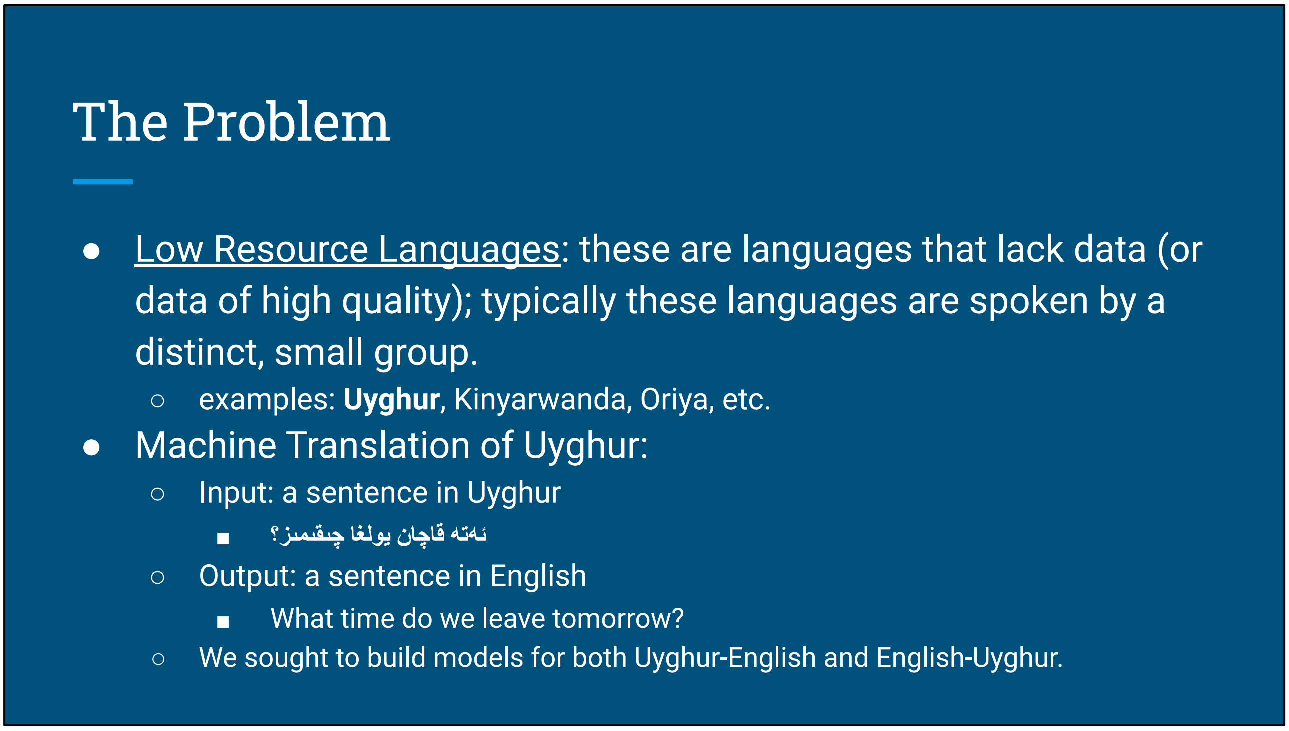
Low Resource Neural Machine Translation of Uyghur

CIS 530 Final Project
Efe Ayhan & Francesca Marini
Spring 2021

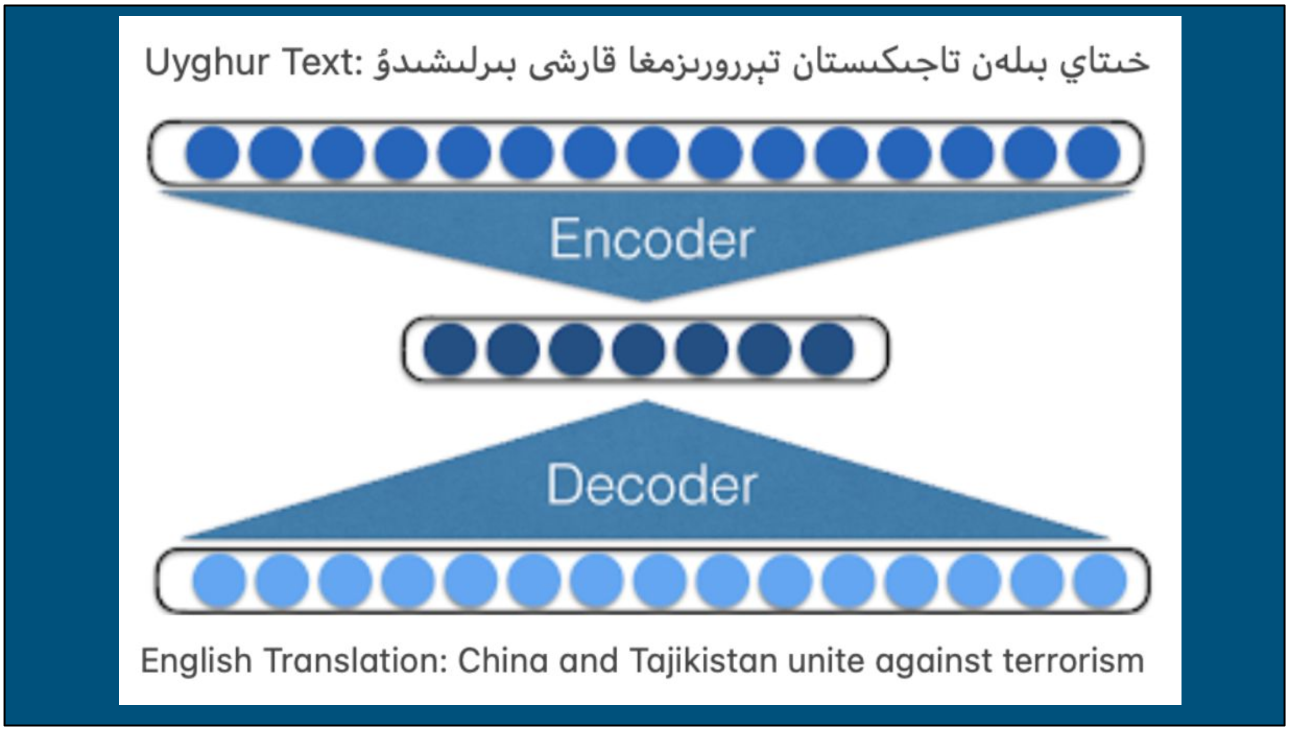


The Problem

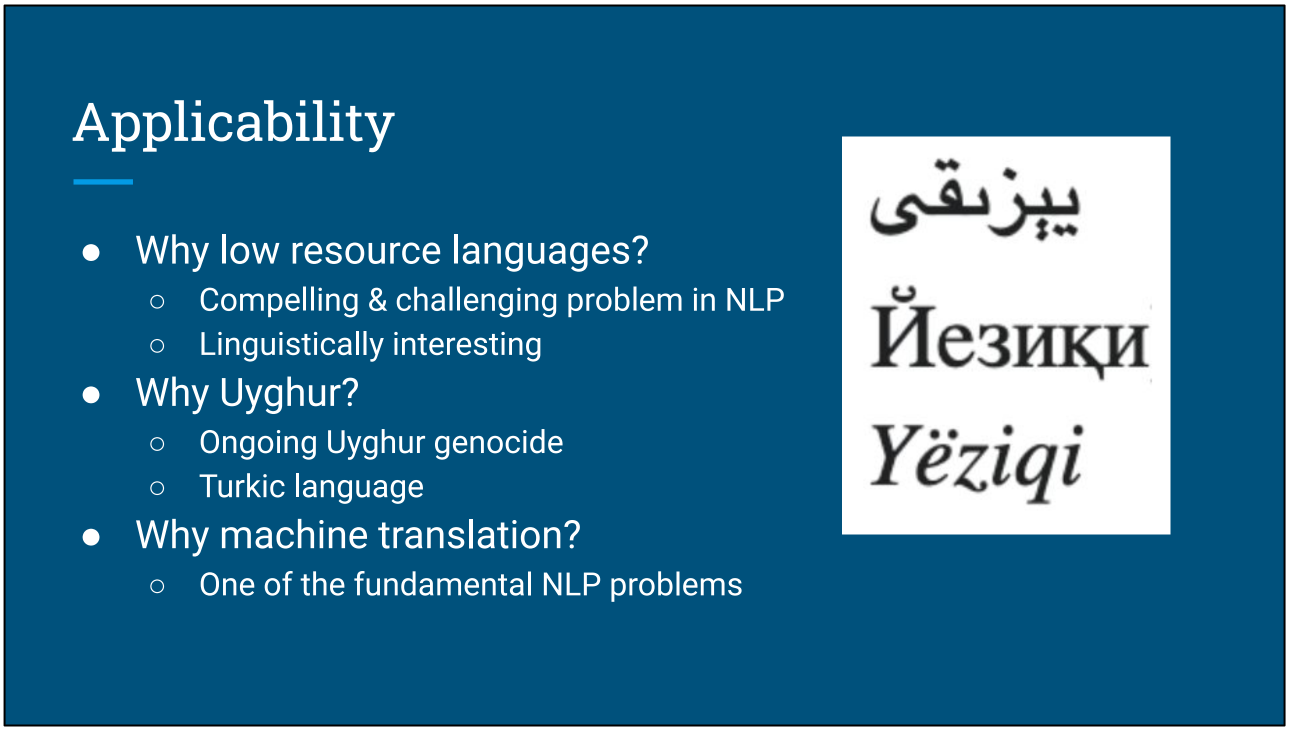
- Low Resource Languages: these are languages that lack data (or data of high quality); typically these languages are spoken by a distinct, small group.
  - examples: **Uyghur**, Kinyarwanda, Oriya, etc.
- Machine Translation of Uyghur:
  - Input: a sentence in Uyghur
    - ئەتە قاچان يولغا چىقىمىز؟
  - Output: a sentence in English
    - What time do we leave tomorrow?
  - We sought to build models for both Uyghur-English and English-Uyghur.

- Low Resource Languages: these are languages that lack data (or data of high quality); typically these languages are spoken by a distinct, small group.
  - These languages pose particular challenges in NLP since there is often little, low-quality data, and current SOTA neural models often need large amounts of data to effectively perform tasks.
  - In addition, because the languages are low resource, it is often difficult to obtain access to native speakers, so annotating or evaluating the quality of annotations in a low resource language can be especially difficult.
- Machine Translation explanation
- Translating from Uyghur to English is useful because it can make documents written by Uyghurs more accessible to the rest of the world (especially given the current events related to their current persecution)

- Translation in the other direction would also be useful to bring written documents in other (high resource) languages to Uyghur people


Uyghur Text: خىتاي بىلەن تاجىكىستان تېررورىزمغا قارشى بىرلىشىدۇ
Encoder
Decoder
English Translation: China and Tajikistan unite against terrorism

Here is an image which conveys what the MT problem does. It also visualizes at a high level how a neural MT model works. This would be a model that translates a sentence in Uyghur (Arabic script) to English.



Applicability

- Why low resource languages?
  - Compelling & challenging problem in NLP
  - Linguistically interesting
- Why Uyghur?
  - Ongoing Uyghur genocide
  - Turkic language
- Why machine translation?
  - One of the fundamental NLP problems

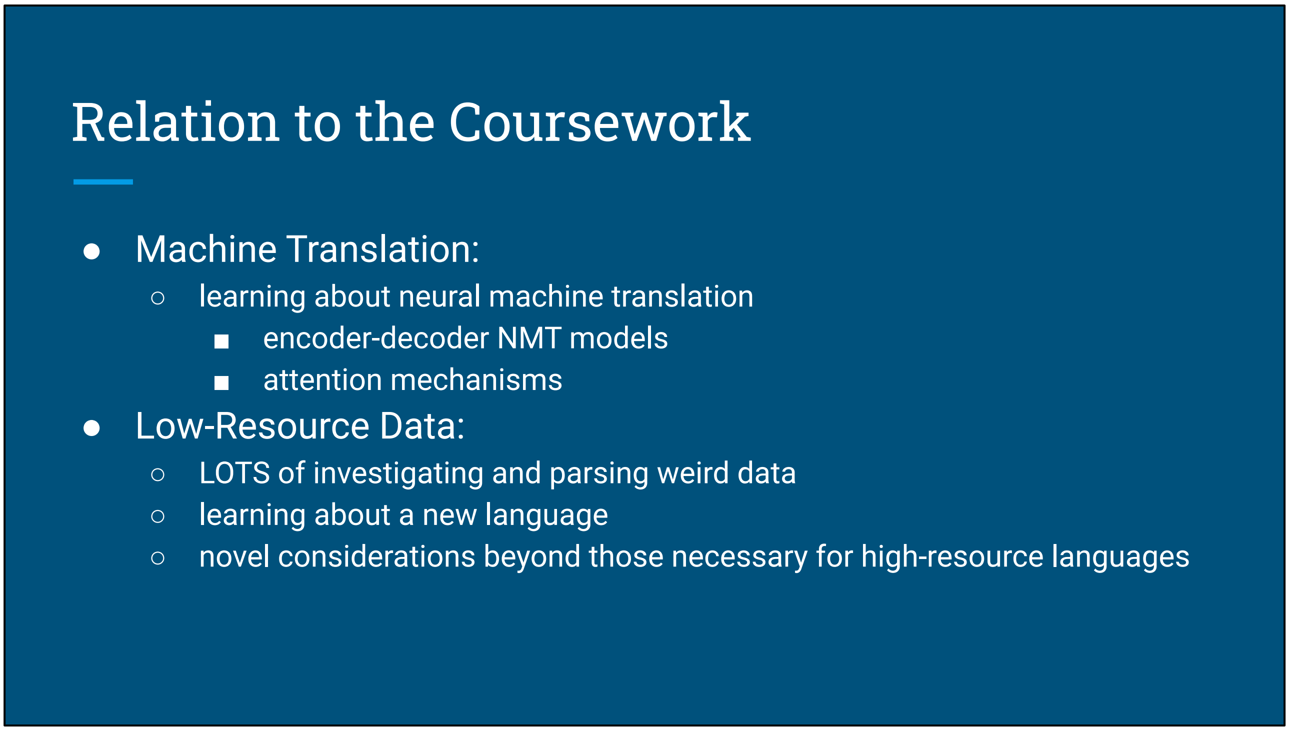
يېزىقى

Йезиқи

Yëziqi

- Why Low Resource?
  - We both have an interest in linguistics in general, and NLP in relation to low resource languages is full of challenging and interesting problems
  - We wanted to do something outside of the box, and we hope to continue working on this project in the future after the completion of this class
- Why Uyghur?
  - Currently, there is a genocide being committed against the Uyghur population in northwest China; it has been going on for several years, but has lately been making a lot of global news.
  - Uyghur is a Turkic language written in Arabic script, but it also has official alphabets in the Cyrillic and Latin alphabets.
  - It is closely related to Uzbek.

    - Efe is Turkish, so he knows a related high-resource language; we thought that this could both come in handy and make the project more interesting.
    - We have access to some Turkish and Uzbek data, and this might be interesting to incorporate into the project.
- Why machine translation?
    - While we both have some experience with languages and machine learning coming into the course, neither of us had any experience with machine translation before.
    - Since we missed that optional homework by electing to do the project, we hoped to learn more about how this works through the work put into the term project.
- Figure is the Uyghur word for “alphabet” in all three scripts.

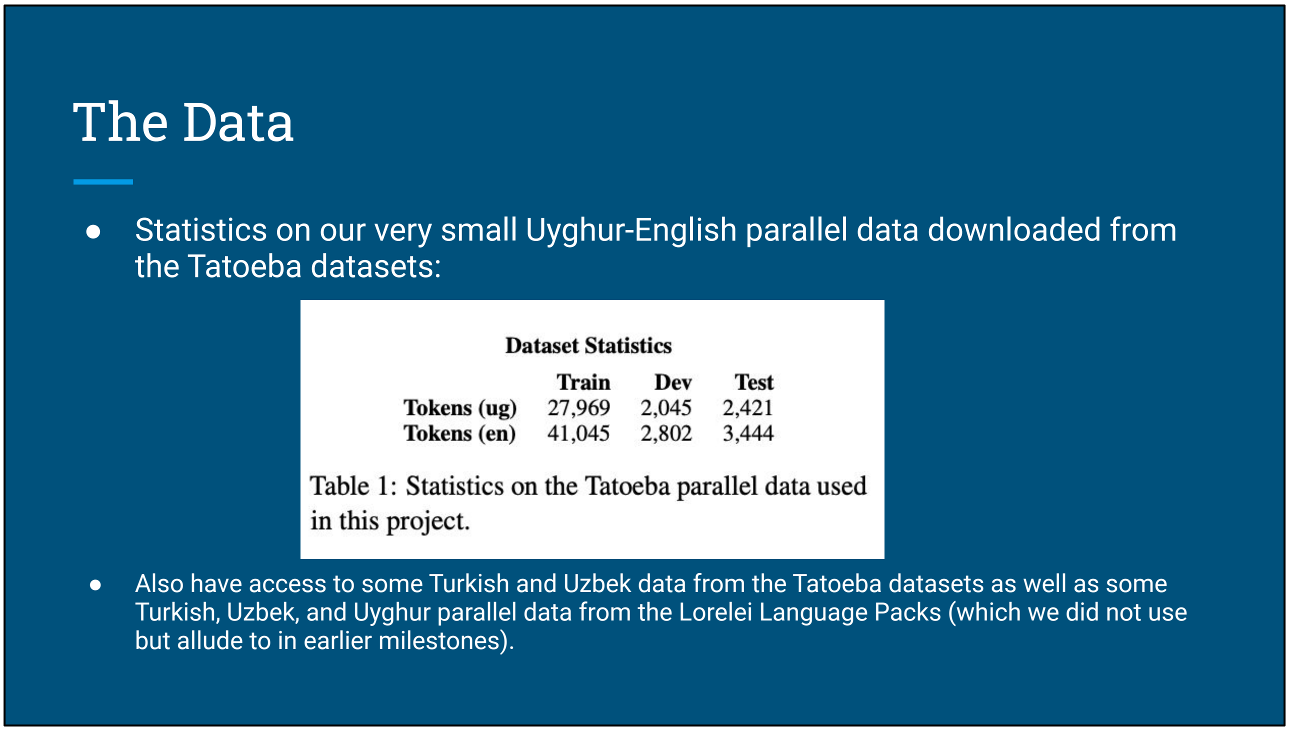


Relation to the Coursework

- Machine Translation:
  - learning about neural machine translation
    - encoder-decoder NMT models
    - attention mechanisms
- Low-Resource Data:
  - LOTS of investigating and parsing weird data
  - learning about a new language
  - novel considerations beyond those necessary for high-resource languages

- Machine Translation:
  - learning about neural machine translation
    - encoder-decoder models /sequence-to-sequence NMT models
    - attention mechanisms
  - even more current SOTA models use transformers / multilingual embeddings
    - we learned about these during the literature review and while researching models to implement for the baseline
    - we originally hoped to implement a more current transformer-style model
    - we want to continue this project and incorporate some of these ideas down the line with more time
- Low-Resource Data:
  - LOTS of investigating and parsing weird data

    - novel considerations beyond those necessary for high-resource languages (like - can we somehow use related high resource languages to make up for the fact that we have little / poor quality data in the source language?)



The Data

- Statistics on our very small Uyghur-English parallel data downloaded from the Tatoeba datasets:

Dataset Statistics

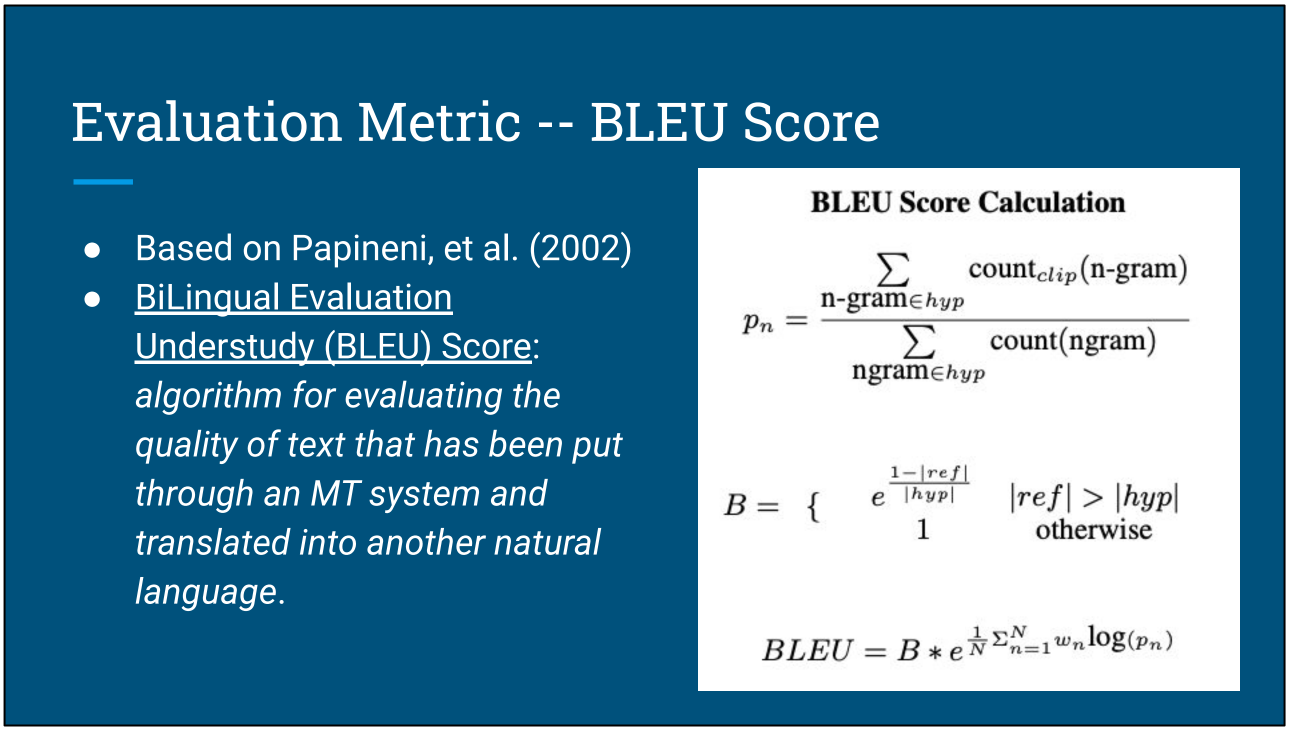
	Train	**Dev**	**Test**
Tokens (ug)	27,969	2,045	2,421
Tokens (en)	41,045	2,802	3,444

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

- Also have access to some Turkish and Uzbek data from the Tatoeba datasets as well as some Turkish, Uzbek, and Uyghur parallel data from the Lorelei Language Packs (which we did not use but allude to in earlier milestones).

- Above is a summary of the Uyghur-English data used for our baseline models
- There was a LOT of pre-processing involved in working with all of this data (especially given the fact that we began working with one dataset and had to end up switching to another dataset in the middle of the project, so we basically had to do everything twice with different starting document formats)
    - We also transliterated all of the Uyghur data in Arabic script into Latin script using the official Uyghur Latin alphabet mappings
- We ended up switching to a different dataset because the sentences in the previous Lorelei Uyghur-English dataset were very long, and models trained on that data failed to learn.
- We tried everything to get models to work with this data, but

- we believe that a combination of a lack of time and resources would not allow us to achieve a model that could effectively deal with this data.
- We instead utilized the Tatoeba Uyghur-English dataset, which we manually randomly split into a 80-10-10 train-dev-test partition.
    - This data is also extremely small, but the sentences are simpler, which lend themselves to this task better.
- Also have access to somewhat larger Turkish and Uzbek datasets from the Tatoeba datasets, as well as some Turkish (~900K tokens), Uzbek (~1M tokens), and Uyghur (~55K tokens) parallel data from the Lorelei Language Packs.



Evaluation Metric -- BLEU Score

- Based on Papineni, et al. (2002)
- BiLingual Evaluation Understudy (BLEU) Score: *algorithm for evaluating the quality of text that has been put through an MT system and translated into another natural language.*

BLEU Score Calculation

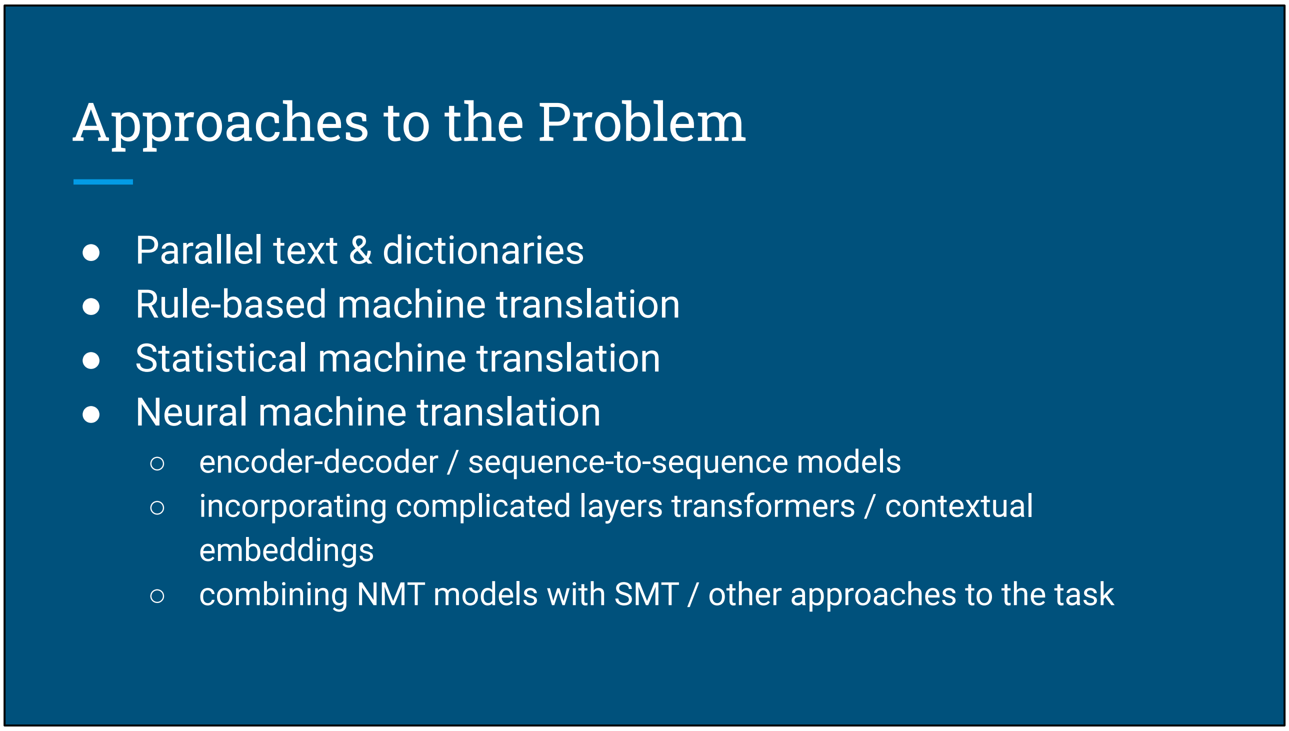
$$p_n = \frac{\sum_{n\text{-gram} \in hyp} \text{count}_{clip}(\text{n-gram})}{\sum_{\text{ngram} \in hyp} \text{count}(\text{ngram})}$$

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

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

- BiLingual Evaluation Understudy (BLEU) Score:
    - MT evaluation metric
    - multiple translations can be “correct”
    - a good translation does not necessarily have to match the gold reference translation(s)
    - modified n-gram precision
    - higher values are better (number between 0 and 1 -- or number between 0 and 100)
    - pros:
        - quick, inexpensive, simple, language-independent
    - cons:
        - does not account for everything (like synonyms are viewed as “different” words, when maybe they shouldn’t really be)
    - Accounts for the fact that multiple translations can be “correct”
    - Accounts for the fact that a translation that does not

  - perfectly match the gold translation might still be a generally good translation
  - Modified n-gram precision computation
  - quick, inexpensive, simple, language-independent
- Algorithm: (based on article here -- https://towardsdatascience.com/bleu-bilingual-evaluation-understudy-2b4eab9bcfd1)
  - Count the maximum number of times a candidate n-gram occurs in any single reference translation; this is referred to as Count.
  - For each reference sentence, count the number of times a candidate n-gram occurs. As we have three reference translations, we calculate, Ref 1 count, Ref2 count, and Ref 3 count.
  - Take the maximum number of n-grams occurrences in any reference count. Also known as Max Ref Count.
  - Take the minimum of the Count and Max Ref Count. Also known as Count clip as it clips the total count of each candidate word by its maximum reference count: CountClip = min(Count, MaxRefCount).
  - Add all these clipped counts.
  - Finally, divide the clipped counts by the total (unclipped) number of candidate n-grams to get the modified precision score: Pn = sum of the clipped n-gram counts for all the candidate sentences in the corpus divide by the number of candidate n-grams.
  - Compute the Brevity Penalty: BP = 1 if c > r, and e^((1-r)/c) else, where c is the count of words in a candidate translation, and r is the count of words in a reference translation.
  - Compute the BLEU Score: BLEU = BP * exp(sum_from_n=1_to_N(w_n * logPn)), where BP is the brevity penalty, N is the number of n-grams, w_n is the weight for each modified precision, Pn is the modified precision.

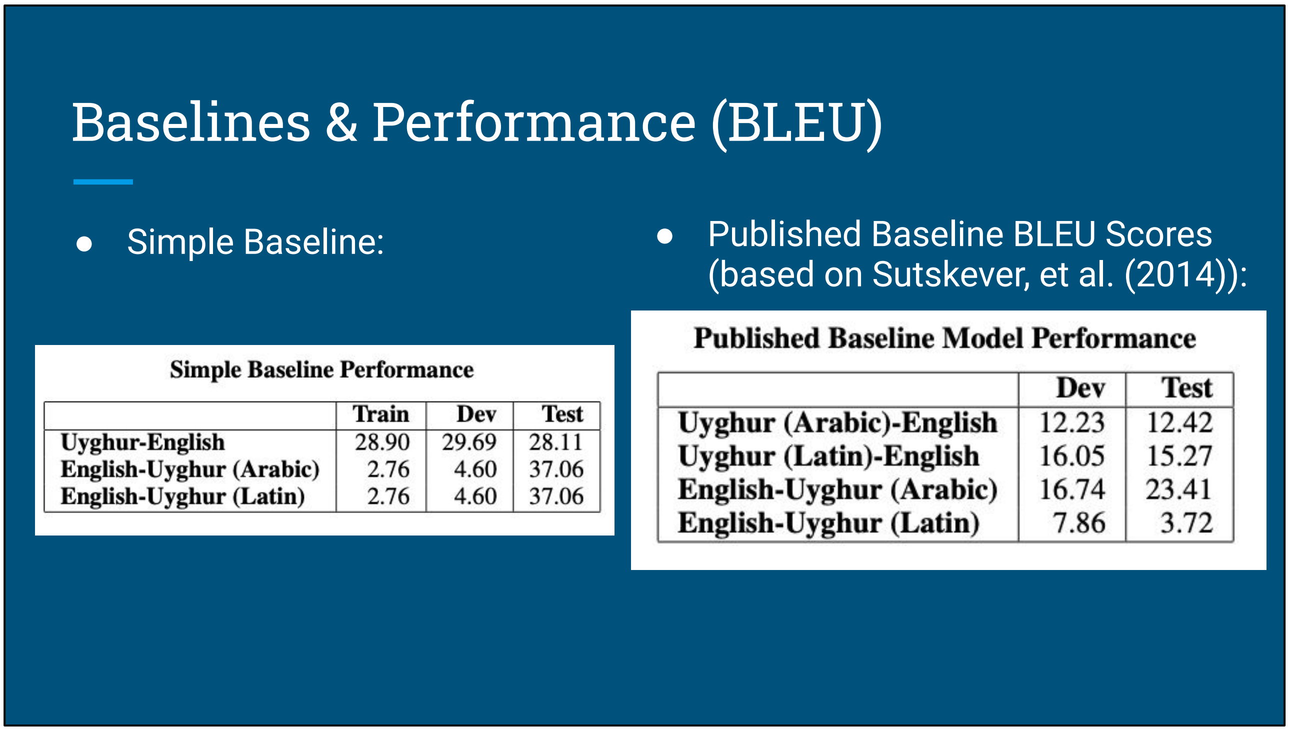


Approaches to the Problem

- Parallel text & dictionaries
- Rule-based machine translation
- Statistical machine translation
- Neural machine translation
    - encoder-decoder / sequence-to-sequence models
    - incorporating complicated layers transformers / contextual embeddings
    - combining NMT models with SMT / other approaches to the task

- info based on the article (http://towardsdatascience.com/machine-translation-a-short-overview-91343ff39c9f) as well as our literature search
- Rule-Based:
    - 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
    - VERY language-specific
    - pipeline format
    - does not require parallel text, but does require dictionaries and well-defined syntactic rules for both languages
- Statistical:
    - This approach uses statistical models based on the analysis of bilingual text corpora

  - three steps: (1) language model; (2) translation model; (3) word-order determiner
  - less-language specific, but also when the source and target languages are very different (especially syntactically), performance suffers
- Neural:
  - encoder-decoder models / sequence-to-sequence
    - encoder is a neural net (often LSTM) that takes source input and converts it into a vector
    - decoder is a neural net (often LSTM) that takes vector and converts it into target output
  - uses parallel text
  - rare word problem, problematic data can make training hard
  - combining NMT and SMT approaches to circumvent issues
  - recent developments have also been attempting to integrate BERT / other contextual embeddings into NMT



Baselines & Performance (BLEU)

- Simple Baseline:

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

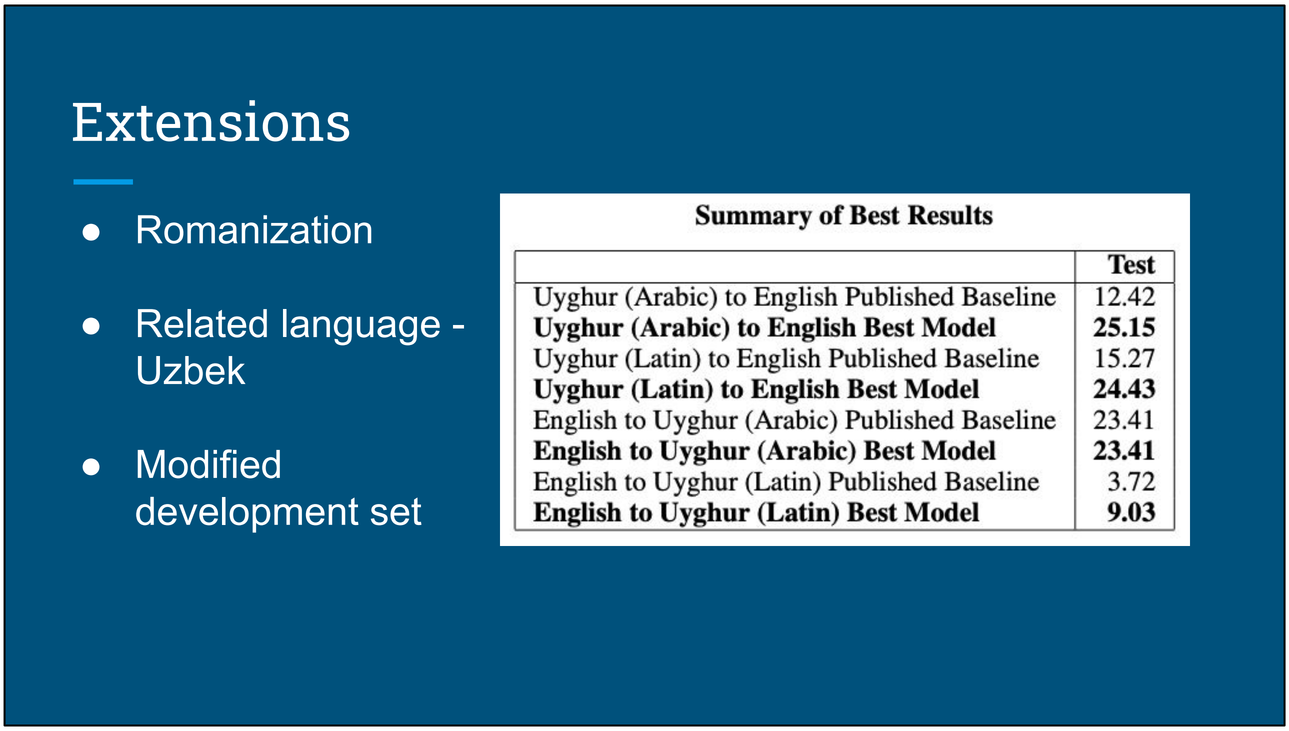
- Published Baseline BLEU Scores (based on Sutskever, et al. (2014)):

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

- Simple Baseline:
  - translate each word into the most common word in the target language
  - artificially high (possibly due to short length of many sentences and small vocabulary)
    - points to a potential downside of using BLEU -- though in general for comparing published baseline models with extension models, BLEU seemed to work fine
- Published Baseline
  - seq2seq model with global attention; bi-LSTM layers for encoder-decoder
  - not current SOTA (since there are things like transformer models for the encoder and decoder networks)
    - but still fairly recent

- but given our time and resource constraints, implementing a similar yet simpler model for this project seemed feasible (especially since Uyghur embeddings are not readily available online -- though this is something we would like to do in the future)
- sanity check models score in the 90s, as would be expected
- generally scores are on the low end of things
- to be expected because low resource
- on par with similar low-resource NMT papers (cited in the report) which work with similar types of data



Extensions

- Romanization
- Related language - Uzbek
- Modified development set

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**

- Romanization - basically didn't change anything
- Incorporating a small amount of a related language (Uzbek) into the training data hurt results
- Modified development set: take the regular disjoint dev set and add in a few randomly selected sentences from the train set (keep train set and disjoint / unseen test set exactly the same)
  - this improved model performance -- we hypothesize because it kept models training longer on the data before early stopping was hit
  - generally would not be a standard practice, but we figured why not try it since we have so little data and low-resource situations generally have different demands
  - results in table - this extension tended to beat the published baseline (with the exception of the

    - English-Uyghur (Arabic) model, where the published baseline was the best model)
- More details about our thoughts / theories for why things worked or didn’t work are contained in the final report

Thank You!