An Natural Language Processing Approach to Sumerian Translation

Ethan Rogers[1], Indrajeet Aditya Roy[1], Emery Jacobowitz[1], Jiajun Wang[1]

[1] *Khoury College of Computer Sciences, Northeastern University*

https://github.com/ThePineappleW/sumerian

Abstract

Sumerian is a dead language isolate previously spoken in what is current day southern Iraq. There have been previous attempts to use machine translation to translate transliterated Sumerian to English, though some of these approaches have failed or been limited in scope. Our goal for this research was to build translation and named entity recognition models with moderately high accuracy. Long Short Term Memory Translation models trained on Sumerian/English word forms, part of speech tags, and agglomerative clustering categories achieved 94-96% token level F1-score. We also explored the impact of SentencePiece tokenization methods on translation accuracy and constructed a context sensitive Named Entity Recognition model. Our results show that a variety of approaches can be used to overcome the limitations of small corpus size and high frequency of lexical gaps when studying ancient languages.

Introduction

Sumerian is an ancient language actively spoken from around 3000BC to 1700BC [1] in Sumer; the area that is now known as modern day southern Iraq. Drawn to settle by the ample wildlife and fertile, wet soil of the Euphrates and Tigris rivers, the first inhabitants of ancient Sumer developed a sophisticated language and writing system, with the first evidence of written Sumerian dating to the end of the fourth millennium [2]. Over roughly 1000 years, Sumerian evolved as the people of Sumer built a vast network of city states, powered by the abundant crop yields and social structures of organized civilization. Sumerian has been preserved through its writing system, Cuneiform. Using reeds and clay tablets, Cuneiform could record routine business administration and historical information, as well as philosophical thought and narrative storytelling.

Drawing from the excellent work of Jagersma, 2010 [3] Sumerian morphology is agglutinative—any given word is likely to contain a number of inflectional morphemes appended (generally as enclitics) onto a stem. Nominals are inflected for number, gender (animate vs. inanimate), case, and possessives. Verb forms are even more complex, allowing both prefixes and suffixes to convey agreement with both subjects, objects, pronominal references, various modalities, and tense and aspect information.

The first decipherment of Cuneiform from a dictionary tablet inscribed with both Sumerian and Akkadian Cuneiform occurred in the early 1800s, shortly after is modern reintroduction in the late 1700s [4]. Since initial decipherment, Academics studying ancient Sumer and the Sumerian language have translated hundreds of individual works, characterized the evolution and mixing of Sumerian with other ancient languages over its period of active use, and developed compendia such as the Pennsylvania Sumerian Dictionary [5]. Even after years of study, translation of Sumerian remains difficult to automate for a variety of reasons – mainly Sumerian's small and fragmented repository of material, its lexical distance from other languages, and the Sumerian's gaps in terminology created by millennia of dormancy.

Related Work

The first published approach to machine translation of Sumerian was Pagé-Perron et al. 2017, whose work detailed a possible system of supervised classifiers trained on rule-based morphologic analysis, POS tagging, and syntactic parsing of existing translated texts [6]. Unfortunately, this approach was unable to overcome the challenge of data sparsity. Punia et al. 2020 built on previous work, with the first attempt at machine translation of whole Sumerian transliterations [7] to English. Along with the public release of a Sumerian training corpus, Punia et al. 2020 tested several model architectures, finding that a sequence-to-sequence encoder/decoder model with pretrained embeddings from a Wikipedia corpus outperformed both a phrase-based n-gram model and transformers-based model for both BLEU score [8] and expert evaluations. These results are encouraging, however as training data was limited to only administrative texts from around 2100BC, the generalizability of this work to Sumerian over its thousand-year lifetime is limited. Although no other work has yet to explore machine Sumerian translation, there are several single word *glossing* models that translate single words without context [9, 10] and computer vision models that can classify Sumerian Cuneiform [11, 12].

Problem Statements

This work is an exploration of how natural language processing (NLP) methods can assist with translation of ancient and low resource languages. There are many possible avenues this research could focus on and therefore many experiments that could be conducted. As the last native speaker of Sumerian died thousands of years ago, our team centered the ideation and design of our research on providing solutions to perceived problems encountered by academics who study Sumerian.

We identified two problems that NLP approaches could potentially address – the need for a bidirectional translator for short-sequence Sumerian transliterations and their English translations; and a named entity recognition (NER) system to partially fill lexical gaps in English created by translation of Sumerian-language specific proper nouns. *Our goal for this project was to build a functional translation model and NER model, with at least 80% training accuracy.*

Methods

Our methods sought to address challenges presented with ancient and low resource languages, specifically the lack of large training corpora, novel syntactic structure, lexical gaps, and transliteration idiosyncrasies. We obtained and cleaned a public dataset of Sumerian literature, then using a custom tokenizer and clustering algorithm created a translator-classifier hybrid model for translation between English and Sumerian transliterations. Additionally, we used entity tagging information from the dataset to generate a spaCy model for NER of 10 classes of Sumerian entities.

The Dataset

The Electronic Text Corpus of the Sumerian Literature (ECTSL) is a project maintained by Oxford University in Cambridge, England [13]. It contains the transliterations and translations of 394 Sumerian documents and meta tags for translation and document quality. Each transliteration word has a corresponding lemma, form, part of speech (POS) tag, and single English translation. To prepare ECTSL for model training, relevant transliterations, tags, and translations were extracted from the raw XML files. Text was cleaned of extraneous formatting characters. Numerals were excluded to constrain project scope.

Data Preprocessing

The ETCSL transliteration scheme is *not* morphemic. Each hyphen represents a logogram boundary, not a morpheme boundary as is the convention in other settings. This can cause confusion when a morpheme's phonological realization changes in various contexts. For example, the genitive case marker –*ak* alternates with –*a* word-finally. Thus, the transliteration scheme does not directly represent the morphemic content of a surface form, but rather indirectly maps between them.

The challenge is thus to preprocess transliterations in such a way that words can be considered at a level closer to the underlying morphology. The model should be able to consider the stem *lugal* 'king' independently of inflection, while also retaining the grammatical functions of the remaining suffixes, whether the word is *lugal-a* 'of the king' or *lugal-ra* 'for the king'. One approach is to train a neural morphological segmenter. This approach was found in Mager, et al. 2022 [14] to be preferable for low-resource polysynthetic languages, outperforming our chosen method. However, this would require a large amount of pre-segmented data, which would take a large amount of time to generate. As such, this avenue is currently unfeasible but would be a valuable effort for future work.

We handled the preprocessing issue with a sophisticated form of subword tokenization. Drawing on the subword-based approach Sennrich et al. 2015 [15] to rare words we trained a set of Google SentencePiece [16] tokenizers on our Sumerian vocabulary of 35,707 word forms. Subword tokenization is often used to handle out-of-vocabulary tokens, as it allows them to be decomposed into smaller components which may be in the vocabulary. With a highly agglutinative language, this allows the model to more effectively handle forms generated through productive processes. For example, some plural forms may be marked with reduplication, as in *gal-gal,* 'all the waters'. Subword tokenization allows this to be decomposed into *gal* and *gal*, both of which are in the vocabulary.

We trained three different SentencePiece tokenizers on the vocabulary. First, a byte-pair encoding (BPE) model, which decomposes terms into characters and builds successively larger subtokens based on frequency. Second, a unigram model, and third, a word-based tokenization scheme akin to standard whitespace tokenization. We also used our own naive delimiter-based tokenization method (default).

Aside from a custom tokenization schema, we incorporated agglomerative clusters as input categories into the translation models. Words can have multiple meanings based on their context, and words used in similar contexts have similar meanings. Agglomerative clustering groups semantically similar words together, helping the model maintain semantic consistency during translation. We used a pretrained Word2Vec model, GoogleNews-vectors-negative300 to transform words into vectors. We chose this model for its broad coverage and robust performance. Principal Component Analysis (PCA) was then used to retain the most significant features before iteratively merging clusters based on Euclidean distance until the desired number of clusters – in our case 140. The choice of linkage criteria for agglomerative clustering, Ward's method, aims to minimize variance within clusters, yielding more balanced and semantically coherent groupings.

Translation Models

As the inputs when translating from each language are different, we created two models to handle each translation direction independently. Although the architecture is shared between models the inputs are different – English text for English to Sumerian and Sumerian transliterations for Sumerian to English, as well as pos tag and agglomerative cluster category. The prolific use of POS tags and lemma forms was informed by Sánchez-Cartagena, et al. 2024 [15], who found that these annotations are especially beneficial for translation quality in the domain of low-resource languages. All inputs were tokenized by the default Keras tokenizer and embedded into their own embedding layer. Although we experimented with SentencePiece, were able to achieve the best performance with the default delimiter-based tokenization scheme. We captured the relationship between semantic meaning from the input text and grammatic information from the POS tag by concatenating the input text embeddings and POS tag embeddings into a combined feature vector.

The translation model consists of the following layers implemented in Python with TensorFlow Keras: an embeddings layer that accepts padded inputs, a long short term memory (LSTM) cell with 512 hidden layers, a dropout layer with dropout probability of 0.5, and a dense connected layer with SoftMax activation function. (See Figure 2C.) Models were trained with the Adam optimizer with a learning rate of 0.001 for 100 epochs (with early stopping enabled) with sparse categorical cross entropy as the loss function. 20% of the dataset was reserved for validation testing during training.

NER Model

The ECTSL dataset contains 10 categories of Sumerian ideophonic entities: *deities, ethnonyms, geographic names, months, objects, persons, royals, settlements, toponyms,* and *watercourses*. As many of these terms have no modern analogs, such as names of gods (deities) or settlements that no longer exist, these ideophones are translated to English as their transliterated Sumerian word forms. To provide context, we trained a spaCy convolutional neural network to label these entities in English translations of Sumerian text. spaCy is an open-source NLP workflow library based on NTLK. Its high-level API includes a named entity recognition pipeline, that automates training of NER models. spaCy's NER pipeline is transition-based, meaning the model's internal state changes as it crawls along a sequence of tokens, taking actions based on each token or sets of tokens [17]. The model itself is a trigram convolutional neural network. It has multiple perceptron layers to further reduce dimensionality of while encoding information of neighboring tokens into the input vector. These models can handle arbitrarily truncated text and are computationally cheaper. Our NER pipeline was trained on English translations with labeled entities from the 10 categories of Sumerian ideophones. The spaCy NER pipeline handles embedding and preprocessing, needing only the original text and a dictionary of indices and labels of each entity in the text.

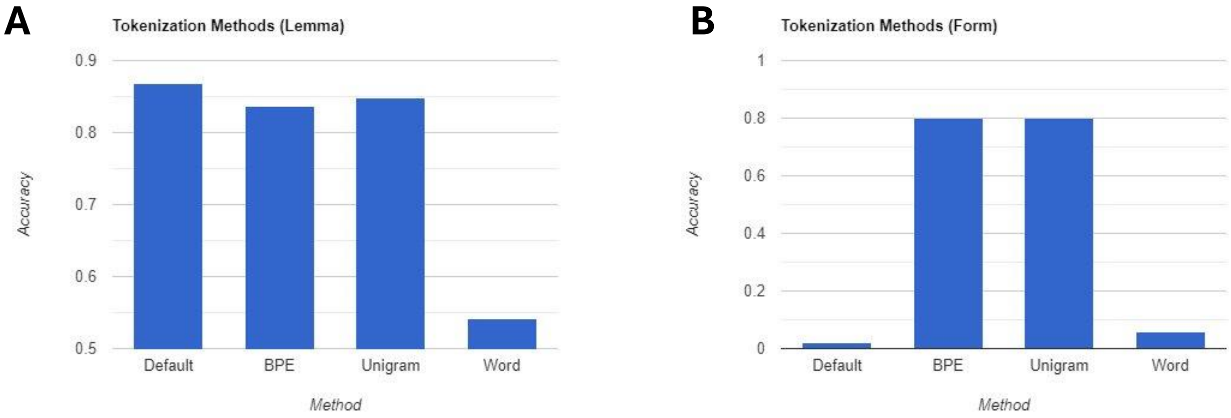
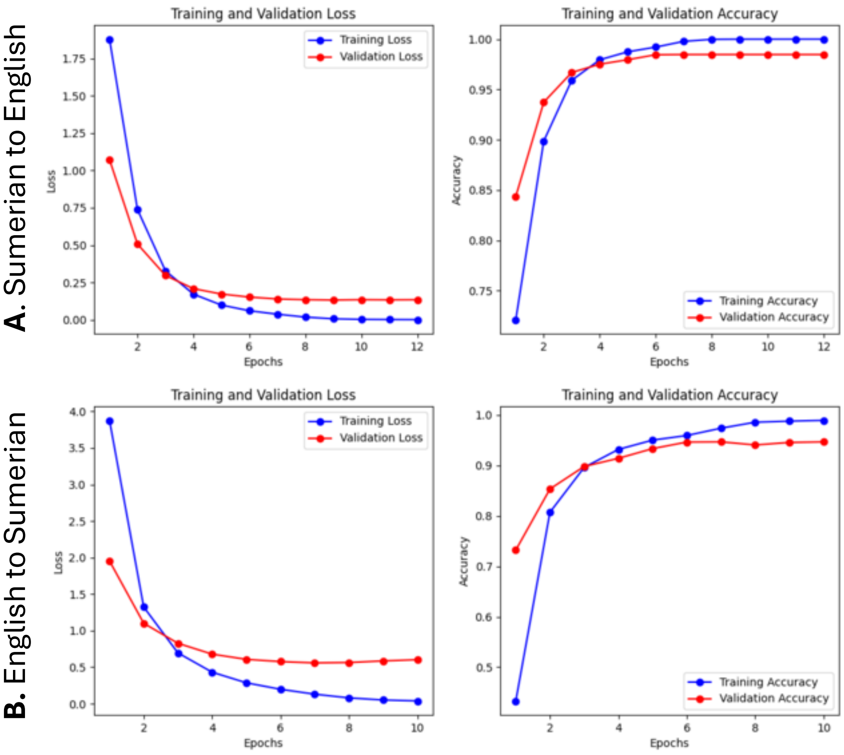


Figure 2: Token level accuracy for a simple Sumerian to English model (LSTM Seq2Seq used in Experiment 2 without parameter tuning) with inputs as lemma (A) or form (B) by method. BPE and Unigram methods perfomed well for both lemma and form.



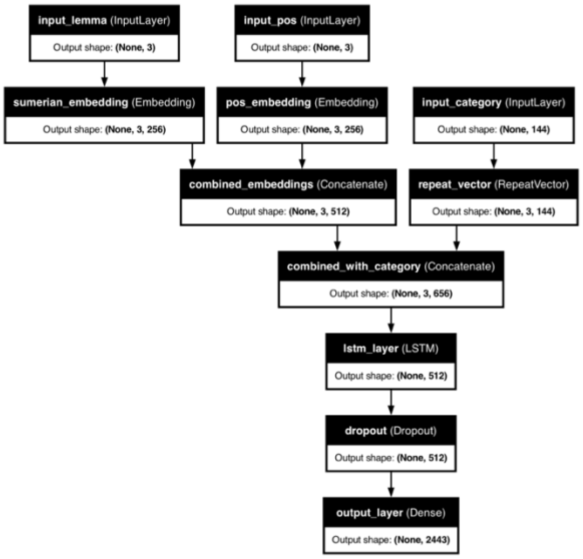

input_lemma (InputLayer)
Output shape: (None, 3)
input_pos (InputLayer)
Output shape: (None, 3)
sumerian_embedding (Embedding)
Output shape: (None, 3, 256)
pos_embedding (Embedding)
Output shape: (None, 3, 256)
input_category (InputLayer)
Output shape: (None, 144)
combined_embeddings (Concatenate)
Output shape: (None, 3, 512)
repeat_vector (RepeatVector)
Output shape: (None, 3, 144)
combined_with_category (Concatenate)
Output shape: (None, 3, 656)
lstm_layer (LSTM)
Output shape: (None, 512)
dropout (Dropout)
Output shape: (None, 512)
output_layer (Dense)
Output shape: (None, 2443)

Figure 2: Sumerian to English (A) and English to Sumerian (B) translation model loss and accuracy during training. (C) shows show model architecture for the Sumerian to English model. Architectures are the same between models, with the only difference being the inputs.

Experiments and Results

Experiment 1: Accuracy of the Tokenization Schemes

On lemmatized data, the differences between the tokenization scheme were minimal. (See Figure 1A.) This is to be expected, as lemmatized data should only ever consist of a single stem. However, this is less useful for the task of machine translation since lemmatization removes semantic complexity. On surface form models, subword tokenization becomes incredibly important. While the default tokenization method performed very poorly, BPE and Unigram methods converged to about 80% accuracy. (See Figure 1B.) This convergence is likely the result of the limited size of the corpus, containing only about 3,000 unique surface forms. These results highlight the importance of subword tokenization for morphologically complex languages. These techniques are especially useful in low-resource languages where proper morphological segmentation is unavailable.

Experiment 2: Accuracy of the Translation Models

Token level F1 score and accuracy of the Sumerian to English and English to Sumerian translation models with default tokenizers were 94% and 96% for both metrics respectively for each direction after 10 epochs of training. (See Figure 2A and 2B). Token level F1 score is the F1 score of all tokens where the model is correct if the highest probability output label is the same as the true label. As the label and prediction are dependent on context, token level evaluation does capture the contextual performance of the model. Token level evaluation was chosen (as opposed to a metric that evaluates the entire sequences together) as we did not have enough full Sumerian-English/English-Sumerian phrases for sequence level evaluation. Overfitting was minimized as both training and validation loss trajectories were similar.

Experiment 3: Accuracy of the NER Model

Overall F1 score of the NER model was 96%. Evaluation was conducted on English translations of Sumerian, with entity class and their corresponding indices as labels. Recognition of the *royal* entity had lower precision – 67%. While the model correctly identified all instances (100% recall), it incorrectly labeled some non-*royal* entities as *royal*. This could be due to Sumerian's overall ambiguity or overlap of entities between *royal* and other classes.

Discussion and Conclusion

The work presented represents an exploration of how NLP techniques can assist with ancient language translation. Our results indicate that tokenization with Google SentencePiece with either BPE or Unigrams can produce relatively lossless subword tokenization, while retaining enough information to train an accurate translation model; but do not outperform the default delimiter-based tokenization. Our translation models also performed well with a token level F1 score of 94-96% depending on the translation direction. Although we minimized overfitting (as evidenced by similar trajectories of training and validation loss for each model) the high accuracy was likely due to lack of diversity within our corpus. NER model F1 score was high, but not unexpected; the model was evaluated on the ability to recognize and classify only entity names it has seen before. However, some entities belong to multiple categories, i.e. *anan* as both a settlement and geographic place. For these entities, their class is determined by context. The NER model's ability to use context to accurately identify the correct entity class sets it apart from a basic classifier.

Although this project has produced encouraging results, there are many limitations to this work and its applicability as a tool for academic linguists. First, our models only function on previously transliterated Sumerian, or in the case of the NER model, English translations of Sumerian transliterations. Transliteration of Sumerian Cuneiform is already an intensive process requiring detailed knowledge of both Sumerian Cuneiform and transliteration conventions – many of which are not standardized and actively evolving [18]. The inability to directly read and translate Sumerian Cuneiform to English limits the applicability of the translation models in the academic context, as the user must already have transliterated Cuneiform or know how to transliterate Cuneiform. Additionally, the ECTSL corpus is small and mostly consists of religious material (hymns, prayers) , philosophical and administrative texts (scribal training, moral and practical guidance, lexical compositions about the language itself), personal letters, and narrative storytelling (proverbs, folk tales, and other stories) [19]. Most of the notable characters are deities or rulers, and only 394 individual works make up the corpus. Although it is expected that an isolated language spoken thousands of years ago would have a limited corpus and our corpus is more varied than previous approaches to machine translation of Sumerian [7], the issues that come with a small and homogenous sample size are unavoidable; there are many phrases and ideas that will never have a translation in Sumerian, like "computer" or "estrogen birth control." Finally, our translation models have only been evaluated with simple categorical metrics with reserved test data from the corpus. These metrics are not sufficient to accurately characterize the performance of our translation models, especially on real world applications of text dissimilar from the training corpus. However, metrics such as BLEU [8] and METEOR [20] rely on matching specific sequences of words, presenting challenges for languages with flexible syntax like Sumerian. Additionally, the presence of multiple valid translations for a single lemma can affect these scores. Other limitations include the inability to translate numerals, as numerals were removed from the training corpus to limit the scope and complexity of the project.

The goal of this project was to explore machine translation of low-resource ancient languages by designing a set of translation tools for academics who study Sumerian. We built a set of translation models that performed bidirectional translation with moderate accuracy and a named entity recognizer. We also examined the impact of SentencePiece tokenization schemes on tokenizer accuracy. This project had many challenges – a small, homogenous training corpus comprised of less than 400 pieces a currently unspoken ancient language with a complex syntax and few premade/pretrained tools – and many limitations; lack of sophisticated evaluation metrics, the need for the user to already have transliterated cuneiform, and lack of support for numerals.

Individual Contributions

Ethan Rogers - writing of the abstract, proposal, and final write-up. Created presentations, managed the project, made decisions, drove research narrative.

Indrajeet Aditya Roy – dataset and model development. Processing, cleaning, and structuring the ETCSL XML files to create training dataset. Translation and NER model development and evaluation. Restructuring codebase.

Emery Jacobowitz – resident Sumerian expert, dataset preprocessing and tokenization, conducted an extensive review of the ETCSL literature, reviewed project methodology to ensure they were linguistically sound.

Jiajun Wang – Initial research on evaluation metrics.

References

[1] W. W. Hallo, “On the Antiquity of Sumerian Literature,” in *The World’s Oldest Literature*, Brill, 2009, pp. i–xxxii. Accessed: Apr. 22, 2024. [Online]. Available: https://brill.com/display/book/9789047427278/Bej.9789004173811.i-768_001.xml

[2] Harriet Crawford, *The Sumerian World*. New York, NY, USA: Routledge, 2013.

[3] B. Jagersma, “A descriptive grammar of Sumerian,” Leiden University, 2010. Accessed: Apr. 22, 2024. [Online]. Available: https://hdl.handle.net/1887/16107

[4] A. H. (Archibald H. Sayce and Society for Promoting Christian Knowledge (Great Britain). General Literature Committee, *The archæology of the cuneiform inscriptions*. London : Society for Promoting Christian Knowledge ; New York : E. S. Gorham, 1908. Accessed: Apr. 22, 2024. [Online]. Available: http://archive.org/details/archaeologyofcun00sayc

[5] “PSD: home page.” http://psd.museum.upenn.edu/ (accessed Apr. 22, 2024).

[6] É. Pagé-Perron, M. Sukhareva, I. Khait, and C. Chiarcos, “Machine Translation and Automated Analysis of the Sumerian Language,” in *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, B. Alex, S. Degaetano-Ortlieb, A. Feldman, A. Kazantseva, N. Reiter, and S. Szpakowicz, Eds., Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 10–16. doi: 10.18653/v1/W17-2202.

[7] R. Punia, N. Schenk, C. Chiarcos, and É. Pagé-Perron, “Towards the First Machine Translation System for Sumerian Transliterations,” in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds., Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 3454–3460. doi: 10.18653/v1/2020.coling-main.308.

[8] E. Reiter, “A Structured Review of the Validity of BLEU,” *Comput. Linguist.*, vol. 44, no. 3, pp. 393–401, Sep. 2018, doi: 10.1162/coli_a_00322.

[9] Y. Liu, C. Burkhart, J. Hearne, and L. Luo, “Enhancing Sumerian Lemmatization by Unsupervised Named-Entity Recognition,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, R. Mihalcea, J. Chai, and A. Sarkar, Eds., Denver, Colorado: Association for Computational Linguistics, May 2015, pp. 1446–1451. doi: 10.3115/v1/N15-1167.

[10] S. Robinson, G. Aumann, and S. Bird, “Managing fieldwork data with toolbox and the natural language toolkit,” *Lang. Doc. Conserv.*, vol. 1, no. 1, pp. 44–57, 2007.

[11] A. Hamplová, D. Franc, J. Pavlíček, A. Romach, and S. Gordin, “Cuneiform Reading Using Computer Vision Algorithms,” in *Proceedings of the 2022 5th International Conference on Signal Processing and Machine Learning*, in SPML ’22. New York, NY, USA: Association for Computing Machinery, Oct. 2022, pp. 242–245. doi: 10.1145/3556384.3556421.

[12] M. Mahmood, F. M. Jasem, A. A. Mukhlif, and B. AL-Khateeb, “Classifying cuneiform symbols using machine learning algorithms with unigram features on a balanced dataset,” *J. Intell. Syst.*, vol. 32, no. 1, Jan. 2023, doi: 10.1515/jisys-2023-0087.

[13] Jeremy A. Black *et al.*, “The Electronic Text Corpus of Sumerian Literature.” Oxford, 2006. Accessed: Apr. 17, 2024. [Online]. Available: https://etcsl.orinst.ox.ac.uk/

[14] M. Mager, A. Oncevay, E. Mager, K. Kann, and T. Vu, “BPE vs. Morphological Segmentation: A Case Study on Machine Translation of Four Polysynthetic Languages,” in *Findings of the Association for Computational Linguistics: ACL 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 961–971. doi: 10.18653/v1/2022.findings-acl.78.

[15] V. M. Sánchez-Cartagena, J. A. Pérez-Ortiz, and F. Sánchez-Martínez, “Understanding the effects of word-level linguistic annotations in under-resourced neural machine translation,” in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 3938–3950. doi: 10.18653/v1/2020.coling-main.349.

[16] “google/sentencepiece.” Google, Apr. 22, 2024. Accessed: Apr. 22, 2024. [Online]. Available: https://github.com/google/sentencepiece

[17] “spaCy’s NER model · spaCy Universe,” *spaCy’s NER model*. https://spacy.io/universe/project/video-spacys-ner-model (accessed Apr. 19, 2024).

[18] Jeremy A. Black, “Transliteration Principles,” ECTSL, Oxford Univeristy, 2005. Accessed: Apr. 19, 2024. [Online]. Available: https://etcsl.orinst.ox.ac.uk/edition2/technical.php

[19] “ETCSLliterature.” https://etcsl.orinst.ox.ac.uk/edition2/literature.php (accessed Apr. 19, 2024).

[20] S. Banerjee and A. Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,” in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss, Eds., Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 65–72. Accessed: Apr. 22, 2024. [Online]. Available: https://aclanthology.org/W05-0909