y_1; \ldots; y_{T_y})$.

[9] Attention allows a decoder to keep track of hidden states from each encoder node at each time step and then make predictions based on which one is more informative. Attention is calculated using three vectors that are associated with an encoder's input vector: a query vector (Q), a key vector (K), and a value vector (V). The attention function maps the query and key-value pairs to an output vector. One form of attention that is commonly used is called scaled dot-product attention. Equation 16 represents scaled dot-product attention.

$$Attention(Q; K; V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{16}$$

[10] Transformers architecture, as illustrated in 10, consist of stacks of encoders and decoders. Each encoder consists of two layers, a self-attention layer and a feed-forward neural network. Each decoder also has a self-attention layer, and a feed-forward layer, but in between those layers, they also have an encoder-decoder attention layer. Self-attention is an attention mechanism that relates several different positions of the same input sequence and allows an encoder/decoder to capture the internal structure of an input sequence. In self-attention, queries, keys, and values all come from the same input sequence. Often scaled dot-product attention is used. Rather than performing a single attention function, Transformers make use of multi-head attention, which involves linearly projecting and scaling down the queries, keys and values h times. Each set of $Q; K$ and V is known as an attention head. On each attention head, Transformers perform the attention function in parallel, and then concatenate the outputs. The size of the concatenated vector is then multiplied by a learned matrix W_o to be fed into the next sub-layer. Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions (Vaswani et al., 2017). Self-attention works similarly in the decoder stack, except it is modified so that the decoder cannot attend to future words. This is called masked multi-head attention The encoder-decoder attention process works similarly to multi-headed self-attention, but it creates queries from the output of the sub-layer below but takes keys and values from the output of the encoder stack.

[11] The logic behind this approach is that the encoder-decoder framework already condition the probability distribution of the next target word on the previous target words, so in cases where the source is empty or uninformative, the model can be forced to learn to rely on previous target words for its prediction.

[12] The process Kim et al. (2019) give for generating cross-lingal word embeddings is outlined in five steps.

1. Learn monolingual embeddings of the child language E^{mono}_{child}
2. Extract source embedding E^{source}_{parent} from a pre-trained parent model
3. Learn the cross-lingual linear mapping W between E^{mono}_{child} E^{source}_{parent} by minimizing the objective below:

$$\sum_{(f;f') \in S} k\, E^{mono}_{child}(f) \quad E^{source}_{parent}(f')\, k_2 \tag{17}$$

4. Replace the source embedding of the parent model with the learned cross lingual embedding.
5. Initialize the child model with the parent model and train.

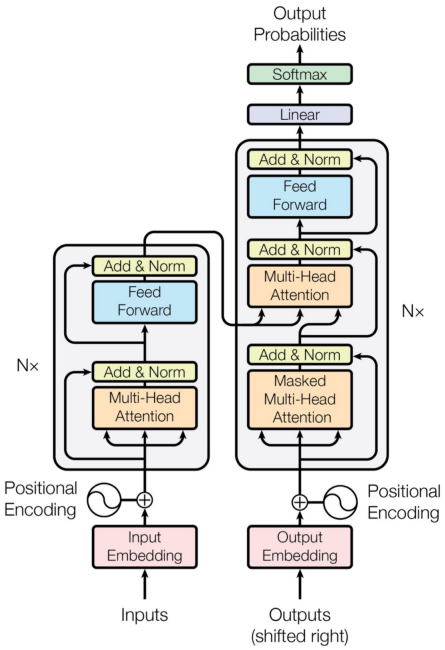

Figure 6: The Transformer Architecture (Vaswani et al., 2018)

[13]Overfitting is an error in machine learning where a model learns the training data too well. Overfitting inhibits a model's ability to generalize because it has learned the specific details and noise of the training data so well that it expects to see the same noise in the test data. The problem is that a training dataset is never a perfect representation of the actual distribution of the data in the wild.

[14]Trigrams'n'Tags (TNT) is a statistical POS tagger that relies on second-order Hidden Markov Models (HMMs). An HMM is a probabilistic sequence model that assigns a labels to units in a sequence. The exact implementation details of HMMs are not relevant to this thesis. TNT is not optimized for one particular language, but is optimized for training on a large variety of corpora.

[15]A bidirectional LSTM is the concatenation of two independent LSTMs. An input sequence is fed in normally for one network and in reverse order for another. At each time step, the outputs of the two networks are concatenated or summed. This structure is useful because it allows the networks to consider both backward and forward information about the sequence at each time step (that is, to consider the context both preceding and succeeding the word or character output). A diagram of a bidirectional LSTM is shown in figure 7.

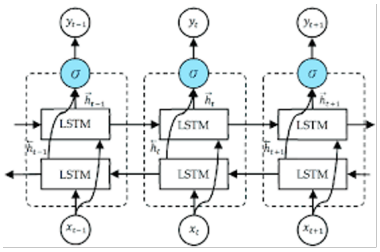

Figure 7: Bidirectional LSTM (Li et al., 2020)

[16]Byte Pair Encoding (BPE) is a sub-word tokenization scheme that addresses the rare word problem. The rare word problem is an issue where certain words do not appear frequently enough in the training data for an NMT system to properly generate word-level embeddings. Translation systems for all languages su□er from the rare word problem, but because morphologically complex languages have enormous vocabularies, their training corpora have even more rare words. Another benefit of BPE is that it prevents the model from encountering "unknown" words during testing. As long as each character in the language's alphabet is represented in the training data, the model will at the very least be able to represent an unknown word by its characters. The end-product of BPE is a vocabulary that is populated with frequently occurring subwords rather than whole words. It is called byte-pair encoding because sub-words are referred to as "bytes". At each iteration of the algorithm, the most commonly co-occurring pair of bytes is merged. Initially all bytes are single characters. The algorithm can be summarized as follows:

- Each word in the corpus is represented as a collection of characters and an end of word token.
- Each byte pair in the corpus is counted
- Each occurrence of the most frequent pair is merged and the byte is added to the vocabulary

- Steps two through three are repeated until the desired vocabulary size is achieved or the desired number of merge operations have been completed.

[17] The model proposed in Pan et al. (2020b) adds a artificial start token before each input to the transformer to signify whether the desired task is a stemming task or a translation task. The architecture is given in figure ??.

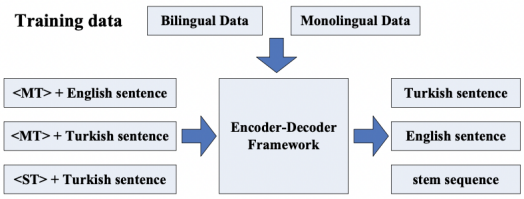

Figure 8: The architecture of the multi-task neural model that jointly learns to perform bi-directional translation between Turkish and English, and stemming for Turkish sentence. (Pan et al., 2020)

[18] Perplexity is the inverse probability of the test set inverse probability that the model predicts the test set normalized by the number of words. Minimizing perplexity is the same as maximizing probability. Equation 18 is the equation for perplexity, where W is the test sequence, w_n is a word in the test sequence, and N is the number of words in the sequence.

$$PP(W) = P(w_1 w_2 \cdots w_N)^{\frac{1}{N}} \tag{18}$$

[19] An early stopping metric is a metric used to halt training once the performance of the model stops improving for an arbitrary number of training epochs. Patience refers to how many epochs we would like to see no improvement before halting.

[20] Dropout is a method used to discourage a neural network from overfitting. During each epoch, nodes are randomly ignored or "dropped out". Conceptually, this technique works because it inhibits a phenomenon called co-adaptation. Co-adaptation is a phenomenon where two or more nodes in a neural network have highly correlated behavior, which is undesirable because it interferes with a model's ability to generalize. Dropout combats co-adaptation by forcing nodes to learn independently because they are unable to rely on nodes that have been dropped out during a given epoch. When we say that a model has a dropout probability of x it means that for any node in the model, there is a probability of x that it will be randomly omitted at the beginning of each epoch. In a Transformer model, it is sub-layers that are dropped out rather than nodes.

[21] Cross-Entropy calculates the difference between two probability distributions. It builds off the concept of entropy, which is a measure of how skewed a probability distribution is. A skewed distribution will have low entropy whereas a balanced distribution will have high entropy. Equation 19 gives the equation for entropy where n is number of possible categories and $p(x_i)$ is the probability that an input x will belong to class i.

$$\sum_{i=1}^{n} p(x_i) \log(p(x_i)) \tag{19}$$

Cross-entropy on the other hand represents the number of bits required to encode data from a one distribution in another distribution. The equation for cross entropy, equation 20, looks rather similar to the equation for entropy.

$$\sum_{i=1}^{n} p(x_i) \log(q(x_i)) \tag{20}$$

The only difference is that p represents the true distribution and q represents the predicted distribution. The more different the distributions, the higher the cross-entropy, which is why it is used as a loss function. Because the goal is for the predicted distribution to be relatively close to the true distribution of the training data, machine learning models aim to minimize cross-entropy loss during training.

References

[1] Aharoni, Roee, Melvin Johnson & Orhan Firat. 2019. Massively Multilingual Neural Machine Translation. 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), 3874–3884. Minneapolis, Minnesota: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1388. https://aclanthology.org/N19-1388 (23 October, 2021).

[2] Araabi, Ali & Christof Monz. 2020. Optimizing Transformer for Low-Resource Neural Machine Translation. In Proceedings of the 28th International Conference on Computational Linguistics, 3429–3435. Barcelona, Spain (Online): International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.304. https://aclanthology.org/2020.coling-main.304 (11 November, 2021).

[3] Arivazhagan, N., Ankur Bapna, Orhan Firat, Dmitry Lepikhin, M. Johnson, M. Krikun, M. Chen, et al. 2019. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. undefined. https://www.semanticscholar.org/paper/Multilingual-Neural-Machine-Translation-Dabre-Chu/b40bcefd215679a36b51ddf6b073aa60d43a5276 (7 October, 2021).

[4] Bahdanau, Dzmitry, Kyunghyun Cho & Yoshua Bengio. 2016. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473 [cs, stat]. http://arxiv.org/abs/1409.0473 (28 October, 2021).

[5] Bender, Emily M., Timnit Gebru, Angelina McMillan-Major & Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 610–623. Virtual Event Canada: ACM. https://doi.org/10.1145/3442188.3445922 https://dl.acm.org/doi/10.1145/3442188.3445922 (13 September, 2021).

[6] Bird, Steven. 2009. Last Words: Natural Language Processing and Linguistic Fieldwork. Computational Linguistics 35(3). 469–474. https://doi.org/10.1162/coli.35.3.469.

[7] Bird, Steven. 2020. Decolonising Speech and Language Technology. https://doi.org/10.18653/v1/2020.coling-main.313.

[8] Blasi, Damián, Antonios Anastasopoulos & Graham Neubig. 2021. Systematic Inequalities in Language Technology Performance across the World's Languages. arXiv:2110.06733 [cs]. http://arxiv.org/abs/2110.06733 (19 October, 2021).

[9] Chimalamarri, Santwana & Dinkar Sitaram. 2021. Linguistically enhanced word segmentation for better neural machine translation of low resource agglutinative languages. International Journal of Speech Technology. https://doi.org/10.1007/s10772-021-09865-5. https://link.springer.com/10.1007/s10772-021-09865-5 (22 October, 2021).

[10] Cieri, Christopher, Mike Maxwell, Stephanie Strassel & Jennifer Tracey. 2016. Selection Criteria for Low Resource Language Programs. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 4543–4549. Portorož, Slovenia: European Language Resources Association (ELRA). https://aclanthology.org/L16-1720 (29 October, 2021).

[11] Dryer, Matthew S. & Martin Haspelmath (eds.). 2013. WALS Online. Leipzig: Max Planck Institute for Evolutionary Anthropology. https://wals.info/ (17 November, 2021).

[12] Eberhard, David M., Gary F. Simons, & Charles D. Fennig (eds.). 2021. Ethnologue: Languages of the World. Twenty-fourth edition. Dallas, SIL International. Online version: http://www.ethnologue.com.

[13] Feng, Weijiang, Naiyang Guan, Yuan Li, Xiang Zhang Zhigang Luo. 2017. Audio visual speech recognition with multimodal recurrent neural networks. https://doi.org/10.1109/IJCNN.2017.7965918.

[14] Forcada, Mikel L., Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez & Francis M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. Machine Translation 25(2). 127–144. https://doi.org/10.1007/s10590-011-9090-0.

[15] Goyal, Naman, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman & Angela Fan. 2021. The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. arXiv:2106.03193 [cs]. http://arxiv.org/abs/2106.03193 (7 October, 2021).

[16] Haddow, Barry, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl & Alexandra Birch. 2021. Survey of Low-Resource Machine Translation. arXiv:2109.00486 [cs]. http://arxiv.org/abs/2109.00486 (13 September, 2021).

[17] Hämäläinen, Mika. 2021. Endangered Languages are not Low-Resourced! https://helda.helsinki.fi/handle/10138/327865 (16 November, 2021).

[18] Hochreiter, Sepp & Jürgen Schmidhuber. 1997. Long Short-term Memory. Neural computation 9. 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

[19] Joshi, Pratik, Sebastin Santy, Amar Budhiraja, Kalika Bali & Monojit Choudhury. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 6282–6293. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.560. https://aclanthology.org/2020.acl-main.560 (7 October, 2021).

[20] Kim, Yunsu, Yingbo Gao & Hermann Ney. 2019. Effective Cross-lingual Transfer of Neural Machine Translation Models without Shared Vocabularies. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 1246–1257. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1120. https://aclanthology.org/P19-1120 (21 October, 2021).

[21] Koehn, Philipp & Rebecca Knowles. 2017. Six Challenges for Neural Machine Translation. In Proceedings of the First Workshop on Neural Machine Translation, 28–39. Vancouver: Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-3204. https://aclanthology.org/W17-3204 (29 September, 2021).

[22] Kreutzer, Julia, Joost Bastings & Stefan Riezler. 2019. Joey NMT: A Minimalist NMT Toolkit for Novices. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, 109–114. Hong Kong, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-3019. https://www.aclweb.org/anthology/D19-3019 (17 November, 2021).

[23] Lackaff, Derek & William Moner. 2016. Local languages, global networks: Mobile design for minority language users. https://doi.org/10.1145/2987592.2987612.

[24] Lewis, Melvyn & Gary Simons. 2010. Assessing endangerment: Expanding Fishman's GIDS. Revue Roumaine de Linguistique 55. https://doi.org/10.1017/CBO9780511783364.003.

[25] Li, Yunghui, Latifa Nabila Harfiya, Kartika Purwandari & Yue-Der Lin. 2020. Real-Time Cuffless Continuous Blood Pressure Estimation Using Deep Learning Model. Sensors 20. https://doi.org/10.3390/s20195606.

[26] Mozer, Michael. 1995. A Focused Backpropagation Algorithm for Temporal Pattern Recognition. Complex Systems 3.

[27] Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman & Aram Galstyan. 2019. A Survey on Bias and Fairness in Machine Learning. arXiv:1908.09635 [cs]. http://arxiv.org/abs/1908.09635 (13 September, 2021).

[28] Mirzakhalov, Jamshidbek, Anoop Babu, Duygu Ataman, Sherzod Kariev, Francis Tyers, Otabek Abduraufov, Mammad Hajili, et al. 2021. A Large-Scale Study of Machine Translation in the Turkic Languages. arXiv:2109.04593 [cs]. http://arxiv.org/abs/2109.04593 (16 September, 2021).

[29] Mirzakhalov, Jamshidbek, Anoop Babu, Aigiz Kunafin, Ahsan Wahab, Behzod Moydinboyev, Sardana Ivanova, Mokhiyakhon Uzokova, et al. 2021. Evaluating Multiway Multilingual NMT in the Turkic Languages. arXiv:2109.06262 [cs]. http://arxiv.org/abs/2109.06262 (16 September, 2021).

[30] Pan, Yirong, Xiao Li, Yating Yang & Rui Dong. 2020a. Morphological Word Segmentation on Agglutinative Languages for Neural Machine Translation. arXiv:2001.01589 [cs]. http://arxiv.org/abs/2001.01589 (18 November, 2021).

[31] Pan, Yirong, Xiao Li, Yating Yang & Rui Dong. 2020b. Multi-Task Neural Model for Agglutinative Language Translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 103–110. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-srw.15. https://aclanthology.org/2020.acl-srw.15 (16 September, 2021).

[32] Papineni, Kishore, Salim Roukos, Todd Ward & Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. https://doi.org/10.3115/1073083.1073135. https://aclanthology.org/P02-1040 (8 October, 2021).

[33] Prates, Marcelo O. R., Pedro H. Avelar & Luís C. Lamb. 2020. Assessing gender bias in machine translation: a case study with Google Translate. Neural Computing and Applications 32(10). 6363–6381. https://doi.org/10.1007/s00521-019-04144-6.

[34] Quiza, Ramon & J. Davim. 2011. Computational Methods and Optimization. In Machining of Hard Materials, 177–208. https://doi.org/10.1007/978-1-84996-450-0.

[35] Reddy, Siva & Serge Sharoff. 2011. Cross Language POS Taggers (and other Tools) for Indian Languages: An Experiment with Kannada using Telugu Resources. In Proceedings of the Fifth International Workshop On Cross Lingual Information Access, 11–19. Chiang Mai, Thailand: Asian Federation of Natural Language Processing. https://aclanthology.org/W11-3603 (18 November, 2021).

[36] Robinson, A. J. & Fallside, F. (1987). The utility driven dynamic error propagation network (Technical report). Cambridge University, Engineering Department. CUED/F-INFENG/TR.1.

[37] Romaine, Suzanne. 2007. Preserving Endangered Languages. Language and Linguistics Compass 1(1–2). 115–132. https://doi.org/10.1111/j.1749-818X.2007.00004.x.

[38] Romero, Sergio. 2016. Bill Gates speaks K'ichee'! The corporatization of linguistic revitalization in Guatemala. Language & Communication 47. 154–166. https://doi.org/10.1016/j.langcom.2015.08.001.

[39] Salimzyanov, Ilnar, Jonathan Washington & Francis Tyers. 2013. A Free/Open-source Kazakh-Tatar Machine Translation System. In Proceedings of Machine Translation Summit XIV: Papers. Nice, France. https://aclanthology.org/2013.mtsummit-papers.22 (26 October, 2021).

[40] Savoldi, Beatrice, Marco Gaido, Luisa Bentivogli, Matteo Negri & Marco Turchi. 2021. Gender Bias in Machine Translation. arXiv:2104.06001 [cs]. http://arxiv.org/abs/2104.06001 (17 November, 2021).

[41] Sennrich, Rico, Barry Haddow & Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. arXiv:1511.06709 [cs]. http://arxiv.org/abs/1511.06709 (18 November, 2021).

[42] Sennrich, Rico & Biao Zhang. 2019. Revisiting Low-Resource Neural Machine Translation: A Case Study. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 211–221. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1021. https://aclanthology.org/P19-1021 (29 October, 2021).

[43] Stanovsky, Gabriel, Noah A. Smith & Luke Zettlemoyer. 2019. Evaluating Gender Bias in Machine Translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 1679–1684. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1164. https://aclanthology.org/P19-1164 (8 October, 2021).

[44] Sutskever, Ilya, Oriol Vinyals & Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. arXiv:1409.3215 [cs]. http://arxiv.org/abs/1409.3215 (28 October, 2021).

[45] Tsarfaty, Reut, Dan Bareket, Stav Klein & Amit Seker. 2020. From SPMRL to NMRL: What Did We Learn (and Unlearn) in a Decade of Parsing Morphologically-Rich Languages (MRLs)? In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 7396–7408. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.660. https://aclanthology.org/2020.acl-main.660 (19 October, 2021).

[46] Tukeyev, U., A. Karibayeva & Z h. Zhumanov. 2020. Morphological segmentation method for Turkic language neural machine translation. (Ed.) Duc Pham. Cogent Engineering. Cogent OA 7(1). 1856500. https://doi.org/10.1080/23311916.2020.1856500.

[47] Vanmassenhove, Eva, Dimitar Shterionov & Andy Way. 2019. Lost in Translation: Loss and Decay of Linguistic Richness in Machine Translation. In Proceedings of Machine Translation Summit XVII: Research Track, 222–232. Dublin, Ireland: European Association for Machine Translation. https://aclanthology.org/W19-6622 (8 October, 2021).

[48] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser & Illia Polosukhin. 2017. Attention is All you Need. In Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (26 October, 2021).

[49] Wang, Rui, Xu Tan, Renqian Luo, Tao Qin & Tie-Yan Liu. 2021. A Survey on Low-Resource Neural Machine Translation. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, 4636–4643. Montreal, Canada: International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2021/629. https://www.ijcai.org/proceedings/2021/629 (17 September, 2021).

[50] Washington, Jonathan, Ilnar Salimzianov, Francis M Tyers, Memduh Gökırmak, Sardana Ivanova & Oğuzhan Kuyrukçu. International Conference on Turkic Language Processing (TURKLANG 2019). 28.

[51] Werbos, Paul J. 1988. Generalization of backpropagation with application to a recurrent gas market model. https://doi.org/10.1016/0893-6080(88)90007-x. (10 December, 2021).

[52] Xia, Mengzhou, Guoqing Zheng, Subhabrata Mukherjee, Milad Shokouhi, Graham Neubig & Ahmed Hassan Awadallah. 2021. MetaXL: Meta Representation Transformation for Low-resource Cross-lingual Learning. arXiv:2104.07908 [cs]. http://arxiv.org/abs/2104.07908 (16 September, 2021).

[53] Zheng, Zaixiang, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai & Jiajun Chen. 2020. Mirror-Generative Neural Machine Translation. International Conference on Learning Representations (ICLR 2020).

[54] Zoph, Barret, Deniz Yuret, Jonathan May & Kevin Knight. 2016. Transfer Learning for Low-Resource Neural Machine Translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 1568–1575. Austin, Texas: Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1163. https://aclanthology.org/D16-1163 (29 October, 2021). [1412.6980] Adam: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980 (18 November, 2021a). isbn9789526055015.pdf. https://aaltodoc.aalto.fi/bitstream/handle/123456789/11836/isbn9789526055015.pdf (18 November, 2021b).