CIS 530 Final Project: Low-Resource Machine Translation of Uyghur

Efe Ayhan
efeayhan@seas.upenn.edu

Francesca Marini
fmarini@seas.upenn.edu

Abstract

The machine translation (MT) of low-resource languages is one of the most challenging problems in modern natural language processing (NLP). Recently, neural machine translation (NMT) models have become increasingly more effective at the MT task. However, in the low-resource setting, due to the lack of robust parallel training data, developing translation models remains a critical obstacle. In this term project, we build NMT models which translate between Uyghur and English, as well as models which translate from English to Uyghur. We built sequence-to-sequence models for this task, and we attempted several extensions to this baseline. We found that incorporating romanized Uyghur text or the addition of training data in a related language generally did not improve the performance of our models. However, modifying our development set to include a combination of seen and unseen data improved the performance of our models on our completely unseen test set. We attain BLEU scores of 25.15 with our best Uyghur-English model and 23.41 with our best English-Uyghur model.

1 Introduction

1.1 Machine Translation

In Natural Language Processing (NLP), the task of Machine Translation (MT) involves developing models that can translate text in one source language into another target language. Figure 1 provides a visual representation of the MT task.

Low-resource languages are those languages which have very little (and often poor-quality) data available. In addition, access to native speakers of the language for the purposes of building useful corpora is often scarce as well. The task of machine translation on low-resource languages is an especially challenging problem because generally MT models are either trained using parallel text documents (but with low-resource languages, it is even more difficult to obtain parallel text documents for this purpose), or by some other means that involves a deep, expert understanding of the languages in question (which is also challenging since people who know both the source and target languages are difficult to find).

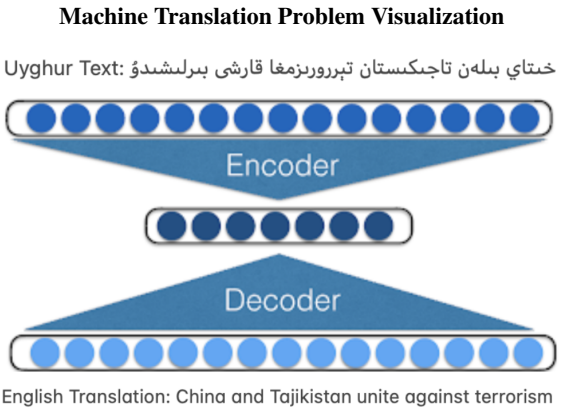

Figure 1: In machine translation (MT), the input to the model is text in one natural language, and the output is translated text in another natural language, capturing the same meaning as the original text. The image above illustrates both how the task works, as well as provides a visualization of a sequence-to-sequence model translating Uyghur into English.

In recent years, the MT task has seen strong improvements in performance due to more powerful models, particularly large, neural models. Generally, these types of models rely on vast amounts of training data to be effective; however, because low-resource languages are less accessible, the difficulty of the problem is further compounded.

Example Uyghur Text

Figure 2: An example of the Uyghur word for "alphabet" text in all three official scripts.

1.2 Uyghur

Uyghur is a low-resource language spoken by around 13.5 million people, most of whom live in the Xinjiang region of China. It is a Turkic language, whose closest related, higher-resource language is Uzbek. It has three official alphabets and can therefore be written in Arabic, Cyrillic, and Latin scripts. Figure 2 illustrates an example of some Uyghur text.

1.3 Motivation

In general, one might want to be able to translate from one natural language to another for many reasons. These include facilitating the verbal communication between parties, sharing literature and other creative cultural arts, and exchanging information between people of various cultures. This is especially useful with respect to low-resource languages, since people who are fluent speakers of both the given low-resource language and another high-resource language are often difficult to find. In addition, many low-resource languages are dying out, so the ability to translate documents written in the original language is crucial to preserving those cultures.

We decided to focus on the Uyghur language because having a means of translating text from Uyghur to English, and from English to Uyghur, would be extremely useful, especially given the ongoing genocide which has been occurring against the Uyghur people for several years. Having such a mechanism would allow for the communication between Uyghurs and other people who speak high-resource languages, and it would also aid in the preservation and sharing of a culture which is currently threatened. In addition, both of the authors have a general interest in linguistics, particularly with regard to low-resource languages. One of the authors is also a native Turkish speaker, so selecting a Turkic language was also of interest for that reason.

We chose to work on the MT task because neither of us had experience with this task in the past, and we wanted to learn more about it while working with an interesting and challenging language.

2 Brief Literature Review

2.1 Historical Approaches to MT

Early approaches to the MT task involved developing rule-based models. A rule-based system requires experts knowledge about the source and the target language to develop syntactic, semantic and morphological rules to achieve the translation. Such models are therefore very language specific and operate in a pipelined format. One benefit of such models is that they do not require the use of parallel text, but this is because they require a lot of expertise to construct. The next types of MT models implemented historically utilized statistical methods. This approach uses statistical models based on the analysis of bilingual text corpora, and it operates in three steps: (1) language model; (2) translation model; (3) word-order determiner. It has the advantage of being less-language specific, but when the source and target languages are very different (especially syntactically), performance suffers. Most recently, research has shifted into developing neural MT (NMT) models, using an encoder-decoder structure, often with an attention mechanism. Such models require parallel corpora as well. The encoder and decoder models traditionally have been layers like bi-LSTMs, or more recently, transformer models.[1] We will discuss neural methods in more detail later in the paper.

2.2 Low-Resource MT

Setiawan et al. 2018 developed a system for low-resource MT in Uyghur for the 2016 LoReHLT evaluation. The authors of this paper first underwent a data selection process. They then were able to interact with a native Uyghur speaker in order to aid in their interaction with the low-resource data. The authors then performed a morphological segmentation of the Uyghur data as a part of the pre-processing. Then, they attempted to developed

[1] http://towardsdatascience.com/machine-translation-a-short-overview-91343ff39c9f

a neural MT (NMT) model, and due to the low-resource setting, they employed transfer learning from Uzbek-English parallel data and data augmentation by including Uzbek data in the training set. They fine-tuned their results further through better-informed data selection, and research into specific task-relevant terms in Uyghur. Their best model attained a BLEU score of 33.80 on this task.

To consider more recent approaches to low-resource MT, we examine Guzmán et al. 2019. In this paper, the authors are developing NMT models for Nepali-English and Sinhala-English. The authors built their own datasets for this task. They then built models for the task in four different settings: unsupervised, semi-supervised, weakly supervised, and supervised learning settings. They also incorporated related high resource languages into their models. They considered both phrase-based statistical MT (PBSMT) and neural MT (NMT) model types. The authors found that the NMT transformer model that they employed outperformed the PBSMT model in the supervised setting, and that the NMT model was generally better at the task, though BLEU scores were overall quite low. Their best models achieved a BLEU score of 21.5 on the Nepali-English task in the semi-supervised setting, 8.8 on English-Nepali in the same setting, 15.1 on the Sinhala-English in the semi-supervised setting, and 6.5 on English-Sinhala in the same setting. While these scores are very low, they serve to illustrate the difficulty of working with low-resource languages in this task as well as working with a self-generated dataset. The authors also touch on the fact that there are negative elements of using BLEU score as a metric in the low-resource setting for MT.

2.3 Novel MT Techniques

We also investigated a relatively recent paper from the post-BERT era, which focused on integrating BERT into NMT models (Zhu et al., 2020). Utilizing pre-trained contextual embeddings for multilingual NLP tasks has proven to be extremely useful and resulted in large bumps in evaluation metric scores for virtually all NLP tasks. These benefits have been seen even in the domain of low-resource NLP. In the paper, the authors tried two possible methods for integrating BERT into MT. They used BERT to initialize downstream models and then fine-tuned the models, and they also used BERT as context-aware embeddings for downstream models. They found that the first strategy actually did not result in any significant performance improvements on the MT task, but the second strategy seemed promising. They present a new type of "BERT-fused" model which feeds the BERT output into all layers of the model rather than solely using them as input embeddings only. According to the paper, "An input sequence is first transformed into representations processed by BERT. Then, by the BERTencoder attention module, each NMT encoder layer interacts with the representations obtained from BERT and eventually outputs fused representations leveraging both BERT and the NMT encoder. The decoder works similarly and fuses BERT representations and NMT encoder representations." The paper focused on high-resource languages, but it could have some beneficial applications in the low-resource setting. Ultimately, they found that they were able to achieve BLEU scores of 36.69 on a German-English task, 30.75 on an English-German task, and 43.78 on an English-French task, all of which were improvements on the current baseline scores.

Finally, we consider Siddhant et al. 2020, in which the authors attempt to build a massively multilingual NMT transfer learning model, which could take data in one of several source languages and translate it into one of several target languages. Their model works with high resource languages, and it utilizes the BERT-like strategy of employing pre-training and fine-tuning. They generated their own massive parallel dataset in 102 crawled languages, and generated a wide variety of models. Their goal was to use these NMT models towards other types of NLP tasks, such as POS tagging and NER. They develop a Massively Multilingual Translation Encoder (MMTE), and they present contrasts between mBERT and MMTE. Ultimately, their models were able to achieve various BLEU scores (withheld to save space, since this was a massively multilingual task), some of which were able to effectively beat their baseline, and others which closely approached baseline scores. This paper serves to support the idea that cross-lingual models have a place in the NMT task.

3 Experimental Design

3.1 Data

Finding parallel data between Uyghur and English was a primary challenge for this project. We had originally hoped to utilize data developed by

Dataset Statistics

	Train	Dev	Test
Tokens (ug)	27,969	2,045	2,421
Tokens (en)	41,045	2,802	3,444

Table 1: Statistics on the Tatoeba parallel data used in this project.

the Linguistic Data Consortium (LDC) for the Low Resource Languages for Emergent Incidents (LORELEI) program (Strassel and Tracey, 2016), which was used in the LoReHLT 2016 Evaluations task. (This data is described in more detail in our past milestone submissions.) However, developing models which worked with this data proved to be too difficult for our small group given the limited time and constraints on compute power that we faced. The documents contained in this corpus contained long and complex sentences, which our simpler models struggled to effectively translate, given our constraints. As a result, we decided to pivot to a different, simpler set of data for this project in the hopes of building models which can perform MT of Uyghur-English or English-Uyghur on simple sentences. We felt that this would be a good first step in developing any model that can translate between the two languages.

Instead, we used data downloaded from the Tatoeba database[2], which is a collection of parallel sentences in 330 languages. It is crowd-sourced and available under a creative commons license. This database contains parallel sentences in Uyghur and English (as well as the parallel Uzbek-English data utilized in one of our extensions). An example of a parallel sentence in from this dataset is as follows: English sentence - "Everyone has strengths and weaknesses"; Uyghur (Latin) sentence - "herkimning aartuqchiliqimu, aajizliqimu bar". (We omit the Arabic text here, because it was not compiling in Latex, but see Figure 1 for an example of parallel Uyghur Arabic script and English sentences from the data.)

Some statistics on the parallel data used are contained in Table 1. We recognize that this data is very small and simple, but this only serves to further emphasize the difficulty of building models that work with such low-resource languages. That being said, this data was quite useful for us to build some basic MT models between Uyghur and English. Given that this was crowd-sourced data in a low-resource language, it is difficult to ascertain its accuracy. However, since the sentences are fairly simple, yet diverse, we believe that the translations are generally accurate; in addition, by informally looking at some of the other languages' translations on the database website, we felt that the data we had obtained was suitable to this task.

[2] https://tatoeba.org/eng/

BLEU Score Calculation

$$p_n = \frac{\sum_{\text{n-gram} \in hyp} \text{count}_{clip}(\text{n-gram})}{\sum_{\text{ngram} \in hyp} \text{count}(\text{ngram})}$$

$$B = \begin{cases} e^{\frac{1-|ref|}{|hyp|}} & |ref| > |hyp| \\ 1 & \text{otherwise} \end{cases}$$

$$BLEU = B * e^{\frac{1}{N}\Sigma_{n=1}^{N} w_n \log(p_n)}$$

Figure 3: Here are the equations necessary to compute BLEU score. The term hyp is the hypothesis (model-translated sentence), and ref is the reference sentence(s) (gold data). In addition, p_n is the n-gram precision, B is the brevity penalty, w_n is the weight we wish to assign to each modified precision, and N is the order of n-grams we wish to consider.

3.2 Evaluation Metric

While there are several useful metrics utilized to evaluate the efficacy of an MT model (including BLEU, METEOR, NIST, etc.), we decided to use BLEU score (Papineni et al., 2002), which is one of the standard evaluation metrics for this task. BLEU score, or BiLingual Evaluation Understudy, is an algorithm for evaluating the quality of text that has been put through an MT system and translated into another natural language.

The idea behind this metric is that a high quality machine translation will be more similar to a human translated version of the text into the other natural language. BLEU score is a number between 0 and 1, and numbers closer to 1 indicate a higher quality machine translation (more similar to a human translation); we report our values in this paper as percentages (so all scores have been multiplied by 100). This metric computes a modified version of precision to compare the machine translated text to various reference human translations. BLEU computes the precision of each order of n-gram and averages them together, incorporating an addi-

Simple Baseline Performance

	Train	Dev	Test
Uyghur-English	28.90	29.69	28.11
English-Uyghur (Arabic)	2.76	4.60	37.06
English-Uyghur (Latin)	2.76	4.60	37.06

Table 2: Corpus BLEU scores of our simple baseline measures.

tional brevity penalty.

Figure 3 contains the equations used to compute the BLEU score. The modified precision is the sum of the clipped n-gram counts for all the hypothesis sentences in the corpus divided by the number of hypothesis n-grams. This is useful because a translation using the same words as in the references will tend to score higher, and longer n-gram matches between hypothesis and reference translations will also tend to improve the score. The brevity penalty is incorporated to accurately reflect in the scoring metric that a good hypothesis should be similar in length to the reference(s). Hypotheses that are too short are penalized.

Some obvious benefits of using BLEU score are that it is quick, inexpensive, simple, and language-independent. On the down side, BLEU score does not account for synonyms when evaluating the correctness of a translation based on the similarity with human translations. Generally, BLEU score is a very standard evaluation metric for this task; it is the metric used in every MT paper cited in the literature review above.

3.3 Simple Baseline

For a simple baseline, we decided to naively generate translations of text by taking every word in the source language and "translating" it into the most common word in the target language. In English, the most common word is "the," so all Uyghur words were translated to "the." In Uyghur, the most common word is غ, in Arabic script, or "u", in Latin script, so all English words were translated to those words (depending on whether we were attempting to translate into the Latin or Arabic script version of the language).

Table 2 details the performance of our simple baseline on our data. We note that some of the results reported above appear to be artificially high, and we hypothesize that this is due to the fact that many of the sentences in our data are simple and short. However, clearly translating every word into the most common word in another language is not a useful model since all meaning is lost in the translation process. This touches upon a potential downside to using BLEU scores as a metric, and in future work it would be interesting to explore the results of other evaluation metrics for this task with this data. That being said, we find that for our later baselines and extensions, BLEU score does appear to accurately reflect the performance of such models relative to one another.

In general, the BLEU scores reported for this simple baseline are higher than those reported for the published baseline models or the extension models. However, by looking at the actual results of translations produced by these later models, we see that those models are generating better translations than our simple baseline. In addition, it is to be expected that our later models will achieve relatively low BLEU scores since they are operating under a low-resource setting, and compared to the BLEU scores reported in other low-resource MT papers, such as Guzmán et al. 2019, the scores reported by our later models are not atypical.

4 Experimental Results

4.1 Published Baseline

Since Uyghur is a low-resource language, there have not been many papers focusing on building Uyghur-English or English-Uyghur MT models. In addition, there are no published models performing this task using the data we obtained. Therefore, we decided to implement a model similar to the work discussed in Sutskever et al. 2014. This paper described one of the first key implementations of an NMT system by using the encoder-decoder paradigm to construct a sequence-to-sequence model.

In such a model, the encoder is a neural network that converts an input sentence in one language into a fixed-length representation vector, and the decoder is another neural network that takes in that representation vector and outputs (in the case of MT) the translated sentence. The encoder and decoder are jointly trained to increase the likelihood of the model outputting the correct translation of a sentence.

In Sutskever et al. 2014, the authors describe a model which uses LSTMs in the encoder and decoder networks. They find that their model actually tended to perform well on long sentences, which is a traditional pitfall of encoder-decoder NMT models. The authors of this paper used a different dataset than ours. (We selected the paper

Published Baseline Model Performance

	Dev	Test
Uyghur (Arabic)-English	12.23	12.42
Uyghur (Latin)-English	16.05	15.27
English-Uyghur (Arabic)	16.74	23.41
English-Uyghur (Latin)	7.86	3.72

Table 3: Corpus BLEU scores of our published baseline models.

which presented the baseline model most similar to what we wished to implement; we were not easily able to find a paper which implemented our baseline model using the Uyghur data to which we had access.) The authors reported overall success with their model, achieving a BLEU score of 34.81 on an English to French translation task. Naturally, these are very high-resource languages, so we do not expect to achieve the same degree of success, but the results are nonetheless promising and were state-of-the-art at the time of publication.

We decided to implement a similar sequence-to-sequence model which uses bidirectional LSTMs in the encoder and decoder networks, with a global attention mechanism, as described in Luong et al. 2015 (which achieved a BLEU score of 25.9 on a similar English to German task). The attention mechanism helps to improve the model performance by telling the model how much "attention" to pay to each token's representation in the intermediate vector between the encoder and decoder.

We trained our models for 100 epochs, with early stopping in place (models tended to stop training around 30 epochs). We used a batch size of 48, an initial learning rate of 0.001, the number of layers set to 2, and a dropout of 0.1. The model uses perplexity on the validation set to compute the loss. The performance of our models on the data is contained in Table 3. The scores reported appear to be on par with those found in similar low-resource MT papers working with similar types of data, such as Guzmán et al. 2019.

4.2 Extension - Romanization

We were fortunate enough to have access to a small mapping of Uyghur words in Arabic script to their corresponding Latin script words. In addition, we had a mapping between Arabic script characters and Latin script characters for the official Uyghur alphabets. We used these to provide a transliteration function which would take the Arabic script Uyghur sentences and transform the tokens into their corresponding Latin script. This is a unique facet to the Uyghur language, and we had hoped to leverage this in our design. In addition, we thought it might be useful to have models which operate in either official script, and for the purposes of error analysis we found it to be of use since neither author is very familiar with Arabic script.

Related Language Model Performance

	Dev	Test
Uyghur (Arabic)-English	11.28	9.65
Uyghur (Latin)-English	13.76	13.54

Table 4: Corpus BLEU scores of our related language models.

We were curious to see if models trained to translate between English and Uyghur would perform better when both the source and target languages were in the same script. However, we found that in general the opposite was true. The models trained to translate between English and Uyghur Arabic script, in either direction, tended to slightly outperform those working with Uyghur Latin script. Overall the models tended to perform very similarly to one another, but we hypothesize that the Arabic script models may have done slightly better due to some potential information loss during the romanization process. The results discussed can be seen in Tables 4 through 10.

4.3 Extension - Related Languages

The next idea that we had was to consider whether we might be able to augment the training data with some data in a closely related language. We downloaded some parallel Uzbek-English data from the Tatoeba database site to this end. We chose Uzbek because it is the language most closely related to Uyghur, and syntactically their structures are very similar. We simply appended this Uzbek data to the training dataset, and kept our development and test sets the same. The small additional Uzbek data contained 1652 tokens (494 additional sentences). For this extension, we only constructed models that translated from Uyghur to English, since using Uzbek for English-Uyghur translation in this context did not make sense.

Table 4 contains the results of this extension. This method tended to hurt model performance, which is not wholly unexpected, but we figured that we would try it. If it had worked, this would have been useful since Uzbek is a higher-resource language and could then be further used to augment training data for Uyghur models. Given that cross-

Uyghur (Arabic) to English Extended Models

Dropout	Num. Layers	Test
0.1	1	21.49
0.1	2	23.00
0.1	5	17.04
0.1	10	14.45
0.2	**1**	**25.15**
0.2	2	18.73
0.2	5	13.32
0.2	10	1.83
0.3	1	23.22
0.3	2	23.58
0.3	5	18.67
0.3	10	6.50
0.5	1	24.01
0.5	2	21.43
0.5	5	14.17

Table 5: Corpus BLEU scores of our modified-dev-validated models as a result of hyperparameter search, translating from Uyghur (Arabic) to English. Models tended to stop training anywhere from middle 40s epochs or more, with best models training for around 50 epochs in general.

lingual language model pretraining has proven to be highly effective for other NLP tasks including those in the low-resource setting (Lample and Conneau, 2019), it would nonetheless be interesting to see, in the future, if incorporating related languages into training data could help with MT when working with larger or higher quality datasets or more advanced types of neural models.

4.4 Extension - Modified Development Set

For this extension, we constructed a modified development set which contained some seen and some unseen data. We left our disjoint test set the same. The modified dev set contained all of the original unseen dev set plus a few randomly chosen instances from the training set. We hypothesized that including some of these seen values might force the model to train longer and guide the training process. Though this is typically not standard practice when training NLP models, since the dev set is used to ensure that a model will generalize to unseen data, we found that, in this case, since we were working with such small amounts of data, this was actually helpful.

Since this seemed to provide some improved performance on the test set, we decided to perform a hyperparameter search for these models, which were trained on the train set, validated on the modified dev set, and finally evaluated on the test set. Tables 5, 6, 7, and 8 contain the results of that hyperparameter search. Generally, using either 1 or 2 layers was better for our models, and lower dropout (either 0.1 or 0.2) values also tended to lead to better results. We also tried out this extension technique for the related language models, and the results are contained in Table 10.

Uyghur (Latin) to English Extended Models

Dropout	Num. Layers	Test
0.1	1	23.54
0.1	2	22.12
0.2	**1**	**24.43**
0.2	2	22.06
0.3	1	24.13
0.3	2	23.67
0.5	1	22.39
0.5	2	24.43

Table 6: Corpus BLEU scores of our modified-dev-validated models as a result of hyperparameter search, translating from Uyghur (Latin) to English. Models tended to stop training around 50 epochs.

English to Uyghur (Arabic) Extended Models

Dropout	Num. Layers	Test
0.1	1	9.10
0.1	**2**	**9.12**
0.1	1	8.52
0.2	2	8.48
0.3	1	8.68
0.3	2	8.47
0.5	1	8.98
0.5	2	8.80

Table 7: Corpus BLEU scores of our modified-dev-validated models as a result of hyperparameter search, translating from English to Uyghur (Arabic). Models tended to stop training around 50 epochs.

4.5 Discussion

Ultimately, we found that incorporating romanization did not greatly affect model performance, and incorporating related languages tended to actually hurt performance a bit. Validating on a modified dev set during the training process significantly improved performance. Models translating from Uyghur to English tended to achieve higher BLEU scores than those translating from English to Uyghur, regardless of script. Our best Uyghur-English models (regardless of script) were our extended (modified-dev) models which are bolded in the tables above. The best Uyghur (Arabic) to English model achieved a BLEU score of 25.15, while the best Uyghur (Latin) to English model achieved a BLEU score of 24.43. Our best English to Uyghur (Arabic) model was actually our published base-

English to Uyghur (Latin) Extended Models

Dropout	Num. Layers	Test
0.1	1	8.86
0.1	2	8.53
0.1	2	8.80
0.2	**2**	**9.03**
0.3	1	8.80
0.3	2	8.89
0.5	1	8.57
0.5	2	8.93

Table 8: Corpus BLEU scores of our modified-dev-validated models as a result of hyperparameter search, translating from English to Uyghur (Latin). Models tended to stop training around 50 epochs.

Summary of Best Results

	Test
Uyghur (Arabic) to English Published Baseline	12.42
Uyghur (Arabic) to English Best Model	**25.15**
Uyghur (Latin) to English Published Baseline	15.27
Uyghur (Latin) to English Best Model	**24.43**
English to Uyghur (Arabic) Published Baseline	23.41
English to Uyghur (Arabic) Best Model	**23.41**
English to Uyghur (Latin) Published Baseline	3.72
English to Uyghur (Latin) Best Model	**9.03**

Table 9: Summary of the relevant corpus BLEU scores described in the various tables above.

line model, which achieved a BLEU score of 23.41, while our best English to Uyghur (Latin) model was an extension model that achieved a BLEU score of 9.03. While these scores are on the lower end, it is important to note that they appear to be on par with many of the BLEU scores reported by the other low-resource MT papers, such as Guzmán et al. 2019. Table 9 summarizes these results.

4.6 Brief Error Analysis

We will now briefly consider some of the mistakes made by some of our best models, and relate them to those made by our published baseline models. Generally, our extended models tended to outperform our published baseline models. Generally, our models dealing with Uyghur in the Arabic script tended to make similar errors to those dealing with the Latin script. It is difficult to tell much beyond that for the models which translate from English into Uyghur (Arabic) since neither author knows the language. All we can do for those cases are look at the predicted sentence text and visually compare it to the gold sentence text, coupled with the sentence-level BLEU score. However, since we found that models dealing with Arabic script seemed to do well and make mistakes in a similar fashion to those dealing with Uyghur (Latin), we will primarily consider our Uyghur-English and English-Uyghur models that operate with a Latin script. (This is also easier to show in the Latex document.)

For our Uyghur (Arabic)-English models, an example gold sentence was: “This man is your friend, remember?” The published baseline produced the following translation: “This is your man is here.” The best model produced the following translation: “Is this man your friend?” When we consider the same gold sentence for our Uyghur (Latin)-English models, the published baseline produced the following translation: “Is this man your friend?” The best model produced the following translation: “This man is your friend?” Generally, this example is fairly illustrative of model performance. Sentences that tended to stump the best models, equally tended to stump the published baseline, but when the best models performed well, they outperformed the published baseline by constructing generally more coherent sentences.

For our English-Uyghur models, we consider the gold sentence: “u doxtur emes aoqutquchi.” The published baseline produced the following translation: “u bir chirayliq aoylinip”. The best model produced the following translation: “u doxtur bir lazim aemes.” (The word “doxtur” means “doctor,” so this gives us a clue as to the meaning of the sentence.) We see that neither translation is perfect, but the best model tends to get more words correct. It is difficult for us to determine whether the translations constitute grammatically correct Uyghur sentences, but we felt that this example was illustrative of our overall model performance in this area.

5 Conclusion

In this term project, we devloped NMT models for Uyghur to English, as well as English to Uyghur. Our best Uyghur-English model achieved a BLEU score of 25.15 (an extended model which beat our published baseline), and our best English-Uyghur model achieved a BLEU score of 23.41 (our baseline model). We implemented several extensions to varying degrees of success. We achieved model performance on par with similar low-resource NMT findings, conducted on similar datasets. Ultimately, low-resource MT remains a difficult challenge in NLP, but we were grateful to be able to learn more about this task in a hands-on manner. We would like to continue improving upon the work begun

Related Language Extended Model Performance

	Test
Uyghur (Arabic)-English	24.10
Uyghur (Latin)-English	21.40

Table 10: Corpus BLEU scores of our modified-dev-validated related language models.

in this project in the future, possibly by developing more complex state-of-the-art models trained on larger data, with more time and computing resources.

6 Acknowledgements

We would like to thank Professor Mark Yatskar and all of the course staff of CIS 530: Computational Linguistics at the University of Pennsylvania for all of their instruction and support throughout the semester. In addition, we would like to thank the author of the code in this repository[3], upon which we based parts of our project code. We used Google Colab for our project, and a notebook containing all of our work is submitted along with the rest of the project.[4]

References

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Kishore Papineni, Salim Roukos, Todd Ward, and 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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Hendra Setiawan, Zhongqiang Huang, and Rabih Zbib. 2018. Bbns low-resource machine translation for the lorehlt 2016 evaluation. *Machine Translation*, 32(1):45–57.

[3] https://github.com/snnclsr/nmt/

[4] https://colab.research.google.com/drive/1bQSA6_5qea88qLAr3KnTQbfUUv3SxDlz?usp=sharing

Aditya Siddhant, Melvin Johnson, Henry Tsai, Naveen Ari, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman. 2020. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8854–8861.

Stephanie Strassel and Jennifer Tracey. 2016. Lorelei language packs: Data, tools, and resources for technology development in low resource languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3273–3280.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*.