Integrating Automated Segmentation and Glossing into Documentary and Descriptive Linguistics

Sarah Moeller   Mans Hulden
University of Colorado Boulder
first.last@colorado.edu

Abstract

Any attempt to integrate NLP systems to the study of endangered languages must take into consideration traditional approaches by both NLP and linguistics. This paper tests different strategies and workflows for morpheme segmentation and glossing that may affect the potential to integrate machine learning. Two experiments train Transformer models on documentary corpora from five under-documented languages. In one experiment, a model learns segmentation and glossing as a joint step and another model learns the tasks into two sequential steps. We find the sequential approach yields somewhat better results. In a second experiment, one model is trained on surface segmented data, where strings of texts have been simply divided at morpheme boundaries. Another model is trained on canonically segmented data, the approach preferred by linguists, where abstract, underlying forms are represented. We find no clear advantage to either segmentation strategy and note that the difference between them disappears as training data increases. On average the models achieve more than a 0.5 F_1-score, with the best models scoring 0.6 or above. An analysis of errors leads us to conclude consistency during manual segmentation and glossing may facilitate higher scores from automatic evaluation but in reality the scores may be lowered when evaluated against original data because instances of annotator error in the original data are "corrected" by the model.

1 Introduction

This paper examines the direct effects of variations in research design when integrating machine learning into the morphological analysis and annotation of endangered languages. Morpheme segmentation and glossing are traditionally the first tasks undertaken by linguists after documenting a language's sound system. Both tasks provide essential linguistic information. Segmenting words into morphemes clarifies relationships between various word forms and can reduce confusion caused by data sparsity in NLP models. Glosses make implicit linguistic structures explicit and accessible for analysis and they can be leveraged to improve NLP models in low-resource settings, such as for machine translation (Shearing et al., 2018; Zhou et al., 2020). Therefore, automating these tasks with NLP systems and integrating those systems into the documentary and descriptive workflow is important to both linguistics and NLP.

When we bring together two disciplinary fields for mutual benefit, different expectations or accepted conventions are also brought together. The issues that this paper addresses stem from conventional methods in natural language processing (NLP) and linguistic analysis. The methods are based or have led to differing expectations which raise potentially conflicting issues. For example, it is generally expected in NLP that textual data will be orthographic representations and that the goal is to process that form, whereas linguists may prefer to work with phonetic representations and see their goal to process underlying linguistic forms. These differences can make the interdisciplinary collaboration unnecessarily slow or confusing. When the differences affect overall research design, it is easy to simply choose one or the other convention without testing which choice might actually benefit the task at hand or be more efficient for long-term goals. This paper studies and compares the short-term affect of two pairs of differing expectations which have arisen during the authors' research.

The first study asks *whether morpheme segmentation and glossing should be done jointly or sequentially*. In other words, are NLP systems more accurate when trained to do these two tasks sepa-

Proceedings of the 4th Workshop on the Use of Computational Methods in the Study of Endangered Languages: Vol. 1 Papers, pages 86–95,
Online, March 2–3, 2021.

rately or when trained to do them jointly? Instead of arbitrarily choosing one or the other method, we could test whether one approach gives more accurate results than the other. If the sequential approach is more accurate, then linguists might consider adjusting their workflow in order to gain optimal benefit from the NLP system, but if the joint task approach performs better, then perhaps NLP would benefit by adjusting experiments to match the linguists' expectations about data annotation methods.

This second issue we investigate is *how morpheme segmentation strategies affect NLP performance*. Linguistic theory assumes the existence of underlying morpheme forms and generally the goal to discover these forms determines research design. The morphemes are often represented in their theoretical, underlying forms, which also allows orthographic changes triggered by surrounding phones to be ignored. This contrast with surface segmentation which simply inserts morpheme breaks in the orthographic representation. The two segmentation strategies are compared in (1) where the first two surface letters of each word in (1a) are represented by identical canonical segments in (1b). Since NLP almost always deals with orthographic representations, its systems are trained to perform surface segmentation almost exclusively. In practice, both strategies are encountered during language documentation and description, the initial strategy depending in part on software tools. For example, the older, but still popular, Toolbox[1] allows surface segmentation whereas ELAN (Auer et al., 2010) supports both but as separate tasks, while FLEx (Baines, 2018) requires surface segmentation but facilitates simultaneous canonical segmentation. It might seem reasonable that linguists who want to integrate automated assistance would adjust their strategy to match NLP expectations. But without testing, are we sure that NLP systems perform better at surface or at canonical segmentation?

(1)	a.	il-legal	in-capable	im-mature
	b.	in-legal	in-capable	in-mature
	c.	NEG-legal	NEG-capable	NEG-mature

This paper describes experiments that test results of Transformer models (Vaswani et al., 2017) trained on segmented and glossed data and then compare those results between a joint and sequential approach to segmentation and glossing and between a surface and canonical strategy to segmentation. After a review of related literature in § 2, § 3 introduces the data used by the models that are described in § 4. The experiments are described in § 5 and results are presented in § 6 and analyzed in § 7.

2 Related Work

Many NLP models have been applied to segmentation and glossing of low-resource languages. Automatic morpheme segmentation was introduced by Harris (1970) and much segmentation research since then has implemented this in an unsupervised fashion (Goldsmith, 2001; Creutz and Lagus, 2002; Poon et al., 2009). This is probably motivated by the difficulty of finding high quality amounts of segmented data that is needed for supervised learning. A recent supervised segmentation experiment (Ansari et al., 2019) had to first manually segment a Persian corpus before being able to conduct the experiment.

NLP experiments with low-resource languages often treat segmentation and glossing as separate tasks. Their approach seems to assume that the two tasks are performed sequentially and that it is reasonable to expect morpheme segments to be available before glosses. However, our field experiences indicates that it is not uncommon for linguists to segment a morpheme and gloss it immediately.

Glossing-only experiments make the assumption that the data is already segmented into morphemes or that it does not need to be segmented. McMillan-Major (2020) trained conditional random field (CRF) systems to produce a gloss line for several high-resource languages and three low-resource languages. The systems incorporated predictions made directly from the segmented line and predictions made with information from the free translation line that was enriched with INTENT (Georgi, 2016). The low-resource language data came from field projects, as does the data in this paper. Both McMillan-Major and Samardzic et al. (2015) used information from other lines of interlinearized texts such as translation and part-of-speech tags, whereas our work assumes the texts have not yet been annotated with any other information.

In general, joint learning is characterized by

[1] https://software.sil.org/toolbox/

training on different types of information and is based on the intuition that one type of linguistic knowledge (e.g. syntax) can improve results in another domain (e.g. morphology) (Goldsmith et al., 2017). Joint learning of segmentation and glossing, or labeled segmentation, is less common but has been successful in low-resource languages (Cotterell et al., 2015; Moeller and Hulden, 2018), usually with non-neural models. The authors' previous work on Lezgi (Moeller and Hulden, 2018) used the same corpus as the current work and compared sequential vs. joint models as well as feature-based vs. deep learning models. The reported F_1-scores were nearly .9. However, a direct comparison between the two studies cannot be made because the previous work only segmented and glossed affixes while the current work includes root and affixes.

The segmentation strategies that an NLP project implements may depend on available data or the type of learning model employed. Unsupervised learning of morphology naturally leans towards surface segmentation. Supervised models depend on annotated data provided by linguists and preprocessed to reduce inconsistencies. Moeller and Hulden (2018) trained a joint system with good results on canonical morphemes in language with little allomorphy or morphophonological processes. In languages with more complicated morphophonology and allomorphy—including null morphemes that must be "segmented" and glossed, or circumfixation—the effect of canonical segementation may be unclear.

3 Data

The selected data represent a range of documentary and descriptive projects that manually interlinearized several texts. Each project's unique priorities and workflow resulted in different amounts of data and percentages of segmented and glossed tokens, as shown in Table 1. We selected only projects that interlinearized with FLEx, since the software always includes both surface and canonical segmentations. Less effort was made to represent various typological features, geographic areas, or language families. The corpora were shared in the form of backup `flextext` XML files.[2]

Language	Tokens	Seg/Gloss	
Alas	4.5k	3,775	85%
Lamkang	101k	49,465	49%
Lezgi	14k	13,262	94%
Manipuri	12k	11,904	98%
Natügu	16.5k	13,925	84%

Table 1: The approximate total token considers multiple word expressions (when parsed as such) as single tokens. The percentage and total number of tokens that are both segmented (canonical and surface) and glossed are shown.

Alas [btz] (Alas-Kluet, Batak Alas, Batak Alas-Kluet) is an Austronesian language spoken by 200,000 people on the Indonesian island of Sumatra (Eberhard et al., 2020). The selected corpus is from the Alas dialect and features reduplication, infixation, and circumfixation.

Lamkang [lmk] is a Northern Kuki-Chin (Tibeto-Burman) language with an estimated 4 to 10 thousand speakers, primarily in Manipur, India but also in Burma (Thounaojam and Chelliah, 2007). It tends toward agglutination with stem-stem patterns that signal syntactic categories and some bound morphemes that are written as separate words. The data is accessible through the Computational Resources for South Asian Languages (CoRSAL) digital archive at the University of North Texas.[3]

Lezgi [lez] (Lezgian) is a highly agglutinative language belonging to the Lezgic branch of the Nakh-Daghestanian (Northeast Caucasian) family. It is spoken by over 400,000 speakers in Russia and Azerbaijan. It features suffixing morphology with one rare negation prefix.[4]

Manipuri [mni] (Meitei, Meetei) is a Tibeto-Burman language spoken by nearly two million people, primarily in the state of Manipur, and is one of India's official languages. It nonetheless has been classified as vulnerable to extinction by UNESCO (Moseley, 2010). It is a tonal language with weakly suffixing, agglutinative morphology (Chelliah, 1997). The data is stored at CoRSAL.[5]

[2] Rights holders gave informed consent to use the data for this research.

[3] `https://digital.library.unt.edu/explore/collections/SAALT`

[4] The Lezgi is currently being deposited at the SIL Language and Cultures Archives.

[5] `https://digital.library.unt.edu/explore/collections/MDR`

Natügu [ntu] belongs to the Reefs-Santa Cruz group in the Austronesian family spoken by about 4,000 people in the Temotu Province of the Solomon Islands. It has mainly agglutinative morphology with complex verb structures (Åshild Næss and Boerger, 2008). The data is stored at SIL Language & Culture Archives.[6]

Gold standard data was assembled by filtering out tokens that were not completely segmented and glossed as far as could be determined automatically by assuring that the surface, canonical, and gloss lines aligned with each other. Morpheme boundary markers such as hyphens and equal signs were preserved to distinguish clitics from bound morphemes and to indicate relative ordering of morphemes (i.e. pre-/suf/infixing); angle brackets (⟨ ⟩) were used to denote circumfixes.

4 Models

All tasks are treated as a problem of converting an input sequence of characters $\mathbf{x} = (x_1, \ldots, x_n)$ to an output sequence of labels $\mathbf{y} = (y_1, \ldots, y_n)$. The output sequence of labels indicate the (canonical or surface) morpheme and/or the morpheme's gloss. Since Conditional Random Fields (CRF) (Lafferty et al., 2001), the state-of-art non-neural sequence labeling model, has not performed as well as neural models on low-resource sequence-to-sequence tasks since about 2016 (Liu and Mao, 2016), we selected the Transformer (Vaswani et al., 2017) as our model. The Transformer is a supervised deep learning system that has achieved promising results for NLP in low-resource languages (Abbott and Martinus, 2018; Martinus and Abbott, 2019). It is a stateless encoder-decoder model that uses additional attention layers to boost speed and performance. We used the Fairseq (Ott et al., 2019) implementation with the modifications and code described by Wu et al. (2020) which have been successful in low-resource character-level morphological tasks.[7]

5 Segmentation and Glossing Experiments

The experiments assume access to field data that has only been segmented and glossed. Therefore, no other information was leveraged from the interlinearized glossed texts or elsewhere. The data was arranged so as to accommodate both joint and sequential learning. That is, after withholding ten percent of the corpus as a test set, the remaining data was split into two equal training sets. It is assumed that segments and glosses exist for the first part which can be used for training in the sequential system, but not for the second part. Ten percent of each part was used as a development set. For easier comparison, the joint model was trained on only one part, the same part used for training the segmentation step in the sequential system. One additional experiment was run with the joint model that trained on both parts together, minus the held-out data. For each experiment, a ten-fold cross validation was run.

5.1 Joint versus Sequential System

The first experiment tested whether joint or sequential segmentation and glossing is a better approach to interlinearization when integrating automated assistance. Joint segmentation assumes that segmented data without glosses is unlikely because identifying a morpheme usually means there has already been an identification of the morpheme's meaning. Joint segmentation requires the model to learn the morpheme boundary and gloss simultaneously for each segment. The sequential system–glossing after segmenting the whole text—assumes that segmentation is easier to do by hand or that unsupervised segmentation tools such as Morfessor (Smit et al., 2014) are available for low-resource languages.

For joint learning, the input is a character-level representation of a word, shown in (2a). Each character is treated as as separate symbol by the model. The output is a sequence of labels, one label per morpheme, shown in (2b). The label combines the morpheme's shape and gloss. The combination allows the system to perform segmentation and glossing simultaneously.

(2) a. **IN:** t a x e s
b. **OUT:** tax#levy -es#PL

The sequential system trains two models: one model learns morpheme segments and the other learns to gloss the predicted morphemes. In the sequential system the first equal part of the data was used for the segmentation step and its output was the training input for the glossing step. The

[6] https://www.sil.org/resources/search/language/ntu

[7] Code available here: github.com/shijie-wu/neural-transducer

output to the first model is a sequence of segments only, shown in (3b).

(3) a. **IN:** t a x e s

b. **OUT:** tax -es

The output of the segmentation model is used as input to the second model, as shown in (4a.) The glossing model then outputs the predicted glosses, shown in (4b).

(4) a. **IN:** tax -es

b. **OUT:** levy PL

5.2 Segmentation Strategy

The second experiment compares the Transformer's performance when trained on different segmentation strategies. Both systems described above are trained on both strategies. Canonical segmentation gives more information about a language's underlying morphological structure. At the same time, it reduces the number of unique labels in languages that reflect allomorphy and morphophonological processes in the orthography. On the other hand, surface segmentation does not require computational models to learn allomorphy or morphophonology (Goldsmith et al., 2017) and does not provide a thorough analysis of the language's morphology by annotators. It simply divide strings of text into segments known as "morphs" (Virpioja et al., 2011) without regard to potential relationships between the segments.

The intention of this study is not to provide a direct comparison, since technically the corpora of surface and canonical segments are different datasets. The study assumes that if one strategy was conducted first, then the other type of segmentation might be more easily learned from it. For example, if a corpus could be surface segmented very quickly with very high accuracy based on initial hypotheses of morpheme shapes, then having the predicted surface segments for the whole corpus might make the discovery of canonical, underlying morphemes easier and faster for linguists, as well as matching a common expectation in NLP.

The difference in the methodology of the two strategies is their outputs. Their input does not change and it is the same as the models described in section 5.1. The output for surface segmentation is shown (5a), and the corresponding output for canonical segmentation is in (5b).

	Surface		**Canonical**	
	Joint	**Seq**	**Joint**	**Seq**
Alas	.4280	**.4565**	.5166	**.5291**
Lamkang	.7091	**.7391**	.5414	**.5785**
Lezgi	.5489	**.6062**	.4993	**.5371**
Manipuri	.4719	**.5067**	.6401	**.6675**
Natügu	**.5423**	.5263	.6083	**.6335**
Average	.5400	.5670	.5011	.5895

Table 2: F_1-scores of Transformer joint and sequential models on both segmentation strategies. Scores are an average across a 10-fold cross-validation. The bottom row shows the average score across all languages.

(5) a. **SURFACE:** tax#levy -es#PL

b. **CANON.:** tax#levy -s#PL

In addition to the alternation between surface morphs and underlying morpheme representations, the data was handled slightly differently for the two strategies. The most obvious difference is the handling of circumfixes. Surface representation only preserves the ordering of morphs and does not require knowledge of morpheme types, so the two parts of each circumfix were treated as two different prefix and suffix morphs. Canonical segmentation represents the circumfix as a single morpheme that repeats before and after the stem. These changes are shown in (6).

(6) a. **SURFACE:** ke- STEM -en

b. **CANON.:** ke⟨⟩en- STEM -ke⟨⟩en

6 Results

Performance was evaluated by a cross-validation on ten training and development sets that were randomly split from the part of the data used for each experiment. The system predictions were automatically evaluated against the gold standard. Scores were calculated as a micro-average on all labels, independent of word accuracy. Since the system may predict more or fewer labels for a word, both precision and recall are calculated. Table 2 compares the average F_1-scores across a 10-fold validation. For joint learning, the scores indicate morphemes that were correctly segmented and glossed. For the sequential system, the score is a weighted average of the scores from both the segmentation and glossing models.

6.1 Joint vs Sequential Results

Overall, sequential learning does better than joint learning, but the differences are not great. The maximum improvement is less than 0.06 points on Lezgi [lez]. The best models achieved over 0.60 F_1 on all but the smallest corpus. Lamkang [lmk], which has the largest number of tokens by far, achieved over 0.70 average F_1 score.

The performance on the Natügu data is the only case where the sequential system is not consistently an improvement over the joint system. When considering word-level accuracy, Natügu joint learning outperformed sequential learning on canonical segmentation. Interestingly, it also has the smallest change in the number of unique labels between surface and canonical segmentation (an increase of 14 labels, compared to next lowest of 46). With so few languages, it is difficult to say whether the relative number of unique labels affect the relative performance when trained on surface vs. canonical segmentation. More corpora should be included for this question to be explored further.

6.2 Surface vs. Canonical Results

When half of the total data is used, the comparison of surface and canonical segmentation paints a less clear picture. The differences when going from surface to canonical segmentation are shown in Table 3. The general trend when comparing segmentation strategies is that languages with a higher ratio of unique labels to total tokens do better with canonical segmentation. The differences are quite small for Alas [btz], Lezgi, and Natügu [ntu]. The biggest differences are found in Lamkang and Manipuri [mni], but their improvement goes in opposite directions. Surface segmentation gives higher scores for Lamkang data while Manipuri has higher scores with canonical. Interestingly, these two languages have the largest difference of the number of unique labels between surface and canonically segmented data. In Lamkang and Manipuri training data, the average number of unique joint labels increased by over 500 and 400, respectively, and in the segmentation step of the sequential system the number of segments increased by over 350. In the other languages the largest average increase of labels is 88 but usually the differences are less than 15. Since Lamkang and Manipuri belong to the same family, it is possible that significant differences in segmentation strategies are due to characteristics of their familial morphological structure, but it could be due to other factors such as idiosyncratic choices in the orthographic representation.

	Joint	**Seq**
Alas	-.0886	-.0726
Lamkang	.1677	.1606
Lezgi	.0496	.0691
Manipuri	-.1682	-.1608
Natügu	-.0660	-.1072

Table 3: Average F_1 differences between surface and canonical segmentation strategies. Positive scores mean surface segmentation outperformed canonical segmentation.

	Surface	**Canon**
Alas	.4280	.5166
Alas all	.6771	**.6902**
Lamkang	.7091	.5414
Lamkang all	.8547	**.8573**
Lezgi	.5489	.4993
Lezgi all	**.7834**	.7735
Manipuri	.4719	.6401
Manipuri all	.8693	**.8903**
Natügu	.5423	.6083
Natügu all	.8965	**.8995**

Table 4: Results the joint model with surface and canonical segmentation strategies when using half the training data compared to all training data.

The differences in the results in both joint and sequential systems are shown in Table 3. The effect of the segmentation strategy is roughly the same in both systems.

The segmentation strategies were also compared using all available data in the joint system. Table 4 shows the how doubling the training data affects the performance. Doubling the training data always improves F_1-scores by about .2 to .4 points. While the difference between the two strategies becomes less noticeable when the data is increased, canonical segmentation tends to outperform surface segmentation, but in all languages the difference between the strategies becomes almost negligible (less than .15 points).

7 Error Analysis

A closer look at the results reveals interesting patterns. One significant factor in system performance is sparsity of data. Unsurprisingly, most errors occur on rarer forms. Another factor is the amount of inconsistencies or errors in the manually annotated data. Annotation quality can amplify data sparsity.

Allomorphy and isomorphy (same character sequence, different meaning) caused repeated errors during the glossing step and joint learning, where it becomes quite obvious that the model must deal with multiple options. For example, the Lezgi suffix *-di*[8] has five possible glosses as shown by the joint labels in (7). These morphological phenomena are a moot issue during the segmentation step.

(7) -ди#ENT
-ди#DIR
-ди#ERG
-ди#OBL
-ди#SBST

Sometimes multiple glosses are not due to morphological structure, but because the same morph(eme) was given different glosses. For example, interchanging ‘be’ and ‘is’ and ‘COP’ for copular verbs or alternating between lexical glosses (e.g. ‘you’) and grammatical glosses (e.g. ‘2SG.ERG’). Sometimes different glosses appear because the item can be translated by different English words depending on the context. For example, one Lezgi word can be, and is, translated as ‘be’ in some context or ’happen’ in others. If alternative labels such as *bahaye#danger* and *bahaye#dangerous* are equally frequent, the model must choose randomly. Such inconsistency is to be expected from manual work and could be reduced with more automated assistance from machine learning.

Another pattern of errors is caused by tokens that were only partially segmented (and therefore, not correctly glossed). We knew that many such tokens were included in the gold standard data but there was no reliable way to eliminate them automatically. It is unclear how many exist in each corpus, although Alas and Natügu seem to have the least. Manipuri [mni] and Lezgi seem to have most incomplete segmentation. This became clear for Manipuri during another project when a language expert was asked to the correct the glosses for several inflected words. It appears that, in the data set, the annotators had been focused only on segmented and glossing certain morphemes on each word, leaving other affixes on the word unsegmented. The Lezgi data was annotated by a non-linguist who was trained to use FLEx and did not fully grasp Lezgi’s unique morphology or simply did not finish segmenting all words.

Many quality issues unpredictably increase the number of possible labels and amplify data sparsity. An example is repeated misspelling of glosses (e.g. apperance—appereance—appearance, fourty—forty). Other misspellings originate in transcription. In the Lezgi test data, over 50 misspelled or incorrectly segmented strings were found in the first 200 hundred unique segments, although a few spelling changes are representation of dialectical variations.

The results from the Alas corpus were quite good when compared to the much larger corpora. However, the errors are less predictable and more random. It seems likely that the small data set increased the noise to signal ratio and obscured general patterns. One noticeable confusion was caused by the canonical representation of circumfixes. This is shown in (8) where the model predicted a prefix *n-*. This prefix is a correct surface allomorph of the circumfix at that position.

(8) a.	**GOLD:**	n⟨⟩ken-	nindekh	-n⟨⟩ken
	OUTPUT:	n-	nindekh	-n⟨⟩ken

Nevertheless, error analysis shows that the models deal with data sparsity quite well. Even incorrect segments often have very similar character sequences to the correct choice, particularly when the difference is due to a change in the root vowel (e.g. dakhi ∼ dikhi). One of the most interesting errors, indicating the model’s strong ability to learn patterns even in the face of data sparsity, occurred in Lezgi. The transcribed oral speech has a few dozen codeswitched Russian words. The test data include one or two examples, and in one case the model substituted one codeswitched word with another codeswitched word.

Many errors noted during error analysis were not actually errors. Since the annotation was originally done by hand, sometimes by multiple annotators, the glosses varied due to misspellings or synonomous glossing choices (e.g. ’BE.PST’ vs. ’was’). There was a clear pattern in all datasets for one of the variants to be predicted rather than a random, unrelated label. These cases would not be considered errors by human annotators but were evaluated automatically as errors in the test

[8] In running text, Lezgi text is transliterated from the Cyrillic orthography for the reader’s convenience.

data. For instance, one Lezgi demonstrative pronoun was sometimes glossed as ‘these’ and sometimes as ‘this -ABS.PL’. In at least once case, the second (and more linguistically precise) analysis was predicted. Unfortunately, because we did not have access to language experts for every corpora, we were not able to normalize our scores based on this knowledge; however, in the future it may be useful to consider that the performance of models trained on field data may, for all practical purposes, be better than the initial scores indicate.

In other cases, the labels in the test data were evaluated as errors, but closer examination revealed that the original human annotation were incorrect in that particular instance and the predicted label was actually the best fit to the data. So, an human error had been “corrected”. Word instances that had been incorrectly segmented by the human annotators were sometimes correctly segmented by the model, although again these examples were evaluated as incorrect because they did not match the gold standard data. For Lezgi, these examples of “correction” by the model were more frequent in the sequential system, and may explain why biggest improvement by the sequential system over the joint system is found in the Lezgi data, which we know had many incorrect or incomplete segmentations. Again, due to the lack of language experts, we are unable to say whether this holds true for all corpora but this should be explored deeper in future research.

8 Discussion and Conclusions

This paper is aimed at smoothing the road to more interdisciplinary work with NLP and linguistics by articulating and examining the results of different research designs. Different research designs arise from different expectations or conventions in the two fields. Although they do not present barriers to mutually beneficial research, different expectations, such as in segmentation strategies, and different workflows, such as joint or separate segmentation/glossing, should not be dismissed when they arise. This paper tests the possible effects of these two differences.

The small difference between surface and canonical segmentation for three of the five languages suggests either strategy is a useful approach with minimal data, although this changes when data is increased in the joint model. Even though surface segmentation increases the number of labels in a dataset, this appears to be balanced by the by the abstract character of canonical morphemes, most noticeably by circumfixes. The fact that the difference almost disappears when the data size is doubled indicates that the question of segmentation strategy can be eliminated by simply annotating more data with whatever strategy suits the project at hand. However, larger differences on Lamkang and Manipuri corpora indicate that the reasons why segmentation strategies does sometimes differ in performance on the same corpsu should be explored more across other Tibeto-Burman languages. Testing the differences in related languages might indicate whether certain linguistics features influence the results of different segmentation strategy when integrating NLP systems.

The consistent improvement of the sequential system over joint learning may be a reason to consider separating segmentation and glossing tasks in order to leverage the higher accuracy of segmentations , and a more completely segmented corpus, when glossing the corpus. The strenght of the sequential system might be applied when a corpus cannot be completely segmented and glossed due to budget or time constraints. Instead, a strategy would be to prioritize segmenting and benefit from computational assistance when glossing.

Finally, these studies could serve as a foundation towards more efficient use of computational methods in linguistic analysis and annotation. This paper shows, for example, that the glossing-only model performs well even on inaccurate segmentation predictions and can even “correct” manual segmentation errors. The study presented here assumes that the model’s segmentation is not corrected by the language experts before training the glossing model. If a human-in-the-loop workflow was introduced to first correct segmentations, then the glossing-only model could improve even more. Such methodological considerations should be tested to see to what extent linguistic analysis and annotation of endangered language might benefit.

Finally, as McMillan-Major (2020) noted in glossing research, consistency of the annotations has a strong effect on system performance. This is most clearly seen in Lezgi which is known to be particularly noisy. Random strange characters were found at morpheme boundaries (e.g. ⋆ instead of -). The human annotators frequently segmented one pair of characters whenever it occurred because it matched a frequent suffix. Allomorphs were frequently glossed as if they were different morphemes, undoing the benefit of canonical segmentation. Finally, its unique case-stacking caused confusion both to the human annotator and to the system results, in particular because one morpheme with several semantically-motivated allomorphs is (incorrectly) glossed one way when it stands as a single case marker and glossed another way when it precedes additional case markers.

So what would happen if linguists emphasized quality over quantity? We can answer this question by comparing Lezgi to Alas. According to the accounts of the linguists involved, and evidenced by our experimental results, the Alas data was annotated much more consistently and meticulously. With a corpus one third the size of the Lezgi corpus, the Alas model performs almost equally well. It is possible but seems unlikely that this is due to differing morphological structure. Unlike Lezgi—which is overwhelmingly suffixing and has fairly limited morphophonological changes—Alas features prefixing, suffixing, circumfixing, and infixing with various morphophonological processes. The main difficulty for the Alas systems was the sparsity of stems, compared to oft-repeated affixes.

Interestingly, Alas showed the least marked preference between sequential and joint learning. This may indicate that higher consistency may eliminate the need to consider any change to segmentation/glossing workflow, but it should be investigated with further experiments focused on differences in annotation quality. Preferably these experiments would conducted on closely related languages to reduce effects due to different typology.

When considering low-resource settings, consistency for machine learning seems more important than data size, strategy, or workflow. Ruthless consistency is not something linguists have had reason to put high value on and it is not something to be expected by manual annotation, Consistency can be provided by machine learning integration, but ironically, supervised machine learning needs high consistency in annotated data before it can perform accurately enough to assist human annotators by increasing their speed or accuracy. Our best estimate of the accuracy threshold for practical integration of machine learning into annotation is 60% (Felt, 2012). This threshold on F_1-scores was soundly passed by Lamkang because it over 18k manually annotated tokens for training but it was barely reached by the corpora with 4.5k-5.5k tokens. However, the meticulously annotated Alas corpus came close to this threshold with only 1.5k training tokens. If linguists wish to successfully integrate machine learning into the documentation and description of underdocumented and endangered languages, then they must adopt from NLP an emphasis on highly consistent annotation.

References

Jade Z. Abbott and Laura Martinus. 2018. Towards neural machine translation for African languages. *arXiv:1811.05467 [cs, stat]*.

Ebrahim Ansari, Zdeněk Žabokrtský, Mohammad Mahmoudi, Hamid Haghdoost, and Jonáš Vidra. 2019. Supervised Morphological Segmentation Using Rich Annotated Lexicon. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 52–61, Varna, Bulgaria. INCOMA Ltd.

Eric Auer, Albert Russel, Han Sloetjes, Peter Wittenburg, Oliver Schreer, S. Masnieri, Daniel Schneider, and Sebastian Tschöpel. 2010. ELAN as flexible annotation framework for sound and image processing detectors. In *European Language Resources Association LREC 2010: Proceedings of the 7th International Language Resources and Evaluation*, pages 890–893. European Language Resources Association.

David Baines. 2018. An overview of FieldWorks and related programs for collaborative lexicography and publishing online or as a mobile app. In *Proceedings of the XVIII Euralex International Congress*, Ljubliana, Slovenia. Ljubliana University Press.

Shobhana Lakshmi Chelliah. 1997. *A Grammar of Meithei*, volume 17 of *Mouton Grammar Library*. Mouton de Gruyter, Berlin.

Ryan Cotterell, Thomas Müller, Alexander M. Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-markov models. In *CoNLL*, pages 164–174.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, pages 21–30. Association for Computational Linguistics.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2020. *Ethnologue: Languages of the World*, twenty-third edition. SIL International, Dallas, Texas.

Paul Felt. 2012. *Improving the Effectiveness of Machine-Assisted Annotation*. Phd thesis, Brigham Young University.

Ryan Georgi. 2016. *From Aari to Zulu: Massively Multilingual Creation of Language Tools using Interlinear Glossed Text*. PhD, University of Washington.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.

John Goldsmith, Jackson Lee, and Aris Xanthos. 2017. Computational learning of morphology. *Annual Review*, 3.

Zellig S. Harris. 1970. From phoneme to morpheme. In *Papers in Structural and Transformational Linguistics*, pages 32–67. Springer Netherlands, Dordrecht.

John Lafferty, Andrew McCallum, and Fernando C N Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289.

Ling Liu and Lingshuang Jack Mao. 2016. Morphological reinflection with conditional random fields and unsupervised features. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 36–40.

Laura Martinus and Jade Z. Abbott. 2019. A focus on neural machine translation for African languages. *ArXiv*, abs/1906.05685.

Angelina McMillan-Major. 2020. Automating gloss generation in interlinear glossed text. volume 3.

Sarah Moeller and Mans Hulden. 2018. Automatic glossing in a low-resource setting for language documentation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 84–93. Association for Computational Linguistics.

Christopher Moseley, editor. 2010. *Atlas of the World's Languages in Danger*, third edition. UNESCO Publishing, Paris.

Åshild Næss and Brenda H. Boerger. 2008. Reefs–santa Cruz as Oceanic: Evidence from the verb complex. *Oceanic Linguistics*, 47:185–212.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217. Association for Computational Linguistics.

Tanja Samardzic, Robert Schikowski, and Sabine Stoll. 2015. Automatic interlinear glossing as two-level sequence classification. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, Beijing, China. Association for Computational Linguistics.

Steven Shearing, Christo Kirov, Huda Khayrallah, and David Yarowsky. 2018. Improving low resource machine translation using morphological glosses. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*. Association for Machine Translation in the Americas.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Harimohon Thounaojam and Shobhana L. Chelliah. 2007. The Lamkang language: Grammatical sketch, texts and lexicon. *Linguistics of the Tibeto-Burman Area*, 30(1):1–212.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010, Long Beach, California, USA. Curran Associates Inc.

Sami Virpioja, Ville Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *TAL*, 52(2):45–90.

Shijie Wu, Ryan Cotterell, and Mans Hulden. 2020. Applying the transformer to character-level transduction. *arXiv:2005.10213 [cs.CL]*.

Zhong Zhou, Lori S. Levin, David Mortensen, and Alex Waibel. 2020. Using interlinear glosses as pivot in low-resource multilingual machine translation. *arXiv: Computation and Language*.