Automating Gloss Generation in Interlinear Glossed Text

Anonymous NAACL submission

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 Zamarava 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
(A/the) girl entered (a/the) room. [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 are widely avail-
able (Creutz and Lagus, 2007), and in the document-
umentation setting translations may be provided
by a native speaker consultant, so there are cases
where this scenario is realistic. 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.

A survey of the literature on IGT curation, aug-
mentation and automation is provided in §2. In §3,
I present the data used for developing and testing
the system. §4 describes both the machine learn-
ing methods and the rule-based methods of this
particular system. 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 handling out
of vocabulary, also referred to as 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 Lan-
guage Explorer (FLEX) (Baines, 2009). FLEX in-
cludes functionality for manually annotating inter-
linear text in addition to creating dictionaries and
other language resources. The annotation software
assists the user by retaining source-gloss pairs pre-
viously 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 lo-
cal 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 se-
lects 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 with an active learning
framework. They train a maximum entropy classi-
ifier for morpheme annotation by presenting a hu-
man annotator with a small number of IGT to gloss
and then incorporating those IGT into the current
machine learning model. This classifier predicts
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 bet-
ter when presented with the model’s most uncer-
tain predictions, but the naive annotator annotated
more accurately when presented with random IGT
rather than the most uncertain. These results sug-
gest 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 pro-
vide 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 Language Cor-
pus dataset (Bickel et al., 2009). This corpus is
significantly larger than the average documenta-
tion project with approximately 955,000 glossed
tokens and a lexicon with POS tags. Samardžić
et al. used two classifiers to generate their la-
bes. The first classifier is based on Shen et al.
(2007)’s version of Collins and Roark (2004)’s
Perceptron learning algorithm jointly learns the
order in which to tag the sequence and the pre-
dicted tags. It annotates grams with their appro-
 priate label and stems with their POS tags, as in
Palmer et al., to limit the total number of labels.
The final step replaces the POS labels with an ap-
propriate English lemma using the provided lexi-
on which maps English lemmas to Chintang mor-
phemes. Samardžić et al. train 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 re-
search on IGT generation by developing a machine
learning framework that can apply to languages
with fewer resources. Whereas these previous im-
plementations rely on linguists’ input or language
specific resources, such as source language POS
tags, to produce the final output, the system pre-

sented 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). Currently, 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. A second version of ODIN 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 Mat-sigenka [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, the jpn and kor sets have slightly more than 2000 IGT each, the rus has set just under 1500 IGT, the nob and tur sets have around 1000 IGT each, and the ita set has around 800 IGT. Of the held-out datasets, mcb is the smallest, with just under 450 IGT due to a large portion of the corpus having Spanish rather than English translations. The abz and ctn sets are much larger with approximately 4700 IGT and 7000 IGT. 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 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 provide a running example from the jpn dataset, originally from Harley (1995), 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. This is a subsample of the nearly 1 million word Chintang dataset (see §2).
v0.3.6. 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 A for an example). No processing, 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. The SRC model then outputs the following predicted sequence:

(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 outputs the bilingual alignments 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 A for an example). The TRS model then outputs the following predicted sequence:

(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 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 see 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 pre-
dicted 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 ‘nom- inative’ 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 to 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 SRC 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 macro-average over all gloss types in each language’s 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 macro-average over all the stems and the macro-average over all the grams.

Ex. 2 does not contain any instances of a single label containing multiple glosses, so the combined score for the stems and morphemes is not different from the morpheme score. In a more complicated example from the jpn dataset originally from Bobaljik (n.d.), there are two instances of multigloss labels, last.night and by.dat.

(6) yuube kuruma-ga dorobo-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 77.8% to 53.2% accuracy. There is almost a direct correlation between the number of test IGT and the model accuracy, with the exception of the rus 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,
Table 1: Development languages, number of IGT training and test instances for each model, and test accuracy.

| 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%|

The system achieved a higher accuracy for mcb, and comparable accuracies for the abz dataset. The system achieved higher accuracies over the development languages with precision and recall for stems and grams.

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.

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

| Lang. | Prec. | Stem | Gram  | Rec. | 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%|       |

The system performed well over the jpn and kor datasets. The system performed worst over the rus dataset, at 53.2% accuracy on 1322 training instances, almost a third more than the nob 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 73.3% and 60.9% and recall scores range between 71.6% and 59.9%, whereas the precision scores for grams range between 88.2% and 60.6% and the recall scores range between 85.4% and 48.8%. Jpn, kor, and nob all have higher scores for grams than stems in both precision and recall. That trend reverses for tur, ita, and rus, where the recall for grams is lower than stems. Jpn, kor, and tur have much lower ratios of stems to grams, each having about 3 stem morphemes for every 2 grams. Rus, ita, and nob have about 5, 7, and 10 stems, respectively, for every 2 grams. Nob’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 lower scores on morpheme accuracy even if the stem is correctly predicted because the evaluation considers the predicted label as a whole.

5.2 Held-out Languages

The system achieved higher accuracies over the mcb and ctn datasets than the development sets and comparable accuracies for the abz dataset. The system performed less well over the nob, tur and ita datasets at 63.1%, 60.3%, and 59.9%, respectively. These datasets had less than half the data of the jpn and kor datasets. The system performed worst over the rus dataset, at 53.2% accuracy on 1322 training instances, almost a third more than the nob dataset.

With multiple train and test splits for abz and ctn. 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 abz and ctn to see the effect of training set size on the system accuracy and train again on a random 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

The system had the highest accuracies with the jpn and kor datasets at 77.8% and 75.6%. The jpn training set had just over 2000 IGT. Both sets had slightly more than 200 test IGT. The system performed less well over the nob, tur and ita datasets at 63.1%, 60.3%, and 59.9%, respectively. These datasets had less than half the data of the jpn and kor datasets. The system performed worst over the rus dataset, at 53.2% accuracy on 1322 training instances, almost a third more than the nob dataset.

Training instances were selected randomly if random or from the beginning of the dataset if initial. Test IGT were held constant.

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

| Lang. | Prec. Stem | Gram | Rec. 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%|

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 abz and ctn, resulting in 7 total experiments for each language. Table 2 shows the results on the various splits. The abz results range from 66.1% to 71.7% on 447 test IGT, and the ctn results range from 69% to 80.3% on 677 test IGT.

The held-out languages do pattern with the well-performing development datasets in terms of higher precision and recall for stems than grams. Table 4 shows the gram precision ranging from 96.0% to 79.6% and the gram recall ranging from 95.8% to 81.4% over all of the datasets. The stem scores have greater ranges, from 73.5% to 51% for precision and 72.5% to 51% for recall. The ctn and abz subsets do not differ more than 2% accuracy between the randomized and the non-randomized training set pairs. The ctn stem precision and recall increase the most between the 75% and full sets, but the abz 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 ctn dataset using approximately 800,000 word tokens for training. My system is maximally tested on 7250 ctn morphemes using only 55,000 training morphemes and achieves 80.3% accuracy. My system also does not assume any language-specific meta-

data, while Samardžić et al. (2015) make use of a ctn lexicon containing high-quality 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 changes 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 jpn dataset. The morpheme ga indicates the nominative case, but is labeled as n in Ex. 2 and nom in Ex. 6. The morpheme ni also appears with different labels, however the difference is due to ni having multiple grammatical functions. In Ex. 2, ni is a postposition, whereas in Ex. 6 ni is a case marker. 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. 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 kor dataset the system relies too heavily on the source...
line, ignoring the correct TRS model predictions.

(7) 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 preference may be more balanced 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.

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

In this case the SRC model predicts 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. Project-specific gram lists may need to be provided for high-confidence gram and stem scores.

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. 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 Conclusions

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.
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 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 ACL.

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. University of Washington, Seattle.

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. Wa- thomigie, 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.

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 Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

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 corpus.

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.

Telma Can Pixabaj, Miguel Angel Vicente Méndez, María 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 Samardzić, 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 Hammarstrøm, and Stephen C. Levinson. 2018. Language documentation twenty-five years on. Language, 94(4):E324–E345.
Libin Shen, Giorgio Satta, and Aravind K. Joshi. 2007. Guided learning for bidirectional sequence classification. In ACL.

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.

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.