Our Language Model: Our team developed a multi-language character recommendation system by training a neural language model. Our neural network has 3 components: an embedding layer, an RNN layer, and a fully connected layer (Figure 1 below) [4, 12]. The embedding layer converts character tokens into a continuous vector representing non-contextual character embeddings [3, 4]. These embedded characters are fed into an RNN layer to obtain each character's hidden state that summarizes the preceding characters [4, 7, 12]. Each character's hidden states are then put into a fully connected layer to obtain prediction vectors for each character. Our model's prediction for the next character to come at the end of the sequence is the argmax of the last character's prediction vector in the input sequence.

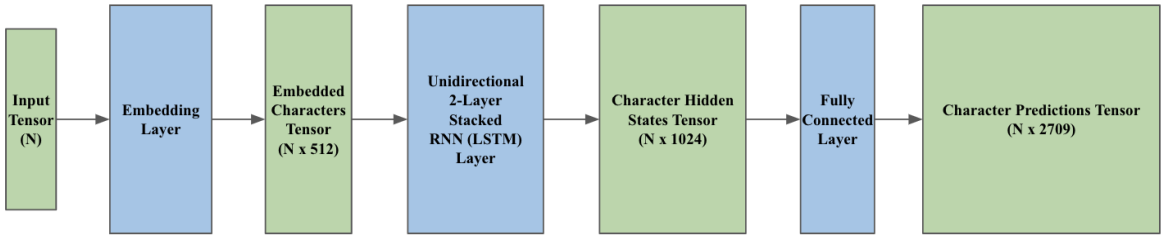

Figure 1. *The structure of our model's architecture that creates predictions for each character in a given sentence.*

Our network's core lies in the LSTM layer that we chose to avoid the vanishing gradient problem for long sequences of characters [4, 12]. This layer is unidirectional because we wanted our model to only depend on preceding characters to simulate the test environment. Additionally, it is a 2-layer stacked LSTM; it makes our model complex enough to handle the complex task of predicting characters for various languages, grammars, and contexts [7, 12]. Our model's key component is that it is language-agnostic, and we do not try to parse what language is given before feeding it into our model. This allows our model to learn patterns across languages and subsequently make predictions on languages that it has never seen before, a critical functionality particularly considering we do not know the test languages. However, this is a trade-off as we don't have the inductive bias that would give us higher accuracies on trained languages. Another downside of our model is its complexity and a large number of parameters, which requires a lot of data to learn latent features of languages.

Data: Our dataset contains 10 languages: English, Japanese, Spanish, German, Polish, Russian, French, Greek, Korean and Chinese [5, 6, 8, 9, 10]. The corpora from these dataset are diverse in both linguistic origin and context (e.g. formality, tv scripts, social media, etc.). We split this data into training, development, and test sets of size 100,000, 20,000, and 20,000 sentences respectively; with each language contributing 10% to each dataset. We then built up a character vocabulary consisting of frequently occurring characters from the training set, along with a UNK character so our model is able to handle predictions on sequences with unseen characters [3]. We utilized the dev set while training our model so that we could observe if our model was generalizing well or overfitting, and we tuned our model accordingly [2]. We only used the test set when a model finished training to examine its performance. Our final model had 82.4% training accuracy and 80.8% development and test accuracy.

Key Resources: Our model was built with the PyTorch library. We also utilized PyTorch's official character-level RNN tutorial [11] for the early stages of our project as a tool to get a deeper understanding of how RNNs could be used for character-level models. We also used the course-provided sentiment analysis notebook as starter code [1]. The combination of these resources provided us the ability to try a series of different model architectures that ultimately led up to our final model.

References

[1] CSE 447 Course Staff. 2021.Sentiment Classification Example Using Gru On The Imdb Dataset. https://colab.research.google.com/drive/14GAMb7c6FbDnhWvqcliCZ8KYNvqdnQz7?usp=sharing

[2] CSE 447 Course Staff. 2021. Introduction, Gradient Descent, and A1. (2021) https://drive.google.com/file/d/1f917ObCpPq0li0v1zPja_tgRDhjrIzcJ/view

[3] CSE 447 Course Staff. 2021. Probability Review, LM. (2021) https://drive.google.com/file/d/11ds23Mx48630atDlag4Sg1SA9gy8QF17/view

[4] CSE 447 Course Staff. 2021. Neural Language Model. (2021) https://drive.google.com/file/d/1MroXHe9dDDR8gF9fJaFH3XevKfa2YAYG/view

[5] Aesop Chow. 2017. ChineseNLPCorpus. (2021). https://github.com/InsaneLife/ChineseNLPCorpus/tree/master/NER/MSRA

[6] Pryzant R. Chung Y. Jurafsky D. Britz D. 2018. JESC: Japanese-English Subtitle Corpus. Language Resources and Evaluation Conference (LREC). (2021) https://nlp.stanford.edu/projects/jesc/

[7] Yoav Goldberg. 2016. A Primer on Neural Network Models for Natural Language Processing. (2021) https://www.jair.org/index.php/jair/article/view/11030/26198

[8] Philipp Koehn. European Parliament Proceedings Parallel Corpus 1996-2011. (2021). https://www.statmt.org/europarl

[9] Илья Козиев. 2019. Russian-language NLP datasets. (2021). https://github.com/Koziev/NLP_Datasets

[10] Jungyeul Park. 2019. Korean Parallel corpora. (2021). https://github.com/jungyeul/korean-parallel-corpora

[11] Sean Robertson. 2021. NLP From Scratch: Generating Names With A Character-level RNN. https://pytorch.org/tutorials/intermediate/char_rnn_generation_tutorial.html

[12] Noah Smith. 2021. Natural Language Processing (CSE 517 & 447): (Neural)Language Models. (2021) https://drive.google.com/file/d/15xk-qyd3DFBLBYlTBDegfuZJKElJxuk4/view