Automatic Glossing in a Low-Resource Setting for Language Documentation

Sarah Moeller and **Mans Hulden**
Department of Linguistics
University of Colorado
first.last@colorado.edu

Abstract

Morphological analysis of morphologically rich and low-resource languages is important to both descriptive linguistics and natural language processing. Field efforts usually procure analyzed data in cooperation with native speakers who are capable of providing some level of linguistic information. Manually annotating such data is very expensive and the traditional process is arguably too slow in the face of language endangerment and loss. We report on a case study of learning to automatically gloss a Nakh-Daghestanian language, *Lezgi*, from a very small amount of seed data. We compare a conditional random field based sequence labeler and a neural encoder-decoder model and show that a nearly 0.9 F_1-score on labeled accuracy of morphemes can be achieved with 3,000 words of transcribed oral text. Errors are mostly limited to morphemes with high allomorphy. These results are potentially useful for developing rapid annotation and fieldwork tools to support documentation of other morphologically rich, endangered languages.

1 Introduction

Thousands of languages lack documented data necessary to describe them accurately. In the early 1990s it was suggested that linguistics might be the first academic discipline to preside over the its own demise, since numbers indicated that as much as 90% of the world's languages would be extinct by the end of the 21st century (Krauss, 1992). Linguists quickly responded by developing methodology to record previously under- or undocumented languages (Himmelmann, 1998). Almost as quickly, they realized that unannotated data of a language that is no longer spoken is almost as inaccessible as an undocumented language. Language documentation and the initial descriptive work that often accompanies it is time- and labor-intensive work, but it is foundational to the study of new languages. It also benefits the community of speakers by supporting efforts to revitalize or maintain the language. Although the estimated number of languages in imminent danger of extinction has been reduced (Simons and Lewis, 2013), the task remains urgent.

Computational linguistics generally considers human annotation prohibitively expensive because it relies on linguistic expertise (Buys and Botha, 2016). However, employing this expertise has long been accepted practice in documentary and descriptive linguistics. Documentation data is not produced by a linguist alone; rather, it is created in close cooperation with native speakers who receive minimal training in general linguistics and software. The documentation work includes transcription of oral recordings, translation, then ends with, as descriptive work begins with, interlinearization (i.e. POS-tagging, morpheme segmentation, and glossing). The first task alone may takes an average of 39 times longer than the original recording, according to a recent survey of field linguists (CoEDL, 2017). No matter how many oral texts are recorded during a field project, time constraints often mean that only the annotations required to support a particular short-term goal are completed. For example, the data used in the current paper was collected by a linguist for his MA thesis. Since his topic was verbs, only the verbs were thoroughly annotated. More funds had to be found to hire another native speaker who could simultaneously

Proceedings of Workshop on Polysynthetic Languages, pages 84–93
Santa Fe, New Mexico, USA, August 20-26, 2018.

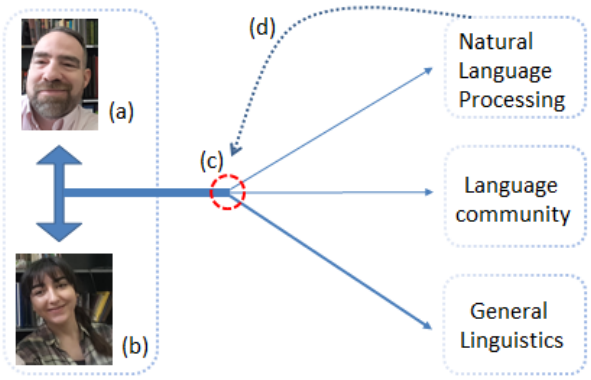

Figure 1: Flowchart of language data production. Descriptive linguists (a) collaborate with native speakers of a language (b) to produce documentary data for all subfields of linguistics, language development efforts by the community of speakers, and the extension of NLP tools to low-resource languages. A bottleneck of time-consuming annotation (c) keeps much of the data inaccessible to all but the community of speakers. The models described in this paper (d) attempt to employ semi-automated interlinearization to increase the trickle of data by .

learn and do basic linguistic analysis. Such manual work is slow and inevitably produces inconsistent annotations. It is noteworthy that many mistakes are not due to the difficulty of the task but because of its repetitive nature. In case marking languages, for example, morphemes marking subjects will be found in practically every clause and those marking objects, dative, or genitive arguments may be nearly as frequent. Only a small percentage of tokens contain unusual and interesting morphological forms. Thus, a large chunk of this highly time-consuming work is as monotonous to the annotator as it is uninformative to language science—in short, we are faced with a bottleneck (Holton et al., 2017; Simons, 2013).

After nearly 30 years of emphasis on increasing accessible documentation data, very few computational tools have been applied to this bottleneck. The most popular software packages designed for linguistic analysis, ELAN (Auer et al., 2010) and FLEx (Rogers, 2010), provide almost no automated aid for common, repetitive tasks, although FLEx does copy the annotator's work onto subsequent tokens if they are identical to previously analyzed tokens.

To address this problem, we apply machine learning models to two common tasks applied to documentation data: morpheme segmentation and glossing. The models use about 3,000 words of manually-annotated data that train sequence models to predict morpheme labels (and glosses). The goal is to achieve accurate results on more data in less time. A case study on *Lezgi* [lez] explores three issues as a first step toward integrating the models into linguistic analysis software. First, can the linguist and native speaker expect machine learning techniques to successfully widen the data bottleneck after they have manually annotated a few transcribed texts? Second, could a sequence labeler achieve reasonable accuracy using features that are generalizable to most languages? If a feature-based tool could be applied to several languages without tweaking features in a language-specific fashion, it would be accessible even to native speakers without linguistic skills who wish to create structured language data. At the same time, if high accuracy is achieved an agglutinative language like *Lezgi*, then minimal feature-tweaking could make the model equally successful on more fusional languages. Lastly, what might the errors in the case study indicate for typologically different languages?

Section 2 reviews related work. Section 3 introduces the case study and Section 4 describes the models used. The results are compared and analyzed in Section 5. Implications and future work are discussed in Section 6, before the conclusion in Section 7.

2 Related Work

Computational linguistics boasts a long history of successful unsupervised morphology learning (Goldsmith, 2001; Creutz and Lagus, 2005; Monson et al., 2007). One feature that unsupervised models share

is the requirement for large amounts of data. Ironically, languages with large amounts of data available likely already have published morphological descriptions and some interlinearized text, even though they may be considered low-resource languages. Under-documented languages rarely have sufficient data for a thorough morphological description. If unsupervised approaches were better known among documentary linguists, it might encourage them to archive more minimally-annotated data, which is a high-priority but rarely-met goal in language documentation.

For language documentation methods, more interesting approaches are those that augment small amounts of supervised data with unsupervised data. Supervised and semi-supervised learning generally requires less data to train and yields better results than unsupervised methods (Ahlberg et al., 2014; Ruokolainen et al., 2013; Cotterell et al., 2015; Kann et al., 2017). Several recent CoNLL papers (Cotterell et al., 2017) showed that very small amounts of annotated data could be augmented by exploiting either structured, labeled data, raw texts, or even artificial data. This assumes, however, that the data has already been processed in some way and made accessible. This paper looks at ongoing annotation and not generally accessible data.

This paper is most closely related to experiments on whether active learning could speed the time-consuming analysis of documentation data (Baldridge and Palmer, 2009; Palmer, 2009; Palmer et al., 2010). The experiments used field data processed with linguistic analysis software that are no longer supported. Our paper uses data from FLEx, currently one of the two most popular software modules for linguistic analysis. Earlier work has encountered complications because the analysis of certain morphemes has changed the middle of the project. This is normal—linguistic analysis, especially when a language has not been well-described before, is a dynamic, continually evolving process. Palmer et al. (2010) performed unsupervised morphological processing and semi-automatic POS tagging, combined with active learning. This seems to assume that the data is transcribed but not annotated in any way and would be most appropriate near the beginning of a documentation project. By contrast, we use supervised learning methods on data already tagged for parts of speech and assume that the annotation process is well underway. We also assume a fixed morpheme analysis applied consistently to the data which makes the methods more appropriate for later stages of a documentation project, or for a project that is willing to start with an less-than-accurate analysis and make bulk changes in FLEx. Most generally, previous work in the field has examined several factors affecting speed and accuracy of the annotators and the results seem to demonstrate that machine-supported annotation holds great promise for speeding language documentation. That promise lays the foundation for our case study.

3 Case Study: Lezgi

Three sequence labelers were tested on transcribed oral data from the Qusar dialect of *Lezgi* [lez]. *Lezgi* belongs to the Nakh-Daghestanian (Northeast Caucasian) family. It is spoken by over 600,000 speakers in Russia and Azerbaijan (Simons and Fennig, 2017). The endangered Qusar dialect in Azerbaijan differs from the standard written dialect in several ways, including a locative case morpheme borrowed from Azerbaijani that is used alongside the native inessive (locative) case morpheme with the same meaning. The dialect also has freer word order. *Lezgi* is an highly agglutinative language with overwhelmingly suffixing morphology. Fourteen noun cases are built by case-stacking, a characteristic of Nakh-Daghestanian languages. Case-stacking is characterized by composing a case inflection by a sequence of morphemes instead of a unique morpheme for each case. A simplified example of Lezgi case-stacking is shown in Table 1. Case-stacking morpheme sequences can be de-constructed into individual agglutinating morphemes, or, since the semantics of the morphemes are not entirely compositional, the sequence can be viewed as a single, fusional morpheme. Verbal inflectional morphology is no less complicated, with 22 base affirmative forms, corresponding negative forms, and an often suppletive imperative stem. From these finite forms, affirmative and negative participles are formed, as well as secondary verb forms that communicate adverbial meanings or non-indicative moods.

The aim of this case study is to assist and speed human annotation of the documentation data. Our original goal was to perform segmentation and glossing with at least 80% accuracy. This goal is inspired by the Pareto Principle—the idea that 20% of one's effort produces 80% of one's results, and *vice*

itim-di	SG.ERG 'the man'	itim-ar	ABS-PL 'men'
itim-di-q	SG.POSTESSIVE 'behind the man'	itim-di-q-di	SG.POSTDIRECTIVE 'to behind the man'
itim-ar-di-k	PL-ADESSIVE 'at the men'	itim-ar-di-k-ay	PL-ADELATIVE 'from the men'

Table 1: An example of case-stacking on the Lezgi noun *itim* 'man'. Absolutive (ABS) case and singular number (SG) are unmarked. The plural suffix (PL) attaches directly to the noun stem. The ergative suffix (ERG) attaches in the second slot after the stem. Other cases add suffixes to the ergative morpheme (oblique stem (OBL) cf. Haspelmath (1993, p.74). The elative and directive meanings are added to the fourth slot after the stem. The semantics are only partially compositional. In the largest possible sequence (postdirective and adelative), the final (directive) *-di* and (elative) *ay* suffixes add directed-motion meaning to the penultimate locative (-essive) morphemes *k* or *q*, but the previous (ergative) morpheme seems to serve a purely grammatical purpose.

Figure 2: Interlinearization in FLEx. Lezgi uses the Cyrillic alphabet. Segmentations are on the second line; glosses on the third. POS tags are below the glosses. The work is almost completely manual in FLEx. The goal is to complete the 2nd and 3rd lines automatically.

versa. A baseline that segmented correctly but assigned morphemes the majority label would perform at approximately 65% accuracy.

Data Ten texts amounting to a little over 3,000 words were excerpted from a small corpus of transcribed oral narratives. Of the 3,000 words, only nominals, pronouns, and verbs were morphologically analyzed. Every word had been tagged for part of speech. A linguist had provided morpheme glosses for all verbs. Other parts of speech were only partially glossed or segmented, if at all. A native speaker of the dialect finished segmenting the morphemes and glossed all affixes. The annotator often skipped core arguments with simple morphology, such as subjects or the extremely common aorist verbs, perhaps because the forms were so repetitive. She was more likely to annotate morphologically complex, but less common, tokens. Her initial annotations varied a great deal in quality, but once she identified morpheme boundaries, it was possible to refer to the descriptive grammar (Haspelmath, 1993) and make the annotations consistent. It seemed reasonable to expect that a native speaker educated in another language could quickly learn to recognize basic parts of speech in her own language, so the models assume that POS tags will exist in documentation data. The *Lezgi* data included two exceptions that are not basic parts of speech. Participles and demonstrative pronouns are more abstract than the general category of pronouns and verbs but these distinctions were kept simply because they had already been consistently annotated. After the linguist, native speaker, and author(s) each reviewed the gold standard annotations, all but three inflectional affixes had been accurately identified. These three were labeled UNK.

In our work, all but the neural model assume that (1) the data has been analyzed in FLEx, as shown in Figure 2, and exported as a FlexText XML format, (2) words have been tagged for part of speech, (for the case study - verb, participle, adjective, adverb, noun/proper noun, particle, pronoun, demonstrative pronoun, and postposition), (3) morpheme segmentation and glosses are consistent, and (4) all affixes, but not stems, are glossed. Inflectional morphemes are a closed class so the models could be easily trained to gloss them (e.g. ERG = ergative case, PST = past tense, etc.). Stems, however, are an open class, so the models were trained merely to recognize them as "stems". All characters that are not part

Figure 3: Excerpt from FLExText XML format. It shows the morpheme breaks of one word consisting of a root morpheme followed by a plural suffix. The POS tag is attached at the word level.

of a word (e.g. digits and punctuation) were eliminated in pre-processing.

Pre-processing the data showed that even the most careful annotation team will make mistakes, even on a corpus as small as 3,000 words. A few POS tags and affix glosses were still missing, and others were incorrectly labeled. Non-linguist annotators may use slightly different labels for the same morpheme. As long as the computational linguist has expert knowledge of the language, missing glosses can be corrected as a debugging step. For incorrect labels, printing out tags allowed a linguist to spot check annotations and check if, for example, the distribution of POS tags appeared unusual.

Features Most of the extracted features generalize to all languages. Certain features, such as the number of surrounding letters viewed, are specific to *Lezgi*. Affixes in the language are rarely more than 3 letters long, so the models viewed only the surrounding 1–4 letters to ensure that at least one letter in the immediately surrounding morphemes was seen. However, the average length of a morpheme can be automatically calculated from the training data for any language. The features include an assumption that a unit labeled as "phrase" in FLEx is equivalent to a complete clause in the language. In reality, some "phrases" contain more than one sentence, some contain only a sentence fragment. This makes the word position feature inaccurate. The word position feature is the only feature customized to *Lezgi*. It is measured from the end of the phrase to take into account the language's strong tendency for verb-final word order. Other features, included position of the letter in the word, and, of course, POS tags taken from the data.

4 Model Description

This section describes three models that perform supervised morphological segmentation and labeling on limited data. All three models expect 2,000–3,000 words of cleanly annotated data. The first two expect the data to be annotated with POS tags.

4.1 Conditional Random Field

We use a linear-chain Conditional Random Field (CRF) (Lafferty et al., 2001) to train a sequence model where the input consists of individual characters and the output of a BIO-labeling (Ramshaw and Marcus, 1999) of the sequence, i.e. we treat this as a labeling problem of converting an input sequence of letters $\mathbf{x} = (x_1, \ldots, x_n)$ to an output sequence of BIO-labels $\mathbf{y} = (y_1, \ldots, y_n)$.

BIO-labeling In the training data, each letter is associated with a Beginning-Inside-Outside (BIO) tag—a type of tagging where each position is declared either the beginning (B) of a chunk or morpheme, inside (I) or outside (O). The BIO tags are specific to each type of morpheme. BIO tags include (1) the morpheme type for stem morphemes (e.g. B-stem) or (2) affix glosses (e.g. I-DAT for a non-initial letter of a morpheme marking dative case). This combination of BIO tags and specific labels allows the system to perform segmentation and labeling/glossing simultaneously. For example, in tagging the word ава, with the morphemes й (PTP), and ди (SBST) the representation would be as follows:

а	в	а	й	д	и	**input**
B-stem	I-stem	I-stem	B-PTP	B-SBST	I-SBST	**output**

CRF model We model the conditional distribution of the output BIO-sequence $\mathbf{y}$, given the input $\mathbf{x}$ in the usual way as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z}\exp\Big(\sum_{i=1}^{n}\phi(y_{i-1}, y_i, \mathbf{x}, i)\Big) \quad (1)$$

where ϕ is our feature extraction function which can be expressed through a sum of k individual component functions

$$\phi(y_{i-1}, y_i, \mathbf{x}, i) = \sum_k w_k f_k(y_{i-1}, y_i, \mathbf{x}, i) \quad (2)$$

Here, Z is the "partition function" which normalizes the expression to a proper distribution over all possible taggings given an input. We use the *CRFsuite* (Okazaki, 2007) implementation together with a Python API.[1]

The training parameters used L-BFGS optimization (Liu and Nocedal, 1989) and Elastic Net regularization, i.e. a linear combination of L_1 and L_2 penalties. Maximum iterations for early stopping were set at 50.

4.2 Segmentation and Labeling Pipeline: CRF+SVM

Since the subtask of morpheme segmentation is presumably much easier than joint segmentation and labeling, we also experimented with a pipeline model that would first segment and then label, where we could use a richer set of contextual features for the subsequent labeling process. Here, the CRF is employed only for segmentation and is used as described above but without the morpheme specific labels. After predicting BIO tags at the character-level, characters are combined into predicted morpheme strings. We then train a multi-class linear Support Vector Machine (SVM)[2] which classifies the segmented morphemes with the specific labels ("stem" or individual affix glosses). This allows the SVM to use surrounding morphemes as features (though not future labels). The SVM is trained on the concatenated features of every letter in each predicted morpheme but only the morpheme labels of the predicted initial letter.

4.3 Neural Model

As the currently strongest performing models for the related task of morphological inflection (Cotterell et al., 2017; Kann et al., 2017; Makarov et al., 2017) use an LSTM-based sequence-to-sequence (seq2seq) models (Sutskever et al., 2014) with an additional attention mechanism (Bahdanau et al., 2015), we also experiment with such a model for our task. In other words, we treat this as a translation task of input character sequences directly to output BIO-labels, as in the CRF model, but without POS-tags in the input. After initial experiments, we set the hidden layer size at 128, the batch size as 32, the *teacher forcing* (Williams and Zipser, 1989) ratio at 0.5. Similar to the CRF-model, it jointly predicts morpheme boundaries and specific BIO-labels.

5 Results

Once the features are extracted and the training complete, the models predict morpheme segmentation and morpheme type for stems, or glosses for affixes. This section discusses the results, compared in Table 2, of all three models. The goal was for a model to complete at least 80% of the segmentation and glossing correctly, leaving the most difficult, rare, and hopefully informative forms for a human to annotate. Originally, a 90/10 split was tried but the test data was encountering a dozen or less labels. With an 80/20 split, the test encountered nearly twice as many labels and the variance of F_1-score was less between each test run. All three models performed near or above the target.

Joint segmentation and glossing/labeling produced the best results. The data is read letter by letter and each letter is associated with a BIO tag and specific morpheme type/gloss label. This identifies the letters

[1] `https://python-crfsuite.readthedocs.io/en/latest/`

[2] Using the LIBLINEAR implementation (Fan et al., 2008).

CRF	pipeline	seq2seq
0.895	0.861	0.763

Table 2: Labeled position results (F_1-score) compared across CRF-only, CRF+SVM pipeline, and seq2seq models. The first two are averages across multiple runs on random data splits.

in the morphemes as well as the morpheme boundaries. The letters were grouped into predicted morphemes for labeled position evaluation. Table 3 demonstrates the model's ability to produce reasonable results with limited training data. It appears that for *Lezgi* 3,000 words is a sufficient number of training examples.

Label	Precision	Recall	F_1	Instances
stem	0.98	0.97	0.97	127
AOR	0.93	1.00	0.97	14
FOC	1.00	1.00	1.00	10
OBL	0.75	0.67	0.71	9
GEN	0.67	0.40	0.50	5
ERG	0.67	0.40	0.50	5
DAT	1.00	1.00	1.00	4
NEG	1.00	0.75	0.86	4
PTP	0.80	1.00	0.89	4
SBST	1.00	1.00	1.00	3
IMPF	1.00	1.00	1.00	2
PERF	1.00	1.00	1.00	2
ELAT	1.00	1.00	1.00	1
SUPER	1.00	1.00	1.00	1
total/avg all	0.92	0.87	0.90	191
total/avg affixes	0.84	0.80	0.82	64

Table 3: CRF-only model labeled position results from one run over a randomized test set with 80/20 split. Averages are macroaverages.

The most acute issue is the reduction of accuracy when predicting stems compared to predicting affixes. The last line of Table 3 shows that the precision, recall, and F_1-scores of affixes have lower performance compared with the overall scores. The pipeline model results discussed in below results in a similar pattern but slightly worse results. Since training was done at character-level and affixes tend to be 1–3 letters long while stem length varies greatly, transitions between morphemes become less accurate. Also, single-letter affixes may coincide with any first or last letter of possible surrounding morphemes. The classifier is, however, adept at splitting affixes from stems, and this in itself would be helpful to human annotators. The good results on the much larger number of stems suggests that the performance on affixes will keep improving as training examples increase.

The model was provided with no information about the language's morphophonology. Its accuracy strongly correlates with the extent of isomorphism between affixes or the amount of allomorphy that a particular affix exhibits. Most affixes are unique from other morphemes and have few or no variant forms. On the other hand, the oblique affix and the ergative case morpheme are identical, but the ergative morpheme is always the last morpheme on a word while the oblique is always followed by other case morphemes. Letter position features should have caught this difference. However, the oblique and ergative case also have more allomorphs (over 10 different forms) than any other morpheme. The genitive case and the aorist tense morphemes are identical to some other morphemes, which also causes diffi-

culty. All but a handful of affixes are identified with very high accuracy. These exceptions—aorist tense (AOR) - identical to the aorist converb, genitive case (GEN) - identical to the nominalized verb marker (masdar), ergative case (ERG) and the oblique affix (OBL) which are identical to each other and highly allomorphic—indicate that limited data may not be sufficient for languages with extensive allomorphy.

When the CRF is placed in a pipeline with a SVM classifier, the CRF only identified morpheme boundaries. Overall accuracy of the pipeline was worse than the CRF-only model, achieving an average 0.86 F_1-score. This echoes the findings of Cohen and Smith (2007) and Lee et al. (2011) that joint training of syntax and morphology produce better results than separate training. The pipeline model had slightly higher accuracy on morphemes with multiple allomorphs but tended to perform worse on less frequent morphemes.

Lastly, the data was run on a bidirectional sequence-to-sequence deep neural network. The best result on the test set was over 0.76 F_1, reached at 500 epochs with early stopping.

6 Discussion and Future Work

It is crucial to test the models on other languages, especially polysynthetic languages which may not have many more morphemes per word but have more fusion and may have more complicated morphophonology. Requests were sent to field linguists working in a variety of languages, but time constraints did not allow them to achieve consistent annotation on a sufficient number of words. Yet, most features described in Section 3 are basic for all languages. It seems reasonable that extracting features specific to polysynthetic languages could produce just as high results.

The feature-based models surpassed the 80% accuracy goal using features informed by general linguistic knowledge or features that can be extracted directly from data. These features proved sufficient for *Lezgi*, though expanding to other languages might uncover other general linguistic features that would maintain high accuracy for more languages. If generic features prove insufficient, questions could be presented to linguists who provide the data and language-specific features could be extracted based on their input. The questionnaire of the LinGO Grammar Matrix (Bender et al., 2002) is a possible initial model for an interface.

The poorer results caused by the language's allomorphy do not bode well for languages with more complex series of allomorphs. An interactive interface could request human annotators for infrequent or problematic inflected forms, or such cases where the model has little confidence in the labeling. For example, noun stems harvested from FLEx's automatic lexicon builder could be presented for a *Lezgi* annotator to provide the various ergative and oblique morphemes. These single forms would augment annotated text.

Predicted morphemes and glosses need to be checked and corrected by trained annotators. Previous experiments (Baldridge and Palmer, 2009; Palmer, 2009; Palmer et al., 2010) strongly imply that vetting a portion of the data and correcting a smaller portion of machine-generated annotations is faster than manually annotating every single token. The next step is to bring the human back into the training loop by having the native speaker check and correct the model's performance on unlabeled data. The corrections would serve as additional supervised data. As more texts are annotated with the help of the model, more data could be fed into the training, increasing accuracy. In addition, although currently only affixes are glossed, the model could leverage its high success at identifying stems and present them to be glossed so that they could be added to future training data. Each iteration of prediction and correction will incrementally speed the task. In the future, it is hoped that automated support for annotation could be integrated with software such as FLEx, or another interface familiar to documentary and descriptive linguists.

7 Conclusion

We have explored a case study on *Lezgi* to examine whether machine learning techniques could break the bottleneck of documentation data production by achieving reasonable accuracy using a few interlinearized texts and general linguistic features. The results demonstrate that current NLP tools and human and data resources commonly found in documentary linguistic field projects can be combined in order

to speed annotation of valuable documentary data. A CRF classifier, a CRF+SVM pipeline, and a neural seq2seq model were tested and compared to show that machine learning could remove up to 90% of that labor from human annotators and place it upon a potential field assistant tool. Models such as these could be integrated into the workflow of language documentation and force open the annotation bottleneck. Further training should improve the accuracy of the model which, in turn, will further speed the availability of new language data. This will increase the amount of natural language data available to the language communities, linguists, and computational experiments. It achieves high accuracy with basic cross-linguistic features. A little feature engineering might transfer the high success to polysynthetic and fusional languages, or at least achieve the original Pareto tradeoff goal of 80%.[3]

References

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2014. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 569–578. Association for Computational Linguistics.

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 Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *European Language Resources Association LREC 2010: Proceedings of the 7th International Language Resources and Evaluation*, pages 890–893. European Language Resources Association.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Jason Baldridge and Alexis Palmer. 2009. How well does active learning actually work? Time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 296–305. Association for Computational Linguistics.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the 2002 workshop on Grammar engineering and evaluation—Volume 15*, pages 1–7. Association for Computational Linguistics.

Jan Buys and Jan A. Botha. 2016. Cross-lingual morphological tagging for low-resource languages. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1954–1964.

CoEDL. 2017. Early results from survey exploring transcription processes. `http://www.dynamicsoflanguage.edu.au/news-and-media/latest-headlines/article/?id=early-results-from-survey-exploring-transcription-processes`. Accessed: 2018-06-26.

Shay B. Cohen and Noah A. Smith. 2007. Joint morphological and syntactic disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 208–217. Association for Computational Linguistics.

Ryan Cotterell, Thomas Müller, Alexander M. Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-markov models. In *CoNLL-SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection in 52 Languages*, pages 164–174.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Gėraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *CoNLL*, pages 1–30. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2005. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Helsinki University of Technology Helsinki.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

[3]We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

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

Martin Haspelmath. 1993. *A grammar of Lezgian*. Walter de Gruyter, Berlin.

Nikolaus P. Himmelmann. 1998. Documentary and descriptive linguistics. *Linguistics*, 36(1):161–196.

Gary Holton, Kavon Hooshiar, and Nicholas Thieberger. 2017. Developing collection management tools to create more robust and reliable linguistic data. In *Workshop on Computational Methods for Endangered Languages*.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. Neural multi-source morphological reinflection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 514–524. Association for Computational Linguistics.

Michael Krauss. 1992. The world's languages in crisis. *Language*, 68(1):4–10.

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 Eighteenth International Conference on Machine Learning*, pages 282–289.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. 2011. Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 1–9. Association for Computational Linguistics.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528.

Peter Makarov, Tatiana Ruzsics, and Simon Clematide. 2017. Align and copy: UZH at SIGMORPHON 2017 shared task for morphological reinflection. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 49–57. Association for Computational Linguistics.

Christian Monson, Jaime Carbonell, Alon Lavie, and Lori Levin. 2007. ParaMor: Finding paradigms across morphology. In *Advances in Multilingual and Multimodal Information Retrieval*, Lecture Notes in Computer Science, pages 900–907. Springer, Berlin, Heidelberg.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (CRFs). http://www.chokkan.org/software/crfsuite/.

Alexis Palmer, Taesun Moon, Jason Baldridge, Katrin Erk, Eric Campbell, and Telma Can. 2010. Computational strategies for reducing annotation effort in language documentation. *Linguistic Issues in Language Technology*, 3(4):1–42.

Alexis Mary Palmer. 2009. *Semi-automated annotation and active learning for language documentation*. Phd thesis, University of Texas at Austin.

Lance A. Ramshaw and Mitchell P. Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Chris Rogers. 2010. Review of fieldworks language explorer (FLEx) 3.0. *Language Documentation & Conservation*, 04:78–84.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *CoNLL*, pages 29–37.

Gary F. Simons and Charles D. Fennig. 2017. Ethnologue: Languages of the world. *SIL, Dallas, Texas*.

Gary F. Simons and M. Paul Lewis. 2013. The world's languages in crisis. *Responses to language endangerment: In honor of Mickey Noonan. New directions in language documentation and language revitalization*, 3:20.

Gary F. Simons. 2013. Requirements for implementing the AARDVARC vision. Presented at workshop of the Automatically Annotated Repository of Digital Video and Audio Resources Community (AARDVARC), Eastern Michigan University, Ypsilanti, May 9-11, 2013.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112.

Ronald J. Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.