Zetong Chen, Kaito Minami, Menjia Qi, Xusheng Shangguan, Yiming Yao

Purpose of This Project

Within the Spanish-speaking communities in the US, there are problems where Spanish speakers cannot understand each other for their unique dialects despite speaking the same underlying language. Our project was based on the result of investigation during the service-learning work to address this issue of communication barriers to further flourish socialization and promote understanding of diversity and cultures within the community. This project is particularly interesting because it tackles the traditional challenge of machine translation as well as the minor differences between multiple dialects. We also explored multiple methods of machine translation from pre-Neural Networks age Statistical Machine Translation to modern Transformer models. Upon completion of this project, we expect it to contribute to fostering the inter-cultural Spanish speaking community across the US, where immigration from Latin American countries is most common.

Data

Our project focuses on English Spanish machine translation with dialect variation. For the core training English Spanish parallel data, we took from two major sources: first, the OPUS GNOME localization corpus and second, the Tatoeba sentence collection. This project uses two OPUS for 15 Spanish dialects from Latin American regions (Argentina, Chile, Colombia, Costa Rica, Dominican Republic, Ecuador, Honduras, Nicaragua, Panama, Peru, Puerto Rico, El Salvador, Uruguay, and Venezuela). Each dialect folder contains parallel English Spanish technical UI strings such as error messages, and instructions. After cleaning, we have a few hundred thousand parallel sentence pairs across dialects. However, we found that the dialect datasets lacked the daily Spanish conversation phrases, which we aimed to do.

To cover general-purpose language beyond technical strings, we have a second data source: Tatoeba, a large multilingual sentence corpus. Tatoeba provides daily use sentences, such as greetings and general daily language usage. We first strip license and attribution metadata, and then split the remaining content into two plain-text files (spa.en and spa.es) for later use. These sentences are then concatenated to each dialect's cleaned GNOME data. Finally, every dialect has both technical and conversational parallel data.

For subword modeling, we train a shared SentencePiece BPE tokenizer on the standard Spanish training split and then apply it to all other dialects.

Different Models

Baseline model: SMT

SMT stands for Statistical Machine Translation, we used IBM Model 1 as our first model, which is actually one of the simplest classical machine translation models. It only relies on word-level probabilities, and it will ignore grammar, syntax, and context. Since it is so simple, it gives us a good baseline to compare the advanced neural network models.
In the training step, we provided many English-Spanish parallel corpuses which also contain Spanish dialects. Then we set multiple EM (Expectation-Maximization) iterations for the model to train then refine the alignment probabilities of words. During EM iterations,the system infers which Spanish words align with which English words by analyzing co-occurance statistics across the corpus. IBM Model 1 treats all alignments as equally likely

initially, each iteration gradually increasing the probability of word pairs that consistently appear together, while reducing the probability of incorrect or infrequent alignments.

Once the training is done, the model builds a word-level translation dictionary. For each English word, the model selects the Spanish word with the highest alignment probability. Then in the test process, the English input will be tokenized into the whole words, then the model will translate it word by word based on the translation dictionary. If the word appears in the dictionary, the corresponding Spanish word will be used to translate, otherwise, it will leave the word without translation.

RNN seq2seq + Attention

The second model we used for this project is RNN sequence to sequence with attention mechanism. The standard recurrent neural network can capture the word order, syntax, and sequential dependency. In addition to that, we add an attention mechanism to the regular RNN, this is because the attention mechanism can avoid information bottlenecks, so that the decoder can focus on the important source sentence, which really helps improve the semantic alignment and translation quality.

In the encoder, we have an embedding layer and a RNN layer. The embedding layer here converts each word into a vector, and then the RNN layer will process these vectors one by one, and keep tracking the context. After this, the information will be processed to encoder. As what we have described above, we introduced an attention mechanism. This means when the RNN generates spanish word by word, the attention we have implemented will calculate the alignment score for each word, so that the model can understand which part of the sentence should focus on. At last, we have a linear layer and a softmax function to select the next most likely word for the sentence, and repeat this the full sentence will be generated.

For the unknown words, I replaced them with the <UNK> token, so that they will not break the decoding process. During the training, any words that do not appear in the vocabulary are mapped to UNK. This really makes the model more reliable and robust, because none of the encoder and decoder will break, and the model can still generate reasonable outputs when rare and unseen words appear. For the validation, we split the train test based on 80% train, and 20% test, which means 10% will be validation. We calculated the validation accuracy and loss after each epoch’s training was done.

Fine-tuned mT5

Our translation system uses mT5-small, which is Google's multilingual version of the T5 transformer. The core idea is straightforward: we treat translation as a text-to-text problem. You feed in an English sentence, and the model learns to output the corresponding Spanish translation. It's learning the conditional probability P(Spanish | English) by minimizing cross-entropy loss during training[5][6].

General structure: What makes transformers powerful for this task is the self-attention mechanism. Unlike older approaches that might use bag-of-words or fixed-length vectors, the model processes a sequence of subword tokens and builds contextual representations for each one[6]. This means a word like "bank" gets a different representation depending on whether we're talking about a river bank or a financial institution. The encoder reads the English input, and then the decoder generates Spanish tokens one at a time, looking at both what it's already generated and the encoded English representation.

Unknown Words**:** One nice thing about mT5 is that it uses SentencePiece tokenization, which breaks words into subword pieces[5]. So if the model encounters a word it's never seen before, it doesn't just output UNK and breaks the word down into smaller chunks it does recognize. This turned out to be especially useful for our GNOME data since there's a lot of technical jargon and weird formatting like underscores for keyboard shortcuts.

Training**:** We pulled together English-Spanish pairs from two Spanish dialect files (Dominican and Argentine) from the GNOME localization project. We did a 90/10 train/validation split. For the actual training, we used a learning rate of 3e-4, batch size of 4, and ran it for 3 epochs[7]. Each English input gets a prefix "translate English to Spanish: " at the beginning. For each dialect, we used a fresh new model for that dialect so the model can be more specialized to the target person during our service learning.

Decoding**:** Here's where things get interesting. When you want to actually generate a translation, you can't realistically search through every possible output sequence for there are way too many possibilities. So we tried a few different shortcuts. Greedy decoding just picks the most likely word at each step, which is fast but can lead you down bad paths. Beam search is smarter: it keeps track of multiple candidate sequences at once (we tried beam widths of 4 and 8) and picks the best one at the end. We also played with length penalties to see if we could get the model to produce longer or shorter translations.

In conclusion

The first iteration of our results was done on the dataset containing each unique dialect data on top of a shared general Spanish-English sentences dataset. In this experiment, to measure the translated sentence accuracy to the reference sentences, we used 4 metrics: BLEU score, chrF score, METEOR score, and COMET score. BLEU score measures the precision-oriented overlap of 1-gram to 4-gram word-level similarity. ChrF score computes character-level F-score, which can account for precision and recall scores at the same time. METEOR score can quantify recall-oriented word-level matches. And the COMET score evaluates semantic similarity between translated and reference sentences. With a combination of these 4 metrics, we aim to evaluate and achieve well-rounded translated sentences that native Spanish speakers from Latin American countries can fully understand.

Overall, RNN model achieved the highest scores across metrics except METEOR score, which SMT earned the highest. For structural similarity within data and its nature of more direct statistical prediction in SMT, we achieved a somewhat uniform distribution of scores for SMT between each dialect model. With the same reason, we can conclude that the dictionary-style retrieval translation of SMT is superior to the neural translation of our RNN and mT5 in recall-oriented METEOR score evaluation.

Model	BLEU	chrF	METEOR	COMET
es-AR	0.1379	0.4516	0.4618	0.6551
es-CL	0.1379	0.4516	0.4618	0.6551
es-CO	0.1368	0.4565	0.4682	0.6562
es-CR	0.1379	0.4516	0.4618	0.6551
es-DO	0.1379	0.4516	0.4618	0.6549
es-EC	0.1368	0.4565	0.4682	0.6562
es-HN	0.1379	0.4516	0.4618	0.6551
es-NI	0.1379	0.4516	0.4618	0.6551
es-PA	0.1379	0.4516	0.4618	0.6551
es-PE	0.1368	0.4565	0.4682	0.6562
es-PR	0.1368	0.4565	0.4682	0.6562
es-SV	0.1368	0.4565	0.4682	0.6562
es-UY	0.138	0.4519	0.4621	0.655
es-VE	0.1368	0.4565	0.4682	0.6562

Table 1: SMT - General+Dialect

Model	BLEU	chrF	METEOR	COMET
es-AR	4.87	22.47	0.2995	0.4680
es-CL	6.53	22.40	0.3027	0.4720
es-CO	8.17	22.74	0.3009	0.4770
es-CR	5.30	21.59	0.2876	0.4812
es-DO	8.05	23.12	0.2940	0.4909
es-EC	7.56	22.22	0.3090	0.4880
es-HN	5.48	22.20	0.2931	0.4731
es-NI	6.03	22.62	0.2931	0.4722
es-PA	6.04	22.17	0.2977	0.4562
es-PE	7.01	22.03	0.2968	0.4885
es-PR	6.57	23.04	0.2968	0.4663
es-SV	6.32	23.36	0.3024	0.4645
es-UY	6.07	21.33	0.2762	0.4566
es-VE	6.97	21.38	0.2895	0.4640

Table 2: RNN - General+Dialect

Model	BLEU	chrF	METEOR	COMET
es-AR	5.52	3.93	0.0862	0.2400

Table 3: mT5 - General+Dialect

Contrary to our hypothesis before the experiment, the modern machine translation model of mT5 did not perform as well as the less complex neural model counterpart of RNN in our metrics. While the training loss decrease from 0.347 to 0.161 and the validation loss decrease from 0.189 to 0.138 indicates the potential learning, the final model failed to output proper translation, reflected in especially low METEOR and COMET scores. Because of its complex and compute-demanding nature of the Transformer model, within allotted time and compute, mT5 could not complete all 15 dialects and the fine-tuning was especially challenging. As an attempt to improve the result, we implemented multiple decoding processes including greedy and different settings of beam search (1, 4, 8). A larger beam search allowed the model to explore more candidates and produce smoother, more consistent phrasing, but neither gave perfect translations. Changing the length penalty from 0.6 to 1.4 often did not contribute to improvement of the result.

Model	BLEU	chrF	METEOR	COMET
es-AR	16.84	35.85	0.2708	0.6051
es-CL	22.38	38.31	0.2725	0.6081
es-CO	20.26	35.80	0.2782	0.62
es-CR	21.37	34.81	0.2610	0.5879
es-DO	28.65	41.49	0.3094	0.6283
es-EC	27.15	38.72	0.3094	0.6221
es-HN	23.29	38.45	0.3004	0.6187
es-NI	21.24	35.83	0.2802	0.6135
es-PA	26.23	41.78	0.3495	0.6455
es-PE	28.19	40.59	0.3166	0.0.6433
es-PR	24.12	41.10	0.3245	0.6258
es-SV	24.5	37.82	0.2713	0.6081
es-UY	25.32	37.35	0.2949	0.6321
es-VE	21.30	37.63	0.3	0.6180

Table 1: RNN Dialect

In the second iteration of our experiment, we have also trained our model on a dataset that only contains dialects; the results are above. We specifically trained the RNN model, which gained the best overall score in the last experiment. We can see that every metric result has changed significantly. In this experiment, we have higher BLEU, higher chrF, METEOR, and COMET. Take AR as an example, in the previous model, the BLEU for AR was only about 5, but this time, it rose to around 17. We think this is because the general Spanish data makes the model worse at translating specific dialects. This is because of the structural difference between the Spanish for general use and the Spanish for technical documents. The wording in the general use dataset is less formal, and we have multiple versions of Spanish translation for one simple English, but in technical documents, Spanish used in them is unique. Therefore, we conclude that this model improvement comes from data homogeneity.

Future experiments/explorations/additions

Based on our results, we still have room for improvement for the metrics and usability. Currently, the dialect datasets only hold non-daily use technical terminology, which can only be used in very specific context that doesn't align with our purpose of this project and we lack native Spanish speakers to semantically check longer sentence translation. To mitigate this issue, we can research more dialect datasets to add to our training and recruit volunteers from the service-learning host to check and provide dialect-unique phrases and sentences.

Additionally, the model improvement will be a key for the major improvement in accuracy and practicality of the resulting translation. We concluded that the limitation for mT5 model in this experiment was time and compute power. With more time and compute allocated, we can train these data on even bigger models such as mT5-base instead of mT5-small, which we expect to generally learn better representations. The data augmentation techniques such as backtranslation helps to increase the quantity of data to train and test on, which can increase the accuracy of results.

Finally, in the future, we can add audio functionality and web app deployment for the end users like elderly in a service-learning facility to easily access and utilize our machine translation service. Since a lot of conversations happen audibly and to accommodate non-technical elderly generation, Spanish speech recognition will be a perfect addition. Web app deployment also allows us to provide machine translation service across the nation and around the world, extending the circle of inclusion.

Works Cited

1. https://huggingface.co/google/mt5-small
2. https://opus.nlpl.eu/wikimedia/en&es/v20230407/wikimedia
3. https://keras.io/examples/nlp/neural_machine_translation_with_transformer/
4. https://tatoeba.org/en/downloads
5. https://github.com/NeyoNought47/CS-4120-Project-Repo
6. https://arxiv.org/abs/2010.11934
7. https://aclanthology.org/2020.emnlp-demos.6/
8. https://jmlr.org/papers/volume21/20-074/20-074.pdf