Automating Gloss Generation in Interlinear Glossed Text

Angelina McMillan-Major
University of Washington / Seattle, USA
aymm@uw.edu

Abstract

Interlinear Glossed Text (IGT) is a rich data type produced by linguists for the purposes of presenting an analysis of a language's semantic and grammatical properties. I combine linguistic knowledge and statistical machine learning to develop a system for automatically annotating low-resource language data. I train a generative system for each language using on the order of 1000 IGT. The input to the system is the morphologically segmented source language phrase and its English translation. The system outputs the predicted linguistic annotation for each morpheme of the source phrase. The final system is tested on held-out IGT sets for Abui [abz], Chintang [ctn], and Matsigenka [mcb] and achieves 71.7%, 80.3%, and 84.9% accuracy, respectively.

1 Introduction

While language documentation has a long history, warnings from linguists such as Hale et al. (1992) and Krauss (1992) concerning language extinction have revitalized and expanded documentation efforts by communities and linguists, though there is still much work to be done (Seifart et al., 2018). According to Seifart et al. (2018), it can take 40 and 100 hours to transcribe an hour of recorded material, and even more time is required to analyze the language as a whole before annotating a single segment of the data collected. Given the decreasing language diversity in the world, there is an identified and immediate need for automated systems to assist in reducing the human hours spent on the documentation process.

While costly to produce, the glosses in IGT allow linguistic generalizations that are implicitly present in natural text to be explicitly available for natural language processing. In addition to supporting field linguists in collecting data, better and more easily produced IGT would also benefit end-stage projects such as machine translation between low-resource languages by improving the accuracy of the pre-processing modules (Xia and Lewis, 2008). Georgi et al. (2012) used IGT corpora to improve dependency parsing on low-resource languages using bootstrapping methods, while Bender et al. (2014) and Zamaraeva et al. (2019) used IGT to build high-precision grammars. Furthermore, language communities with trained IGT generators would be able to produce IGT for any new text found or created to aid with either language learning, documentation, or future translation efforts.

IGT consist of a source language phrase, a translation of that phrase into the language of the target audience, such as English, and glosses for each source morpheme. The glosses highlight the morphological and syntactic features of the source language. Ex. 1 shows an IGT from the Kazakh dataset in the Online Database of INterlinear text (ODIN) (Lewis and Xia, 2010), modified from Vinnitskaya et al. (2003).

(1) Kyz bolme-ge kir-di
girl.NOM room-DAT enter-PAST
(A/the) girl entered (a/the) room. [ISO 639-3: kaz]

In Ex. 1, the first line is the source line, the second is the gloss line, and the third is the translation line. The strings *girl*, *NOM*, *room*, etc. are all glosses, but glosses that refer to grammatical information, such as *NOM*, will be referred to as *grams* and the glosses that refer to semantically contentful information, such as *girl*, will be referred to as *stems*.

In this paper I describe a system for producing the gloss line of IGT automatically. I restrict my system to producing just the gloss line, given a morphologically segmented source line and its translation line. Morphological segmentation packages such as Morfessor (Creutz and Lagus, 2007) are widely available, and in the doc-

umentation setting translations may be provided by a native speaker consultant. This system could be used in combination with such resources. The input to the system at test time includes the morphemes in the segmented source line and the translation in the bottom line, and the target output is the gloss line.

This system does not, however, produce new analyses of the source language. Rather it is assumed that the linguistic analyses at all levels and the transliteration are already formalized by the documentary team. The system is then learning patterns from the analyses in the training data and reproducing the patterns when given new data. While the system can be trained on one set of analyses and tested on another, the performance will depend on the amount of variation between the analyses. This is especially significant in the low-resource setting, where each data instance contributes a relatively large amount of information as compared to each data instance in a high-resource setting.

A survey of the literature on IGT curation, augmentation and automation is provided in §2. In §3, I present the data used for developing and testing the system. §4 describes both the machine learning methods and the rule-based methods of this particular system, where the rule-based methods provide an implementation for handling out of vocabulary, also referred to as *OOV*, tokens. This section also includes an explanation of the evaluation metrics. §5 presents the results on the development and test languages, as well as a systematic error analysis. Finally, §6 discusses the challenges and limitations inherent in casting annotation as a classification task while exploring possible improvements to the current method for predicting OOV tokens.

2 Related Work

Approaches to IGT creation tools range in terms of how much input is required from the human annotator to yield the finished product. A widely used tool for documentation is FieldWorks Language Explorer (FLEx) (Baines, 2009). FLEx includes functionality for manually annotating interlinear text in addition to creating dictionaries and other language resources. The annotation software assists the user by retaining source-gloss pairs previously entered by the user and suggesting these glosses when the source morpheme appears again. The suggestions are not automatically constrained, however, so FLEx will suggest all previously seen glosses regardless of their likelihood given the local context unless the user explicitly provides the constraint information. By contrast the system presented here calculates the likelihood of a source morpheme being labeled with each possible gloss given the current sequence of morphemes and selects the most likely gloss automatically.

Palmer et al. (2009) (see also Baldridge and Palmer 2009 and Palmer et al. 2010) approached the task of IGT glossing within an active learning framework. In an active learning framework, annotators label the first small batch of input data, which is incorporated into the model in a new training phase, and then the next batch of data is labeled by the model and corrected by the annotators before being incorporated back into the model. They trained a maximum entropy classifier to predict a gloss given a morpheme and a context window of two morphemes before and after the morpheme in question. They had two annotators label IGT for Uspanteko [usp] (Mayan, Guatemala), using data from the OKMA corpus (Pixabaj et al., 2007). This corpus contains 32 glossed and 35 unglossed texts for a total of approximately 75,000 glossed tokens. They restrict the number of labels in the annotation schema by labeling stem morphemes with their part of speech (POS) tags, as provided in the corpus. Palmer et al. found that the expert annotator was more efficient and performed better when presented with the model's most uncertain predictions, but the naive annotator annotated more accurately when presented with random IGT rather than the most uncertain. These results suggest that active learning strategies must take the annotator into account in order to be optimally efficient, whereas automatic annotation does not have this constraint. Fully automated classification approaches provide an alternative method to IGT glossing when IGT have already been completed.

Samardžić et al. (2015) took a classification approach to IGT generation for the Chintang [ctn] (Kiranti, Nepal) Language Corpus dataset (Bickel et al., 2009). This corpus is significantly larger than the average documentation project with approximately 955,000 glossed tokens and a lexicon with POS tags. Samardžić et al. used two classifiers to generate their labels. The first classifier was based on Shen et al.'s (2007) version of

Collins and Roark's (2004) Perceptron learning algorithm and jointly learns the order in which to tag the sequence and the predicted tags. It annotated grammatical morphemes with their appropriate label and contentful morphemes with their POS tags, as in Palmer et al. (2009), to limit the total number of labels. The final step replaces the POS labels with an appropriate English lemma using the provided lexicon which maps English lemmas to Chintang morphemes. Samardžić et al. trained a trigram language model on the lexicon IDs to predict the most likely ID when multiple lemmas are possible, and back-off methods are used when labeling a previously unseen morpheme.

This paper attempts to add to the body of research on IGT generation by developing a machine learning framework that can apply to languages with fewer resources. Whereas these previous implementations rely on linguists' input or language specific resources, such as source language POS tags, to produce the final output, the system presented here runs using only what is given in the IGT training data. The following experiments attempt to answer the question of how much linguistic information statistical machine learning techniques are able to acquire from the linguistic patterns that are made explicit in IGT without any additional resources.

3 Data

The Online Database of INterlinear text (ODIN) is a repository of IGT examples collected from PDFs of linguistic publications (Lewis and Xia, 2010). ODIN contains 158,007 IGT from across 1,496 languages and 2,027 documents. The ODIN IGT datasets are stored in the XML-linearization of the Xigt format (Goodman et al., 2015), which includes a Python API.[1] A second version of ODIN[2] has been released with POS tags, dependency parses, and word alignments provided by the INterlinear Text ENrichment Toolkit (INTENT) system (Georgi, 2016).

I selected six languages from ODIN for developing the system based on set size: Turkish [tur], Russian [rus], Korean [kor], Japanese [jpn], Italian [ita], and Norwegian [nob]. I use a further three languages from language documentation projects as held-out test languages. Poor results on held-out languages compared to development languages would suggest that the system is inherently biased towards one language or one typological feature, such as word order; comparable results between the held-out and development languages provide evidence that the system performance is not dependent on language-specific features. The datasets for Chintang [ctn] (Kiranti, Nepal; Bickel et al. 2009), Abui [abz] (Trans-New Guinea, Indonesia; Kratochvíl 2017), and Matsigenka [mcb] (Maipurean, Peru; Michael et al. 2013) have been collected as part of language documentation projects and thus provide the opportunity to model system behavior in that setting. This setting typically includes consistent glossing schemes and native speaker consultants to provide translation information. In order for the system to produce models for these datasets in the same way as the ODIN datasets, preprocessing included converting the resources to the Xigt format and then enriching the data using the INTENT system (Georgi, 2016).

After filtering for IGT with identical source lines and IGT that were not fully annotated by INTENT, the Japanese and Korean sets have slightly more than 2000 IGT each, the Russian has set just under 1500 IGT, the Norwegian and Turkish sets have around 1000 IGT each, and the Italian set has around 800 IGT. Of the held-out datasets, Matsigenka is the smallest, with just under 450 IGT due to a large portion of the corpus having Spanish rather than English translations. The Abui and Chintang sets are much larger with approximately 4700 IGT and 7000 IGT.[3] For each language the system is trained using 90% of the given language's IGT and tested on the remaining 10%. Table 1 shows the number of IGT in each language's train and test sets from ODIN, while Table 2 shows the numbers for the held-out languages.

4 Methodology

I built one glossing system trained separately on each language dataset. Upon loading each dataset, the system removes IGT with source lines that appear multiple times in the dataset and IGT with missing or incomplete label references to the glosses and source morphemes. The system then

[1] http://github.com/xigt/xigt

[2] Available at http://depts.washington.edu/uwcl/odin/

[3] This is a subsample of the nearly 1 million word Chintang dataset (see §2).

formats the information in the remaining IGT to be fed into two Conditional Random Field (CRF) models (Lafferty et al., 2001). One model predicts the gloss line from the source line, hereafter referred to as the *source model* or *SRC model*, while the second model predicts the gloss line from the translation line, hereafter *translation model* or *TRS model*. Finally, the system incorporates the predictions of both models into the final output.

I use the Japanese example in (2), originally from Harley (1995), as a running example to show the steps in the system.

(2) yakko-ga wakko-o butai-ni agar-ase-ta
yakko-n wakko-a stage-on rise-cause-past
yakko made wakko rise onto the stage [jpn]

The source line, gold glosses, and the translation line are as they appear in the corpus.

4.1 Modeling

Conditional Random Fields (CRF) are able to classify sequences of tokens with a large number of possible labels while being sensitive to the context in which the tokens appear (Lafferty et al., 2001) and have been shown to be effective in low-resource settings (Ruokolainen et al., 2013). The CRF models were built using sklearn-crfsuite v0.3.6.[4] The training algorithm uses stochastic gradient descent with L2 regularization and a maximum of 50 iterations.

The SRC model predicts a gloss for each morpheme in the source line. When training, the system takes in complete IGT and uses the glosses provided as the gold training labels. The first whitespace-separated token in the source line is assumed to align with the first whitespace-separated token in the gloss line, the second source token with the second gloss, and so forth. While the SRC model is able to take advantage of the context provided by adjacent morphemes, it must also be provided with explicit features for source word boundaries. The features for each label include the source morpheme, the current source word, the previous and following words, and whether or not the previous and following morphemes are included in the current word (see Appendix A for an example). No processing of the source language, such as POS tags or dependency labels, other than the morphological segmentation has been assumed in this model, as many languages do not have access to NLP processing during the documentation process. At test time the SRC model would then output the following predicted sequence for the source line in Ex. 2:

(3) yakko-n pizza-acc taro-dat sit-cause-past

The second model, or TRS model, predicts the gloss that is aligned with each word in the translation line. The gold labels for the translation to gloss line predictions are provided by INTENT, which has automatically labeled the bilingual alignments between one gloss and one translation word. As a result, multi-word expressions are not considered in the TRS model unless they are explicit in the glosses. Many of the words in the translation line are not aligned with a gloss, so an additional null label is included. The features for each label include the translation word, its lemma as provided by the StanfordNLP API (Manning et al., 2014), and the POS tag and dependency structure for the translation word as provided by INTENT (again, see Appendix A for an example). The TRS model then outputs the following predicted sequence for the translation line in Ex. 2:

(4) yakko NA NA NA NA NA NA

NA stands for *Not Aligned* and is the most likely tag for the model to output. The content words that would be expected to be aligned in the translation line, *wakko*, *rise*, and *stage*, are not aligned in this case due to *wakko* and *rise* being OOV items, and *stage* having only been seen once in the training data. For further discussion of the TRS model's behavior, see §6. For both models, tokens that contain only punctuation are labeled with the gloss *PUNC*. Additionally, a dummy label is included in case of reference errors while accessing the data or when the features are not available. This may be the case with punctuation or with non-English words that the StanfordNLP lemmatizer is not able to process.

4.2 Integrating Model Hypotheses

At test time the given source line and its translation line are processed by their respective models. The output of each model is then assessed by the system. The system first checks whether the source tokens and their predicted glosses have co-occurred in the training data and whether the translation tokens and their predicted glosses have co-occurred in the training data. If a gloss is predicted

[4] http://github.com/TeamHG-Memex/sklearn-crfsuite

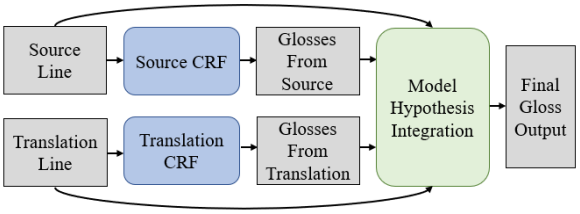

Figure 1: Visualization of the system

by both models and is supported by the training data, it's saved as a final prediction. If the SRC and TRS models disagree and the TRS model's prediction is supported by the training data, the TRS model's prediction is saved as the final prediction. If the original source token has been seen in the training data, but an exact match was not predicted by the translation line, the SRC model's prediction takes precedence. This is motivated by the fact that source tokens that are labeled with grams may not be aligned with a token in the translation.

If the source morpheme has not previously been seen, it is assumed to be a stem, and the lemma of an aligned translation lemma is used as the gloss (see § 6 for further discussion). If the source token is both unseen and unaligned, the system first checks to seen if there is an exact match between the morpheme and a translation word. Otherwise, the system separates the predicted grams, as identified by the gram list, from the SRC model's predicted gloss. Based on the grams, the system attempts to match the morpheme with a translation lemma with the same POS tag or argument role, using the grams to predict the morpheme's POS tag and the INTENT metadata to identify the translation words' POS tags or dependency structure. For example, if a case marker such as nominative is predicted, the system will look for a noun marked as the subject in the translation tokens. This process is implemented for nouns and verbs since OOV items are most likely to be in those categories. Finally, if the model is still unsure of the final prediction, the system selects the lemma of an unaligned translation word or the word itself if it cannot be lemmatized.

Continuing with the example from the previous section, the system now has the prediction information from Ex. 3 and 4. The system confirms that it has seen *yakko*, *ga*, *o*, *ni*, *ase*, and *ta* glossed as *yakko*, *n*, *acc*, *dat*, *cause*, and *past*, respectively, so it keeps those as final predictions. The system has seen *butai* in the training data but not glossed as *taro*, so it replaces the SRC model's prediction with the previously seen gloss, *stage*. The token *wakko* is an OOV item, but an exact match is found in the translation line, so the token itself is used as the gloss, replacing *pizza*. The token *agar* is also an OOV item, but because no grams were predicted by the TRS model, the system does not make any assumptions about the source POS tag and defaults to the token predicted by the SRC model. The resulting final prediction is:

(5) yakko-n wakko-acc stage-dat sit-cause-past

4.3 Evaluation

The system's performance is evaluated by comparing each gloss in each test IGT's final output to the gold standard glosses provided in the datasets. The system produces a label for each morpheme, so the recall provides no additional information. Comparing the final output in Ex. 5 with the gold gloss in Ex. 2, *yakko*, *n*, *wakko*, *stage*, *cause*, and *past* are correct for a total of 6/9. The system precision is given in terms of the micro-average over all tokens in all the IGT in each languages' test dataset.

I further analyze the system output by breaking down the system performance in terms of stems and grams. Labels are identified as grams or stems during the scoring process using a list of grams collected during the development of ODIN. The ODIN gram list covers many frequently used categories such as person, gender and case and has multiple realizations for each category's values.

There may be morpheme labels that contain multiple glosses, each separated by a period. In these cases, the predicted label is evaluated as a whole when scoring the system accuracy. When determining the system performance over stems and grams, however, the predicted label is split on each period and each gloss is checked against the ODIN gram list to determine if it is a gram or not. The gold label is also split if it contains at least one period. For each gloss in the gold label, if it is seen in the predicted label, it is considered correct, regardless of the order. Because the system may predict a label that has more or fewer glosses than the gold label, both the precision and recall are calculated. Each metric is presented in terms of the micro-average over all the stem tokens and the micro-average over all the gram tokens.

Ex. 2 does not contain any instances of a single label containing multiple glosses, so the combined

Lang. [ISO 639-3]	Train	Test	Acc
Japanese [jpn]	2062	229	77.8%
Korean [kor]	1956	217	75.6%
Norwegian [nob]	958	107	63.1%
Turkish [tur]	894	99	60.3%
Italian [ita]	732	81	59.9%
Russian [rus]	1322	147	53.2%

Table 1: Development languages, number of IGT training and test instances for each model, and test accuracy.

score for the stems and morphemes is not different from the morpheme score. In a more complicated example from the Japanese dataset originally from Bobaljik (n.d.), there are two instances of multi-gloss labels, *last.night* and *by.dat*.

(6) yuube kuruma-ga doroboo-ni nusum-are-ta
last.night car-nom robber-by.dat steal-pass-past
Last night, cars were stolen by a thief. [jpn]

The SRC model predicts the sequence *japanese car-nom thief-by steal-pass-past*. The TRS model predicts that *last*, *night* and *thief* are glosses. The rest of the words are not predicted to be aligned, and the final output is determined to be *last car-nom thief-by steal-pass-past*. In this output, the predicted label for *yuube* is missing a stem, *night*, *thief* is predicted instead of *robber*, and the predicted label for *ni* is missing a gram, *dat*. The morpheme score is 5/8, but the stem precision is 3/4, the gram precision is 4/4, the stem recall is 3/5, and the gram recall is 4/5.

5 Results

The results of all the development languages vary greatly, ranging from 53.2% to 77.8% accuracy.[5] There is a noticeable trend in which the relative model accuracy is predictable from the number of test IGT, with the exception of the Russian dataset. Table 1 shows the number of test IGT, training IGT, and system accuracy per development language. Table 2 shows the same information for the held-out languages, with an increasing number of training IGT over the same test set for Abui and Chintang. In addition to training on the full training sets, I also train the system on the initial 25%, 50%, and 75% of the training data for Abui and Chintang to see the effect of training set size on the system accuracy and train again on a random

[5]Code and instructions for reproducing these results are available at https://github.com/mcmillanmajora/IGTautoglossing.

Lang. [ISO 639-3]	Train	Test	Acc
Matsigenka [mcb]	388	43	**84.9%**
Chintang [ctn]	6589	677	**80.3%**
initial 75%	4941	677	**74.6%**
random 75%	4941	677	74.3%
initial 50%	3294	677	**72.6%**
random 50%	3294	677	72.5%
initial 25%	1646	677	68.7%
random 25%	1646	677	**69.0%**
Abui [abz]	4295	447	**71.7%**
initial 75%	3224	447	69.9%
random 75%	3224	447	**70.4%**
initial 50%	2149	447	68.7%
random 50%	2149	447	**69.1%**
initial 25%	1076	447	**66.1%**
random 25%	1076	447	64.9%

Table 2: Held-out languages, number of training and test IGT, and test accuracy. Training instances were selected randomly if *random* or from the beginning of the dataset if *initial*. Test IGT were held constant.

25%, 50%, and 75% of the training data to see the effect of vocabulary overlap. These datasets include IGT from different documentation sessions, so the assumption is that consecutive IGT are more likely to have been created at the same time and therefore contain repeated words. These sets are all tested using the same IGT in the test set for the full training data experiment.

5.1 Development Languages

Among the development languages, the system had the highest accuracies with the Korean and Japanese datasets at 75.6% and 77.8%. The Japanese training set had just over 2000 IGT and the Korean training set had just under 2000 IGT. Both sets had slightly more than 200 test IGT. The system performed less well over the Italian, Turkish and Norwegian datasets at 59.9%, 60.3%, and 63.1%, respectively. These datasets had less than half the data of the Japanese and Korean datasets. The system performed worst over the Russian dataset, at 53.2% accuracy on 1322 training instances, almost a third more than the Norwegian dataset.

A clearer pattern in the system's performance over the development languages arises when the labels are broken down into stems and grams, as seen in Table 3. For stems, precision scores range between 60.9% and 73.3% and recall scores range between 59.9% and 71.6%, whereas the precision

Lang.	Prec.		Rec.	
	Stem	Gram	Stem	Gram
jpn	73.3%	88.2%	71.6%	85.4%
kor	72.1%	83.0%	70.5%	80.5%
nob	63.8%	73.5%	62.7%	65.8%
tur	61.7%	63.8%	61.1%	56.1%
ita	63.6%	60.6%	62.6%	48.8%
rus	60.9%	67.4%	59.9%	49.2%

Table 3: Analysis of system performance on development languages with precision and recall for stems and grams.

scores for grams range between 60.6% and 88.2% and the recall scores range between 48.8% and 85.4%. Japanese, Korean, and Norwegian all have higher scores for grams than stems in both precision and recall. That trend reverses for Turkish, Italian, and Russian, where the recall for grams is lower than stems. Japanese, Korean, and Turkish have much lower ratios of stems to grams, each having about 3 stem morphemes for every 2 grams. Russian, Italian, and Norwegian have about 5, 7, and 10 stems, respectively, for every 2 grams. Norwegian's high ratio is likely due to the syntactic similarity between it and English, which makes glossing with inflected English words easier. Because grams are often not annotated as separate morphemes, poor recall on grams would contribute to over lower scores on morpheme accuracy even if the stem is correctly predicted because the evaluation considers the predicted label as a whole. This is particularly true in the ODIN data, which also suffers from errors introduced when extracting IGT from linguistic papers and from what Lewis and Xia (2008) call *IGT bias*. IGT are most likely presented for a specific phenomena that is unique within the language and is overly represented in the paper compared to broader contexts. As a result, the set of IGT pulled from a single paper are likely skewed and the glossing may reflect the focus on a particular portion of the sentence, if a full sentence is given.

5.2 Held-out Languages

The system achieved higher accuracies over the Matsigenka and Chintang datasets than the development sets and comparable accuracies for the Abui dataset. The system achieved a higher accuracy for Matsigenka, 84.9%, than it did for any of the development datasets, which all had at least twice as much training data. The system was also trained for randomized and initial subsets of the training data for Abui and Chintang, resulting in 7 total experiments for each language. Table 2 shows the results on the various splits. The Abui results range from 66.1% to 71.7% on 447 test IGT, and the Chintang results range from 69% to 80.3% on 677 test IGT.

Lang.	Prec.		Rec.	
	Stem	Gram	Stem	Gram
mcb	**73.5%**	**96.0%**	**70.3%**	**95.8%**
ctn	**71.2%**	**92.5%**	**69.9%**	**92.9%**
init. 75%	**60.7%**	**92.2%**	**60.4%**	**92.9%**
rand. 75%	60.5%	92.0%	59.9%	92.7%
init. 50%	57.2%	91.1%	56.2%	**92.2%**
rand. 50%	**57.3%**	91.1%	56.2%	92.1%
init. 25%	51.0%	88.7%	51.0%	91.1%
rand. 25%	**51.1%**	**89.0%**	**51.3%**	**91.3%**
abz	**70.3%**	**83.4%**	**72.5%**	**85.8%**
init. 75%	68.4%	81.9%	70.6%	84.5%
rand. 75%	**69.0%**	**82.7%**	**71.1%**	**85.1%**
init. 50%	66.9%	81.4%	68.8%	83.4%
rand. 50%	**67.8%**	**82.1%**	**69.5%**	**84.5%**
init. 25%	**63.4%**	79.6%	**65.6%**	**82.9%**
rand. 25%	63.0%	**79.9%**	65.0%	81.4%

Table 4: Analysis of system performance on held-out languages with precion and recall for stems and grams.

The held-out languages do pattern with the well-performing development datasets in terms of higher precision and recall for grams than stems. Table 4 shows the gram precision ranging from 79.6% to 96.0% and the gram recall ranging from 81.4% to 95.8% over all of the datasets. The stem scores have greater ranges, from 51% to 73.5% for precision and 51% to 72.5% for recall. The Chintang and Abui subsets do not differ more than 2% accuracy between the randomized and the non-randomized training set pairs. The Chintang stem precision and recall increase the most between the 75% and full sets, but the Abui stems see the biggest increase between the 25% and 50% subsets.

Samardžić et al. (2015) achieve 96% accuracy on 200,000 test word tokens in the Chintang dataset using approximately 800,000 word tokens for training. My system is maximally tested on 7250 Chintang morphemes using only 55,000 training morphemes and achieves 80.3% accuracy. My system also does not assume any language-specific metadata, while Samardžić et al. make

use of a Chintang lexicon containing high-quality source POS tags. They also provide an analysis of their system's performance over lexical labels (stems) and functional labels (grams). In general, their model's performance over grams increases with the training set size, while the performance over stems remains fairly constant. Samardžić et al. attribute this pattern to the sequential inclusion of IGT collected from source texts that differ lexically or stylistically as well as differing annotation schema over these sources.

6 Error Analysis

In investigating the predictions made by the models and the final output glosses, a number of inconsistencies in the ODIN datasets became apparent. Processing errors occur when there are a mismatched number of source morphemes and gloss labels, such as when a multi-word expression is used as a single gloss and contains whitespace or when a coindexation variable is included in the source line as a separate token. Some instances also include additional punctuation indicating clausal boundaries. Authors of linguistic papers use IGT to illustrate syntactic and semantic properties of languages and these additional annotations are often included to highlight the relevant information for the audience.

Due to the wide range of authors from which the ODIN IGT originate, many grams may refer to the same grammatical concept, as shown in Ex. 2 and 6 from the Japanese dataset. The morpheme *ga* indicates the nominative case, but is labeled as *n* in Ex. 2 and *nom* in Ex. 6. The system treats these labels as unique though they are intended to be synonymous. In contrast to the unintended ambiguity, Ex. 7 and Ex. 9 below both contain the Chintang morpheme *lo*, but in Ex. 6 it is labeled as *okay* and in Ex. 8 it is labeled *surp* as in the morpheme indicates the speaker's surprise.

(7) lo sat na maha na
okay seven top not top
okay, not seven [ctn] (Bickel et al., 2009)

Furthermore, *lo* can also appear as a nominative suffix for the interrogative pronoun *sa*, meaning *who* (Paudyal, 2015). While these functions are difficult for the system to differentiate, it can learn the contexts for each function given enough examples and consistent annotation. Multiple labels for the same function, however, will cause the system to try to discriminate between instances of the same context, as in the case of the truly ambiguous morphemes. Furthermore, the high accuracy over the test languages suggests that the consistency of the annotations has a stronger effect on the system performance than dataset size.

The system also contributes a number of consistent errors. For example, in this IGT from the Korean dataset the system relies too heavily on the source line, ignoring the correct TRS model predictions.

(8) emeni-ka us-usi-ess-up-nita
mother-nom smile-sh-pst-pol-dec
mother smiled [kor] (Yang, 1994)

The SRC model predicts the sequence *mother-nom miss-hon-pst-pol-dec* and the TRS model predicts that *mother* and *smile* are glosses, however the system keeps the incorrect gloss *miss* from the SRC model because *us* and *miss* co-occurred in the training data. This suggests that overall system performance might improve if the source predictions were preferred for grams and the translation predictions for stems.

However, across all of the languages, the TRS model frequently predicts only the null label, as seen in Ex. 4. The training data alignments sometimes do not include alignments between grams and English function words, so a significant portion of the information in the translation line is not incorporated into the model. Including a preprocessing step to supplement the INTENT alignments by aligning English function words with likely glosses, such as *was* and *past*, may improve the TRS model accuracy by decreasing the likelihood of the null label.

Further improvements could also be made in the selection and lemmatization of OOV replacements from the translation. The system often fails to find the correct stem, and even when it does find the stem, it may not be a direct match with gold gloss.

(9) yo-ni terso lo nang
dem.across-dir straight surp but
there straightly [ctn] (Bickel et al., 2009)

In predicting the glosses for the source line in Ex. 9, the SRC model outputs the sequence *dem.across-dir really surp but*. The system identifies *terso* and *straightly* as OOV items, but fails to lemmatize *straightly* to *straight*.

This example also shows that the stem and gram scores for the held-out languages are not entirely accurate, as the non-ODIN annotations contain grams like *surp* not covered by the ODIN gram

list. While this doesn't affect the the overall morpheme score, it may indicate that the patterns seen in the held-out data stem and gram scores don't reflect the system's true performance as reliably as the patterns over the development data. Allowing for project-specific gram lists may improve and provide more confidence in gram and stem scores.

The differing annotation schemata also make it difficult to draw cross-linguistic conclusions as each annotation schema is founded in a different set of theoretical assumptions. These experiments, however, do show some of the challenges that machine learning techniques have with language as a data type as opposed to other sequential data. Because of the learning algorithm's reliance on the surrounding context of each label to make predictions, the linguistic properties that introduce more possible answers to a morpheme's label due to ambiguous contexts make the predictions more difficult. For example, non-concatenative morphology, highly polysemous source morphemes, and irregularities in word order will all compound to make the information that the algorithm is able to learn from the training data more sparse. All languages contain these complexities to some degree, but the amount that is present in the training data will have a large effect on the system performance.

7 Future Work

Over all the languages, the system performance would improve by modifying how the system balances the information from the SRC and TRS models. Providing confidence scores for each predicted gloss and reducing the influence of the SRC model are immediate steps toward better accuracies. A pretrained TRS model over multiple language datasets may also minimize the number of OOV items in the model, thereby increasing the confidence of non-null glosses. Georgi (2016) saw a boost in the precision of alignments between the gloss line and the translation line using this technique with a statistical aligner, though the heuristic approach ultimately had a better F1 score due to higher recall. Georgi proposed that this was due to the variable word order of the gloss line when combining data from across languages, which suggests that the classification approach may be more robust to this variation as the model is learning the mapping from the translation word to the gloss rather than the alignment itself.

While the current implementation focuses on English translations, the submodules for POS tagging and dependency parsing could be modified to support documentation efforts using other high-resource languages. Further modification of the feature input system would allow users to make use of any additional resources available to their project. Confidence scores on all output labels would also help the end user in quickly identifying possible OOV or ambiguous tokens.[6] Once the model performance has been optimized over the available datasets, the true test of the system would be to monitor usability and its effect on the number of human hours required in an ongoing documentation project, as in Palmer et al. (2009).

8 Conclusion

This work outlines an initial supervised system for automatically annotating IGT given a morpheme-segmented source phrase and its translation. The system uses CRFs to predict the glosses from the source and translation lines individually and combines the information in a heuristic fashion to form a final prediction. The system was developed on six languages from ODIN, and tested on held-out languages. The held-out language datasets were provided by linguists and native speaker collaborators, modeling the intended use case of a documentation project. An intrinsic evaluation shows that system performs better on the held-out language datasets than the development data from ODIN, but the error analysis suggests that this is due to differences in annotation practices. Further work is needed to improve the system's final prediction selection, particularly with regards to OOV items.

Acknowledgments

Thank you to Emily Bender, Fei Xia, Michael Goodman, Ryan Georgi, David Inman, Olga Zamaraeva, Kristen Howell, and the anonymous reviewers for their comments and contributions to this work.

References

David Baines. 2009. Fieldworks Language Explorer (FLEx). *eLEX2009*, page 27.

Jason Baldridge and Alexis Palmer. 2009. How well does active learning *actually* work? Time-based

[6] Thank you to an anonymous reviewer for this suggestion.

evaluation of cost-reduction strategies for language documentation. In *Proceedings of EMNLP 2009*.

Emily M Bender, Joshua Crowgey, Michael Wayne Goodman, and Fei Xia. 2014. Learning grammar specifications from IGT: A case study of Chintang. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 43–53.

Balthasar Bickel, Goma Banjade, Toya N Bhatta, Martin Gaenzle, Netra P Paudyal, Manoj Rai, Novel Kishore Rai, Ichchha Purna Rai, and Sabine Stoll. 2009. Audiovisual corpus of the Chintang language, including a longitudinal corpus of language acquisition by six children, plus a trilingual dictionary, paradigm sets, grammar sketches, ethnographic descriptions, and photographs. DoBeS, Universität Leipzig, Nijmegen, Leipzig.

Jonathan David Bobaljik. n.d. 321 Syntax I Lecture Notes: Class 4: NP-Movement.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34.

Ryan Georgi. 2016. *From Aari to Zulu : massively multilingual creation of language tools using interlinear glossed text*. Ph.D. thesis, University of Washington, Seattle, WA, USA.

Ryan Georgi, Fei Xia, and William Lewis. 2012. Improving dependency parsing with interlinear glossed text and syntactic projection. In *Proceedings of COLING 2012: Posters*, pages 371–380. The COLING 2012 Organizing Committee.

Michael Wayne Goodman, Joshua Crowgey, Fei Xia, and Emily M. Bender. 2015. Xigt: extensible interlinear glossed text for natural language processing. *Language Resources and Evaluation*, 49(2):455–485.

Ken Hale, Michael Krauss, Lucille J. Watahomigie, Akira Y. Yamamoto, Colette Craig, LaVerne Masayesva Jeanne, and Nora C. England. 1992. Endangered languages. *Language*, 68(1):1–42.

Heidi Britton Harley. 1995. *Subjects, events, and licensing*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Frantisek Kratochvíl. 2017. Abui corpus. electronic database: 162,000 words of natural speech, and 37,500 words of elicited material.

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

John D. 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*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

William D Lewis and Fei Xia. 2008. Automatically identifying computationally relevant typological features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, pages 685–690, Hyderabad, India.

William D. Lewis and Fei Xia. 2010. Developing ODIN: A multilingual repository of annotated language data for hundreds of the world's languages. *Literary and Linguistic Computing*, 25(3):303–319.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Lev Michael, Christine Beier, Zachary O'Hagan, Harold Vargas Pereira, and Jose Vargas Pereira. 2013. Matsigenka text written by Matsigenka authors.

Alexis Palmer, Taesun Moon, and Jason Baldridge. 2009. Evaluating automation strategies in language documentation. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 36–44, Boulder, Colorado. Association for Computational Linguistics.

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.

Netra P Paudyal. 2015. *Aspects of Chintang syntax*. Ph.D. thesis, University of Zurich, Zurich, Switzerland.

Telma Can Pixabaj, Miguel Angel Vicente Méndez, Maria Vicente Méndez, and Oswaldo Ajcot Damián. 2007. Text collections in Four Mayan Languages. *Archived in The Archive of the Indigenous Languages of Latin America*.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29–37, Sofia, Bulgaria. Association for Computational Linguistics.

Tanja Samardžić, 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)*, pages 68–72.

Frank Seifart, Nicholas Evans, Harald Hammarstrm, and Stephen C. Levinson. 2018. Language documentation twenty-five years on. *Language*, 94(4):E324–E345.

Libin Shen, Giorgio Satta, and Aravind Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 760–767, Prague, Czech Republic. Association for Computational Linguistics.

Inna Vinnitskaya, Suzanne Flynn, and Claire Foley. 2003. The acquisition of relative clauses in a third language: comparing adults and children. In *Proceedings of the 6th Generative Approaches to Second Language Acquisition Conference*, pages 340–345.

Fei Xia and William D. Lewis. 2008. Repurposing theoretical linguistic data for tool development and search. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

Byong-Seon Yang. 1994. *Morphosyntactic phenomena of Korean in role and reference grammar: psych-verb constructions, inflectional verb morphemes, complex sentences, and relative clauses*. Ph.D. thesis, State University of New York at Buffalo, Buffalo, NY, USA.

Olga Zamaraeva, Kristen Howell, and Emily M Bender. 2019. Handling cross-cutting properties in automatic inference of lexical classes: A case study of Chintang. In *Proceedings of the Workshop on Computational Methods for Endangered Languages*, volume 1, page 5.

A Model Features

Using Ex. 2 as an illustration of the tagging process at test time, the system takes the source line as input then formats it to be fed into the SRC model. The representations for the first three morphemes can be seen in Table 5, where i is the current position in the sequence, m_i is the current morpheme, w_i is the current word, w_{i-1} is the previous word, m_{i+1} in w_i is the following morpheme if it occurs within the same word as m_i, and so on. The value *BOS* refers to the beginning of the sentence, and the value for the w_{i+1} feature for phrase-final morphemes is *EOS*, which refers to the end of the sentence.

feat. name	$i = m_1$	$i = m_2$	$i = m_3$...
m_i	yakko	ga	wakko
w_i	yakko-ga	yakko-ga	wakko-o
w_{i-1}	BOS	BOS	yakko-ga
w_{i+1}	wakko-o	wakko-o	butai-ni
m_{i-1} in w_i	NONE	yakko	NONE
m_{i+1} in w_i	ga	NONE	o

Table 5: Feature representation of the source line.

Again using Ex. 2, the TRS model would take the translation line as input and format it to be fed into the model. The representations for the first three words can be seen in Table 6, where i is the current position in the sequence, tw_i is the current translation word, ds_i is the dependency structure tag of the current word as given by the INTENT system, ps_i is the POS tag as given by INTENT, and lem_i is the lemma of the word as given by the StanfordNLP lemmatizer.

feat. name	$i = tw_1$	$i = tw_2$	$i = tw_3$...
tw_i	yakko	made	wakko
ds_i	nsubj	root	dobj
ps_i	nnp	vbd	nnp
lem_i	yakko	make	wakko

Table 6: Feature representation of the translation line.