Automatic Interlinear Glossing for Otomi language

Diego Barriga[1,3] **Victor Mijangos**[1,3] **Ximena Gutierrez-Vasques**[2,3]
[1]Universidad Nacional Autónoma de México (UNAM)
[2]URPP Language and Space, University of Zürich
[3]Comunidad Elotl
{dbarriga, vmijangosc}@ciencias.unam.mx ximena.gutierrezvasques@uzh.ch

Abstract

In linguistics, interlinear glossing is an essential procedure for analyzing the morphology of languages. This type of annotation is useful for language documentation, and it can also provide valuable data for NLP applications. We perform automatic glossing for Otomi, an under-resourced language. Our work also comprises the pre-processing and annotation of the corpus.

We implement different sequential labelers. CRF models represented an efficient and good solution for our task (accuracy above 90%). Two main observations emerged from our work: 1) models with a higher number of parameters (RNNs) performed worse in our low-resource scenario; and 2) the information encoded in the CRF feature function plays an important role in the prediction of labels; however, even in cases where POS tags are not available it is still possible to achieve competitive results.

1 Introduction

One of the important steps of linguistic documentation is to describe the grammar of a language. Morphological analysis constitutes one of the stages for building this description. Traditionally, this is done by means of interlinear glossing. This is an annotation task where linguists analyze sentences in a given language and they segment each word with the aim of annotating the morphosyntactic categories of the morphemes within this word (see example in Table 1).

Sentence	hí	tó=tsogí
Glossing	NEG	3.PRF=leave
Translation	'I have not left it'	

Table 1: Example of morpheme-by-morpheme glosses for Otomi

This type of linguistic annotated data is a valuable resource not only for documenting a language but it can also enable NLP technologies, e.g., by providing training data for automatic morphological analyzers, taggers, morphological segmentation, etc.

However, not all languages have this type of annotated corpora readily available. Glossing is a time consuming task that requires linguistic expertise. In particular, low-resource languages lack of documentation and language technologies (Mager et al., 2018).

Our aim is to successfully produce automatic glossing annotation in a low resource scenario. We focus on Otomi of Toluca, an indigenous language spoken in Mexico (Oto-Manguean family). This is a morphological rich language with fusional tendency. Moreover, it has scarcity of digital resources, e.g., monolingual and parallel corpora.

Our initial resource is a small corpus transcribed into a phonetic alphabet. We pre-process it and we perform manual glossing. Once we have this dataset, we use it for training an automatic glossing system for Otomi.

By using different variations of Conditional Random Fields (CRFs), we were able to achieve good accuracy in the automatic glossing task (above 90%), regardless the low-resource scenario. Furthermore, computationally more expensive methods, i.e., neural networks, did not perform as well.

We also performed an analysis of the results from the linguistics perspective. We explored the automatic glossing performance for a subset of labels to understand the errors that the model makes.

Our work can be a helpful tool for reducing the workload when manually glossing. This would have an impact on language documentation. It can also lead to an increment of annotated resources for Otomi, which could be a starting point for developing NLP technologies that nowadays are not yet available for this language.

Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas, pages 34–43

2 Background

As we have mentioned before, glossing comprises describing the morphological structure of a sentence by associating every morpheme with a morphological label or gloss. In a linguistic gloss, there are usually three levels of analysis: a) the segmentation by morphemes; b) the glosses describing these morphemes; and c) the translation or lexical correspondences in a reference language.

Several works have tried to automatize this task by using computational methods. In Snoek et al. (2014), they use a rule-based approach (Finite State Transducer) to obtain glosses for Plains Cree, an Algonquian language. They focus only on the analysis of nouns. Samardzic et al. (2015) propose a method for glossing Chintang language; they divide the task into grammatical and lexical glossing. Grammatical glossing is approached as a supervised part-of-speech tagging, while for lexical glossing, they use a dictionary. A fully automatized procedure is not performed since word segmentation is not addressed.

Some other works have approached the whole pipeline of automatic glossing as a supervised tagging task using machine learning sequential models, and they have particularly focused on under-resourced languages (Moeller and Hulden, 2018; Anastasopoulos et al., 2018; Zhao et al., 2020). In Anastasopoulos et al. (2018), they make use of neural-based models with dual sources, they leverage easy-to-collect translations.

In Moeller and Hulden (2018), they perform automatic glossing for Lezgi (Nakh-Daghestanian family) under challenging low-resource conditions. They implement different methods, i.e., CRF, CRF+SVM, Seq2Seq neural network. The best results are obtained with a CRF model that leverages POS tags. The glossing is mainly focused on tagging grammatical (functional) morphemes. While the lexical items are tagged simply as stems.

This latter approach especially influences our work. In fact, Moeller and Hulden (2018) highlight the importance of testing these models on other languages, particularly polysynthetic languages with fusion and complex morphonology. Our case of study, Otomi, is precisely a language highly fusional with complex morphophonological patterns, as we will discuss on Section 3.

Finally, automatic glossing is not only crucial for aiding linguistic research and language documentation. This type of annotation is also a valuable source of morphological information for several NLP tasks. For instance, it could be used to train state-of-the-art morphological segmentation systems for low-resource languages (Kann and Schütze, 2018). The information contained in the glosses is also helpful for training morphological reinflection systems (Cotterell et al., 2016), this consists in predicting the inflected form of a word given its lemma. It also can help in the automatic generation of morphological paradigms (Moeller et al., 2020).

These morphological tools can then be used to build downstream applications, e.g., machine translation, text generation. It is noteworthy that these are language technologies that are not yet available for all languages, especially for under-resourced ones.

3 Methodology

3.1 Corpus

Otomi is considered a group of languages spoken in Mexico (around 300,000 speakers). It belongs to the Oto-Pamean branch of the Oto-Manguean family (Barrientos López, 2004). It is a morphologically rich language that shows particular phenomena (Baerman et al., 2019; Lastra, 2001):

- fusional patterns for the inflection of the verbs (it fuses person, aspect, tense and mood in a single affix);
- a complex system of inflectional classes;
- stem alternation, e.g., *dí=pädi* 'I know' and *bi=mbädi* 'He knew';
- complex morphophnological patterns, e.g., *dí=pädi* 'I know', *dí=pä-hu̱* 'We know';
- complex noun inflectional patterns.

Furthermore, digital resources are scarce for this language.

We focus on the Otomi of Toluca variety.[1] Our starting point is the corpus compiled by Lastra (1992), which is comprised of narrations and dialogues. The corpus was originally transcribed into a phonetic alphabet. We pre-processed this corpus, i.e., we performed digitization and orthographic

[1] An Otomi language spoken in the region of San Andrés Cuexcontitlán, Toluca, State of Mexico. Usually regarded as *ots* (iso639).

normalization.[2] We used the orthographic standard proposed by INALI (INALI, 2014), although we had problems in processing the appropriate UTF-8 representations for some of the vocals (Otomi has a wide range of vowels).

The corpus, then, was manually tagged,[3] i.e., interlinear glossing and Part Of Speech (POS). We followed the Leipzig glossing rules (Comrie et al., 2008).

Domain	Count
Narrative	32
Dialogues	4
Total sentences	1769
Total words (tokens)	8550

Table 2: General information about the Otomi corpus

In addition to this corpus, we included 81 extra short sentences that a linguist annotated; these examples contained particularly difficult phenomena, e.g., stem alternation, reduction of the stem and others. Table 2 contains general information about the final corpus size.

We also show in Table 3 the top ten most common POS tags and gloss labels in the corpus. We can see that the size of our corpus is small compared to the magnitude of resources usually used for doing in NLP in other languages.

POS tags	freq	Gloss	freq
V	2579	stem	7501
OBL	2443	DET	733
DET	973	3.CPL	444
CNJ	835	PSD	413
DEM	543	LIM	370
UNKWN	419	PRAG	357
NN	272	3.ICP	341
NEG	176	LIG	287
P.LOC	81	1.ICP	270
PRT	49	DET.PL	269

Table 3: More frequent POS tags and gloss in corpus

3.2 Automatic glossing

We focus on the two first levels of glossing, i.e., given an Otomi sentence, our system will be able to morphologically segment each word and gloss each of the morphemes within the words, as it is shown in the Example 1. Translation implies a different level of analysis and, due to the scarce digital resources, it is not addressed here.

Similar to previous works, we use a closed set of labels, i.e., we have labels for all the grammatical (functional) morphemes and a single label for all the lexical morphemes (*stem*). We can see in the Example 1 that morphemes like *tsogí* ('leave') are labeled as *stem*.

(1) hí tó=tsogí
NEG 3.PRF=stem

Once we have a gloss label associated to each morpheme, we prepare the training data, i.e., we pair each letter with a BIO-label. BIO-labeling consists on associating each original label with a Beginning-Inside-Outside (BIO) label. This means that each position of a morpheme is declared either as the beginning (B) or inside (I). We neglected O (outside). BIO-labels include the morpheme category (e.g. B-*stem*) or affix glosses (e.g. B-PST, for past tense). For example, the labeled representation of the word *tótsogí* would be as follows:

(2) t ó t s o g
B-3.PRF I-3.PRF B-stem I-stem I-stem I-stem
í
I-stem

As we can see, BIO-labels help to mark the boundaries of the morphemes within a word, and they also assign a gloss label to each morpheme. We followed this procedure from Moeller and Hulden (2018). Once we have this labeling, we can train a model, i.e., predict the labels that indicate the morphological segmentation and the associated gloss of each morpheme.

In this task, the input would be a string of characters $c_1, ..., c_N$ and the output is another string $g_1, ..., g_N$ which corresponds to a sequence of labels (from a finite set of labels), i.e., the glossing. In order to perform automatic glossing, we need to learn a mapping between the input and the output.

3.2.1 Conditional Random Fields

We approach the task of automatic glossing as a supervised structured prediction. We use CRFs for predicting the sequence of labels that represents the interlinear glossing. In particular, we used a linear-chain CRF.

The CRFs need to represent each of the characters from the input sentence as a vector. This is done by means of a feature function. In order

[2]The digitized corpus, without any type of annotation, can be consulted in https://tsunkua.elotl.mx/.

[3]The manual glossing of this corpus was part of a linguistics PhD dissertation (Mijangos, 2021).

to map the input sequence into vectors, the feature function need to take into account relevant information about the input and output sequences (features).

Feature functions play a major role in the performance of CRF models. In our case, we build these vectors by taking into account information about the current letter, the current, previous and next POS tags, beginning/end of words and sentences, letter position, and others (see Section 4.1).

Let $X = (c_1, ..., c_N)$ be a sequence of characters representing the input of our model (a sentence), and $Y = (g_1, ..., g_N)$ the output (a sequence of BIO-labels). The CRF model estimates the probability:

$$p(Y|X) = \frac{1}{Z} \prod_{i=1}^{N} exp\{w^T \phi(Y, X, i)\} \quad (1)$$

Here Z is the partition function and w is the weights vector. $\phi(Y, X, i)$ is the vector representing the ith element in the input sentence. This vector is extracted by the feature function ϕ.

The features taken into account by the feature function depend on the experimental settings, we specify them below (Section 4.1). Training the model consists in learn the weights contained in w.

Following Moeller and Hulden (2018), we used CRFsuite (Okazaki, 2007). This implementation uses the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization algorithm in order to learn the parameters of the CRF. Elastic Net regularization (consisting of L_1 and L_2 regularization terms) were incorporated in the optimization procedure.

3.2.2 Other sequential models

We explored three additional sequential models: 1) a traditional Hidden Markov Model; 2) a vanilla Recurrent Neural Network (RNN); and 3) a biLSTM model.

Hidden Markov Model: A hidden Markov Model (HMM) (Baum and Petrie, 1966; Rabiner, 1989) is a classical generative graphical model which factorizes the joint distribution function into the product of connected components:

$$p(g_1, ..., g_N, c_1, ..., c_N) = \prod_{t=1}^{N} p(c_t|g_t)p(g_t|g_{t-1}) \quad (2)$$

This method calculates the probabilities using the Maximum Likelihood Estimation method. Likewise, the tagging of the test set is made with the Viterbi algorithm (Forney, 1973).[4]

Recurrent Neural Networks: In contrast with HMM, Recurrent Neural Networks are discriminative models which estimate the conditional probability $p(g_1, ..., g_N|c_1, ..., c_N)$ using recurrent layers. We used two types of recurrent networks:

1. Vanilla RNN: For the vanilla RNN (Elman, 1990) the recurrent layers were defined as:

$$h^{(t)} = g(W[h^{(t-1)}; x^{(t)}] + b) \quad (3)$$

   Here, $x^{(t)}$ is the embedding vector representing the character c_t, $t = 1, ..., N$, in the sequence and $[h^{(t-1)}; x^{(t)}]$ is the concatenation of the previous recurrent layer with this embedding vector.

2. biLSTM RNN: The bidirectional LSTM (Hochreiter and Schmidhuber, 1997) or biLSTM uses different gates to process the recurrent information. However, it requires of a higher number of parameters to train. Each biLSTM layer is defined by:

$$h^{(t)} = biLSTM(h^{(t-1)}, x^{(t)}) \quad (4)$$

   where $h^{(t-1)} = [\overrightarrow{h}^{(t-1)}; \overleftarrow{h}^{(t-1)}]$ is the concatenation of the forward and backward recurrent layers.

In each RNN model we used one embedding layer previous to the recurrent layers in order to obtain vector representations of the input characters.

4 Experiments

4.1 Experimental Setup

For CRFs we propose three different experimental settings.[5] Each setting varies in the type of features that are taken into account. We defined a general set of features that capture different type of information:

1. the current character in the input sentence;

[4] We used Natural Language Toolkit (NLTK) for the HMM model.

[5] The code is available on https://github.com/umoqnier/otomi-morph-segmenter/

2. indication if the character is the beginning/end of word;

3. indication if the word containing the character is the beginning/end of a sentence;

4. the position of the character in the word;

5. previous and next characters (character window);

6. the current word POS tag, and also the previous and the next one; and

7. a bias term.[6]

To sum up, the CRF takes the information of the current character as input; but in order to obtain contextual information, we also take into consideration the previous and next character. Grammatical information is provided by the POS tag of the word in which the character appears. In addition to this, we add the POS tag of the previous and next words. These are our CRF settings:

- **CRF**$_{linear}$: This setting considers all the information available, i.e., the features that we mentioned above.

- **CRF**$_{POSLess}$: In this setting we excluded the POS tags.

- **CRF**$_{HMMLike}$: This setting takes into account the minimum information, i.e. information about the current letter and the immediately preceding one. We use this name because this configuration contains similar information as the HMMs but using CRFs to build them.[7]

As previously mentioned, we included other sequential methods for the sake of comparison, i.e., a simple Hidden Markov Model, which can be see as the baseline since it is the simpler model, and two neural-based models: a basic vanilla RNN and a biLSTM model.

The embedding dimension was 100 units for both the vanilla RNN and the biLSTM models.[8] In both neural-based models we used one hidden, recurrent layer; the activation for the vanilla RNN was the hyperbolic tangent. The dimension of the vanilla and LSTM hidden layers was 200.[9]

The features used in the CRF settings are implicitly taken into account by the neural-based models. Except for the POS tags, we did not include that information in the neural settings. In this sense, these last neural methods contain the same information as the CRF$_{POSLess}$ setting.

4.2 Results

We evaluated our CRF-based automatic glossing models by using k-Fold Cross-Validation. We used $k = 3$ due to the small dataset size.

For the other sequential methods, we performed a hold-out evaluation.[10] In all cases we preserved the same proportion between training and test datasets (see Table 4).

	Instances (sentences)
Train	1180
Test	589

Table 4: Dataset information for every model

We report the accuracy, we also calculated the precision, recall and F1-score for every label in the corpus. Table 5 contains the results for all settings.

We can see that the CRF based models outperformed the other methods in the automatic glossing task. Among the CRF settings, CRF$_{HMMLike}$ was the one with the lowest accuracy (and also precision and recall), this CRF used the least information/features, i.e., the current character of the input sentence and the previous emitted label.

This is probably related to the fact that Otomi has a rich morphological system (with prefixes and suffixes), therefore, the lack of information about previous and subsequent characters affects the accuracy in the prediction of gloss labels.

The CRF settings CRF$_{POSLess}$ and the CRF$_{linear}$ are considerably better. The variations between these two settings is small, although the accuracy of CRF$_{linear}$ is higher. Interestingly, the lack of POS tags does not seem to affect the accuracy that much. If the glossing is still accurate (above 90%) after excluding POS tags, this could be convenient, especially in low-resource scenarios,

[6]The bias feature captures the proportion of a given label in the training set, i.e., it is a way to express that some labels are rare while others not.

[7]The maximum number of iterations in all cases was 50.

[8]Both RNN models were trained in similar environments: 150 iterations, with a learning rate of 0.1 and Stochastic Gradient Descent (SGD) as optimization method.

[9]The code for the neural-based models is available on https://github.com/VMijangos/Glosado_neuronal

[10]We took this decision due to computational cost.

	Accuracy	Precision (avg)	Recall (avg)	F1-score (avg)
CRF_{linear}	**0.962**	**0.910**	**0.880**	**0.870**
$\text{CRF}_{POSLess}$	0.948	0.909	0.838	0.856
$\text{CRF}_{HMMLike}$	0.880	0.790	0.791	0.754
HMM	0.878	0.877	0.851	0.858
Vanilla RNN	0.741	0.504	0.699	0.583
biLSTM	0.563	0.399	0.654	0.489

Table 5: Results for the different experimental setups

where this type of annotation may not always be available for training the model.

We do not know if this observation could be generalized to all languages. In the case of Otomi, the information encoded in the features could be good enough for capturing the morphological structure and word order that is important for predicting the correct label.

Additionally, we tried several variations on the hyperparameters of Elastic Net regularization (CRFs), however, we did not obtain significant improvements (see Appendix A).

The model that we took as the baseline, the HMM, obtained a lower performance compared to the CRF settings (0.878). However, if we take into consideration that HMM was the simpler model, its performance is surprisingly good.

The performance of $\text{CRF}_{HMMLike}$ is very similar to that of HMM. As we mentioned before, these two settings make use of the same information, but their approach is different: CRFs are discriminative while HMMs are generative.

The neural approaches that we implemented were not the most suitable for our task. They obtained the lowest accuracy, 0.741 for the vanilla RNN and 0.563 for the biLSTM. This result might seem striking, especially since neural approaches are by far the most popular nowadays in NLP.

5 Discussion

5.1 CRFs vs RNNs

We have several conjectures that could explain why neural approaches were not the most accurate for our particular task. For instance, we observed that the performance of the RNN models (vanilla and biLSTM) was highly sensitive to the frequency of the labels. Both neural models performed better for high frequency labels (such as *stem*).

In principle, the models that we used for automatic glossing have conceptual differences. HMMs are generative models, while CRFs and neural models are discriminative. This distinction, however, does not seem to influence the results. The HMM performed better than the neural-based models but it was outperformed by the CRFs.

CRFs and neural networks mainly in the way they process the input data. While CRFs depend on the initial features selected by an expert, neural networks process a simple representation of the input data (one-hot vectors) through a series of hidden layers which rely on a large number of parameters.

The number of parameters is a key factor in neural networks, they usually have a large number of parameters that allows them to generalize well in complex tasks. For example, the biLSTM model has the highest number of parameters, while the vanilla RNN has a considerably reduced number of parameters.

However, theoretically, a model with higher capacity will also require a larger number of examples to generalize adequately (Vapnik, 1998). The capacity on neural-based models depends on the number of parameters (Shalev-Shwartz and Ben-David, 2014). This could be problematic in terms of low-resource scenarios. In fact, in our experiments, the model with the highest number of parameters, the biLSTM, performed the worst. Models with fewer parameters, such as HMM and CRFs outperformed the neural-based models by a large margin.

It is worth mentioning that we are aware that hyperparameters and other factors can strongly influence neural model's performance. There could be variations that result in more suitable solutions for this task. However, overall, this would probably represent a more expensive solution than using CRFs (or even a HMM).

Our results seem consistent with previous works for the same task where neural approaches fail to outperform CRFs in low-resource scenarios (Moeller and Hulden, 2018).

Complex models with many parameters might not be the most efficient solution in these types of low-resource scenarios. However, we leave this as an interesting research question for the future.

Finally, our proposed models, CRF$_{linear}$ and the CRF$_{POSLess}$, seemed to be the best alternative for the task of automatic glossing of Otomi.

5.2 Linguistic perspective

In this section we focus on the results from a more qualitative point of view. We discuss some inguistic particularities of Otomi and how they affected the performance of the models. We also present an analysis of how the best evaluated method, i.e. CRF$_{linear}$, performed for a selected subset of gloss labels.

As we mentioned in previous sections, the information comprised in the features seems to be decisive in the performance of the CRF models. When some of these features were removed, performance tended to decay.

For the correct labeling of Otomi morphology, contextual information (previous and next characters in the sentence) did have an impact in performance. This may be attributed to the presence of both prefixes and affixes in Otomi words. Stem alternation, for example, is conditioned by the prefixes in the word. Stem reduction is conditioned by the suffixes. In order to correctly label both stem and affixes, the system must consider the previous and next elements.

There exist morphological or syntactic elements in the sentence that contributes to identify words category. For example, most of the nouns are preceded by a determiner (*ri*, singular, or *yi*, plural). This kind of information is captured in the features and can help in the performance of the automatic glossing task.

Frequency of labels is a factor that influence the performance of the models. Labels with high frequency are better evaluated. For the neural-based models the impact of frequency was more pronounced. However, despite of the low-resource scenario we were able to achieve good results with the CRFs (above 90%).

Languages exhibit a wide range of complexity in their morphological systems. Otomi has several phenomena that may seem difficult to capture by the automatic models. However, even when languages have complex morphological systems, there are frequent and informative patterns (e.g. inflectional affixes) that can help to the recognition of them. This hypothesis is reflected in the low entropy conjecture (Ackerman and Malouf, 2013), which concerns the organization of morphological patterns in order to make morphology learnable. This hypothesis points out that morphological organization seeks to reduce uncertainty.

Label	Precision	Recall	F1-score	Instances
DET	1	0.99	1	228
DET.PL	0.99	0.99	0.99	91
3.CPL	0.96	1	0.98	144
PRAG	0.97	0.99	0.98	116
stem	0.96	0.97	0.96	2396
CTRF	0.95	0.97	0.96	89
3.ICP	0.93	0.94	0.94	118
3.PLS	1	0.86	0.92	28
3.PSS	0.80	1	0.89	8
PRT	0.50	0.22	0.31	18

Table 6: Results from the CRF$_{linear}$ model on a subset of the glossing labels

Table 6 presents the evaluation results for a subset of the labels used for the automatic glossing. These labels are linguistically interesting as there is a contrast between productive and unproductive elements.

We can observe that labels like *stem*, 3.CPL (third person completive) or CTRF (counterfactual) were correctly labeled most of the time, as they were systematic and very frequent.

Items like PRT (particle) had lower frequency, a lower recall and lower precision. The lower recall could be attributed to the fact that PRT is not systematic, i.e. multiple forms can take the same label. Therefore, it is more difficult to discriminate.

PRAG (pragmatic mark) appears only in verbs, and always in the same position (at the end of the word), this probably made this mark more easy to discriminate, thus, more easy to predict by the model. It is interesting that this morpheme was relatively frequent but it did not bear semantic information as it only provided discursive nuances (it can be translated as the filler word 'well').

The 3.ICP (third person incompletive) label represents an aspect morpheme which is used very often since it is applied in the present tense and habitual situations. It always appears before the verb and in the same position, it seemed easier to predict. Therefore, this label has a high precision and recall.

The 3.PLS (third person pluscuamperfect) label also shows a systematic use before the verb; however, the latter did have a lower frequency on the

corpus, what seems to have caused a lower recall.

Otomi has two determiner morphemes: one for singular number (DET) and one for plural number (DET.PL). The one for the plural is clearly distinguished from other morphemes as it has the form *yi*. However, for the singular number, the form is *ri* which is the same as the form for the third person possessive (3.PSS). We believe that this fact made the label 3.PSS more prone to be incorrectly labeled (it showed a lower precision). In some cases, the model tended to incorrectly label the form *ri* by preferring the most frequent label DET. This could explain the lower accuracy of 3.PSS compared to DET.

In general, productive affixes were correctly labeled by our automatic system. This may represent a significant advantage in terms of aiding linguistic manual annotation. Productive and frequent morphemes may represent a repetitive annotation task that can be easily substituted by an automatic glossing system.

Even in the understanding that the glossing system is not 100% accurate, it is probably easier for a human annotator to correct problematic mislabels than to do all the process from scratch. In this sense, automatic glossing can simplify the task of manually glossing, and, therefore, it can help in the process of language documentation.

6 Conclusion

We focused on the task of automatic glossing for Otomi of Toluca, an indigenous language with complex morphological phenomena. We faced a low-resource scenario where we had to digitize, normalize and annotate a corpus available for this language.

We applied a CRF based labeler with different variations in regard to the features that were taken into account by the model. Moreover, we included other sequential models, a HMM (baseline) and two RNN models.

CRFs outperfomed the baseline (HMM) but also the RNN models (Vanilla RNN and biLSTM). The CRF setting that took into account more information (encoded by the feature function) had the best performance. We also noticed that excluding POS tags do not seem to harm the system's performance that much. This could be an advantage since automatic POS tagging is a resource not always available for under resourced languages.

Furthermore, we provided a linguistically motivated insight of which labels were easier to predict by our system.

Our automatic glossing labeler was able to achieve an accuracy of 96.2% (and 94.8% without POS tags). This sounds promising for reducing the workload when manually glossing. This can represent a middle step not only for strengthen language documentation but also for facilitating the creation of language technologies that can be useful for the speakers of Otomi.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. We also thank the members of Comunidad Elotl for supporting this work. This work has been partially supported by the SNSF grant no. 176305

References

Farrell Ackerman and Robert Malouf. 2013. Morphological organization: The low conditional entropy conjecture. *Language*, pages 429–464.

Antonios Anastasopoulos, Marika Lekakou, Josep Quer, Eleni Zimianiti, Justin DeBenedetto, and David Chiang. 2018. Part-of-speech tagging on an endangered language: a parallel Griko-Italian resource. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2529–2539, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Matthew Baerman, Enrique Palancar, and Timothy Feist. 2019. Inflectional class complexity in the otomanguean languages. *Amerindia*, 41:1–18.

Guadalupe Barrientos López. 2004. *Otomíes del Estado de México*. Comisión Nacional para el Desarrollo de los Pueblos Indígenas.

Leonard E Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Bernard Comrie, Martin Haspelmath, and Balthasar Bickel. 2008. The leipzig glossing rules: Conventions for interlinear morpheme-by-morpheme glosses. *Department of Linguistics of the Max Planck Institute for Evolutionary Anthropology & the Department of Linguistics of the University of Leipzig. Retrieved January*, 28:2010.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational*

Research in Phonetics, Phonology, and Morphology, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

INALI. 2014. *Njaua nt'ot'i ra hñähñu. Norma de escritura de la lengua hñähñu (Otomí)*. INALI.

Katharina Kann and Hinrich Schütze. 2018. Neural transductive learning and beyond: Morphological generation in the minimal-resource setting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3254–3264, Brussels, Belgium. Association for Computational Linguistics.

Yolanda Lastra. 1992. *El otomí de Toluca*. Instituto de Investigaciones Antropológicas, UNAM.

Yolanda Lastra. 2001. *Unidad y Diversidad de la Lengua: Relatos otomíes*. UNAM.

Manuel Mager, Ximena Gutierrez-Vasques, Gerardo Sierra, and Ivan Meza-Ruiz. 2018. Challenges of language technologies for the indigenous languages of the Americas. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 55–69, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Víctor Mijangos. 2021. *Análisis de la flexión verbal del español y del otomí de Toluca a partir de un modelo implicacional de palabra y paradigma.* Ph.D. thesis, Instituto de Investigaciones Filológicas, UNAM, Mexico City.

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

Sarah Moeller, Ling Liu, Changbing Yang, Katharina Kann, and Mans Hulden. 2020. Igt2p: From interlinear glossed texts to paradigms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5251–5262.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Tanja Samardzic, Robert Schikowski, and Sabine Stoll. 2015. Automatic interlinear glossing as two-level sequence classification. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 68–72.

Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Conor Snoek, Dorothy Thunder, Kaidi Loo, Antti Arppe, Jordan Lachler, Sjur Moshagen, and Trond Trosterud. 2014. Modeling the noun morphology of plains cree. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 34–42.

Vladimir Vapnik. 1998. *Statistical Learning Theory*. John Wiley & Sons.

Xingyuan Zhao, Satoru Ozaki, Antonios Anastasopoulos, Graham Neubig, and Lori Levin. 2020. Automatic interlinear glossing for under-resourced languages leveraging translations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5397–5408.

A Appendix

The following are the detailed results of the three different settings for CRF models. We report average accuracy score. The prefixes in the model names mean whether regularization terms L_1 and/or L_2 were configured.

For example, the prefix *reg* means that both terms were present and conversely *noreg* means that no term is considered. Finally, *l1_zero* and *l2_zero* means if L_1 or L_2 term is equal to zero.

The variation of regularization parameters probed slight improvements between models of the same setting as can be showed in tables 7, 8 and 9.

	Accuracy
CRF$_{HMMLike}$_l2_zero	**0.8800**
CRF$_{HMMLike}$_reg	0.8760
CRF$_{HMMLike}$_noreg	0.8710
CRF$_{HMMLike}$_l1_zero	0.8707

Table 7: CRF$_{HMMLike}$ setting results

	Accuracy
CRF$_{POSLess}$_reg	**0.9482**
CRF$_{POSLess}$_l2_zero	0.9472
CRF$_{POSLess}$_l1_zero	0.9442
CRF$_{POSLess}$_noreg	0.9407

Table 8: CRF$_{POSLess}$ setting results

	Accuracy
CRF$_{linear}$_reg	**0.9624**
CRF$_{linear}$_l2_zero	0.9598
CRF$_{linear}$_l1_zero	0.9586
CRF$_{linear}$_noreg	0.9586

Table 9: CRF$_{linear}$ setting results