LISN @ SIGMORPHON 2023 Shared Task on Interlinear Glossing

Shu Okabe and **François Yvon**
Université Paris-Saclay & CNRS
LISN, rue du Belvédère
91405 Orsay, France
{shu.okabe,francois.yvon}@limsi.fr

Abstract

This paper describes LISN's submission to the second track (open track) of the shared task on Interlinear Glossing for SIGMORPHON 2023. Our systems are based on Lost, a variation of linear Conditional Random Fields initially developed as a probabilistic translation model and then adapted to the glossing task. This model allows us to handle one of the main challenges posed by glossing, i.e. the fact that the list of potential labels for lexical morphemes is not fixed in advance and needs to be extended dynamically when labelling units are not seen in training. In such situations, we show how to make use of candidate lexical glosses found in the translation and discuss how such extension affects the training and inference procedures. The resulting automatic glossing systems prove to yield very competitive results, especially in low-resource settings.

1 Introduction

LISN participated in the 'open track' of the shared task on interlinear glossing of SIGMORPHON 2023 (Ginn et al., 2023) with two submissions. Figure 1 presents the format of the sentences for this shared task. In this track, the source sentence **T** is overtly segmented into morphemes (**M**), which yields an explicit one-to-one correspondence between each source morpheme and the corresponding gloss (**G**), thanks to the Leipzig Glossing Rules convention (Bickel et al., 2008). A translation **L** in a more-resourced language (English or Spanish) is also provided, except for Nyangbo. An obvious formalisation of the task that we mostly adopt, is thus to view glossing as a sequence labelling task performed at the morpheme level.

As can be seen in Figure 1, there are roughly two categories of glosses: *grammatical glosses* indicating the grammatical function of the morpheme (e.g., GEN1) and *lexical glosses* expressing a meaning (e.g., son).[1] While the grammatical glosses of a language constitute a finite set of labels, the variety of lexical glosses is unknown, which is one of the main challenges of the task, especially in small training data conditions.

T	Nesis	łˤono	uži	zown.
M	nesi–s	łˤono	uži	zow–n
G	he.OBL–GEN1	three	son	be.NPRS–PST.UNW
L	He had three sons.			

Figure 1: A sample entry in Tsez: source sentence (**T**), and its morpheme-segmented version (**M**), glossed line (**G**), and target translation (**L**)

To accommodate such cases, we assume that lexical glosses can be directly inferred from the translation tier. Recent works on automatic gloss generation, such as (McMillan-Major, 2020; Zhao et al., 2020), also rely on a similar assumption and leverage the available translations. In our model, we will thus consider that the set of possible labels for the morphemes in any given sentence consists of the union of (a) all the grammatical glosses, (b) lemmas occurring in the target translation, (c) frequently-associated labels from the training data. By using a variant of Conditional Random Fields (CRFs) (Lafferty et al., 2001), which enables such local restriction of the set of possible labels, our glossing model can be viewed as an extension of previous sequence labelling systems based on CRFs such as (Moeller and Hulden, 2018; McMillan-Major, 2020; Barriga Martínez et al., 2021). In our approach, using translations as labels during training raises the issue of aligning the translation and the source sentence, which we handle with the neural word alignment model of Jalili Sabet et al. (2020). As alignments are computed at the morpheme level, this technique does not apply for the 'closed track', where the source segmentation is not part of the training annotations.

Our participation is motivated by two factors:

[1] We consider 'compound' glosses such as 'he.OBL' as lexical glosses in our submission.

Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 202–208
July 14, 2023 ©2023 Association for Computational Linguistics

to evaluate the model performance across varying training data sizes (from a few dozen to thousands of sentences) and to challenge its ability to handle a variety of high-resource languages in the target translation. Section 2 describes our system, while Section 3 presents our experimental settings. Section 4 reports the complete set of results obtained with our models.

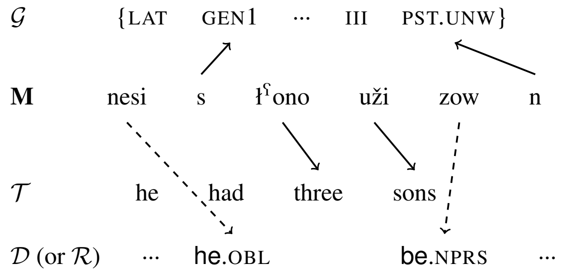

Figure 2: Illustration of our approach to label the example source sentence **M** of Figure 1. 𝒢 represents the set of all grammatical glosses in the training data, 𝒯 the set of words occuring in the translation **L**, 𝒟 the set of lexical labels from the training dictionary, and ℛ the reference lexical labels seen in training. During training, automatic alignments between **M** and 𝒯 are used.

2 System description

Our glossing system uses two main technological components: we (a) rely on an automatic alignment model between the lexical glosses and the target translation during training, which also allows us to exploit additional information regarding target words, such as their Part-of-Speech (PoS) tag or their position; (b) use an extended version of CRFs which allows us to locally restrict the set of possible labels to carry out the glossing task. Figure 2 summarises the main ideas behind our approach.

2.1 Aligning lexical glosses with target words

To align the lexical glosses with the target translation, we use the multilingual aligner SimAlign (Jalili Sabet et al., 2020), which relies on the cosine similarity of the source and target unit embeddings. Three heuristics are available to extract the alignments from a similarity matrix; we use the `Match` method in our submission, since it gave the best results in preliminary experiments. This method considers the alignment task as a maximal matching problem in the bipartite weighted graph containing all possible alignment links between lexical glosses and target words. This heuristic notably ensures that all lexical glosses are aligned with a target word.[2]

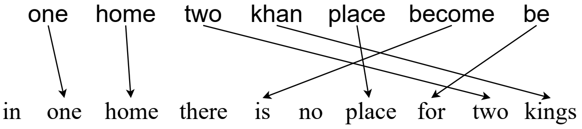

Figure 3: Example of SimAlign alignment between lexical glosses and an English translation (Tsez sentence).

Figure 3 displays an example of alignment computed with the `Match` method. We can note that most alignments are trivial because both units are either identical (e.g. 'one') or have the same lemma (e.g. son/sons). The remaining links are also of great interest in our case. For the alignment pair (khan/'kings'), although the gloss itself is not in the translation, they are synonyms and share valuable properties such as their PoS tag. Besides, the alignment of be with 'for' is obviously wrong and only exists because of the constraint of aligning every lexical gloss. Nevertheless, frequent lemmas such as be occur in multiple sentences, and their possible labels are observed in the training reference annotations.

2.2 Label and label features

Our approach views glossing as a sequence labelling task, meaning that the basic output label for each morpheme is the gloss itself. Our implementation of the CRF model (see below) also enables us to simultaneously predict *label features*, which are arbitrary linguistic properties that can be derived from the label. In our experiments, we chose to incorporate such additional information, which will yield more general, hence more robust, feature functions. In all systems, we thus predict three properties of the label: (a) the actual gloss g, (b) a binary category b about its nature (GRAM for grammatical glosses, or LEX for lexical glosses), and (c) its projected PoS tag p that we collect from the aligned target word.[3]

2.3 Probabilistic sequence labelling model

Our system reuses Lost (Lavergne et al., 2011), a probabilistic model initially devised for statistical machine translation. With Lost, it is possible to label arbitrary segments of a source sentence with

[2] Unless there are more lexical glosses than words in the translation.

[3] As grammatical morphemes have no aligned target words, we use the generic label GRAM for all grammatical glosses.

‘phrases’ from a large bilingual dictionary and to effectively search for the best possible labelling given a set of trained feature weights. Compared to the original translation task, using Lost for automatic glossing brings several simplifications. In particular, there is no need to consider multiple segmentations of the source as the segmentation in morphemes is observed, nor to consider multiple source reorderings, as the translation is also always observed. We thus only focus below on the features of Lost that are relevant for the glossing task.

Lost uses a discriminative model based on the theory of Conditional Random Fields (Lafferty et al., 2001). In a standard CRF, for a sequence $\boldsymbol{x}$ of T observations, the probability of the corresponding label sequence $\boldsymbol{y} \in \mathcal{Y}^T$ is computed as:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z_\theta(\boldsymbol{x})} \exp \left\{ \sum_{k=1}^{K} \theta_k G_k(\boldsymbol{x}, \boldsymbol{y}) \right\}, \quad (1)$$

where G_k are the feature functions with associated weights θ_k and $Z_\theta(\boldsymbol{x})$ is the partition function summing over all possible label sequences. In practice, the features usually test local properties (unigram or bigram). Training is performed by maximising the penalised conditional log-likelihood on a set of fully labelled instances.

Implementing this model for machine translation or for our glossing task is challenging. This is because the set of all possible labels is significantly larger than for most sequence labelling tasks, which means that the computational cost of computing $Z_\theta(\boldsymbol{x})$ can get prohibitive, even for sequences of moderate sizes. The implementation we use, Lost (Lavergne et al., 2011), enables us to specify a local (i.e. for a sentence-specific) set of labels, which defines a restricted *search space* both in training and inference: this means that the normaliser in (1) will only consider a restricted number of possible labellings. Using this implementation, the forward-backward computations performed during training remain tractable, even when the number of possible labels gets extremely large. This feature of Lost is also useful here, as we can restrict the set of possible *lexical* glosses by defining a specific search space for each sentence, as we explain below.

2.4 Defining the search space

During training, we define the search space associated with the source $\boldsymbol{x}$ made of T morphemes as comprising all sequences of T labels from either: the set of known grammatical glosses ($\mathcal{G}$), the lemmas of the words in the translation ($\mathcal{T}$), the most frequent lexical glosses associated with the source morphemes in the training set (this can be viewed as a dictionary $\mathcal{D}$), and the gold glosses ($\mathcal{R}$) for reference reachability (Liang et al., 2006). This ‘simple’ label set comprises two parts: one ($\mathcal{G}$) is common to all sentences, while the remaining labels are defined on a per-sentence basis. In formal terms, the search space is thus $(\mathcal{G} \cup \mathcal{T} \cup \mathcal{D} \cup \mathcal{R})^T$. As explained in Section 2.2, we also consider label features, where the basic labels are augmented with various additional information.

Training the CRF model also requires supervision information, provided here by the reference glosses, from which we readily derive the reference sequence of labels in the search space (an example of reference output labels can be seen on the right-hand part of Figure 4).

During inference, since we have no access to the reference labels, the test search space only comprises the union of the grammatical glosses, the lemmas from the translation, and the labels from the dictionary ($\mathcal{G} \cup \mathcal{T} \cup \mathcal{D}$).[4] Table 1 displays an example output label from each label set for the S1 setting.

set	g	b	p
$\mathcal{G}$	GEN1	GRAM	GRAM
$\mathcal{T}$	king	LEX	NOUN
$\mathcal{D}$	khan	LEX	NOUN
$\mathcal{R}$	khan	LEX	NOUN

Table 1: Example of output labels extracted from each label set (S1 setting), using the example of Figure 3. The reference label set $\mathcal{R}$ is only used during training.

2.5 Feature set

Our two submissions, S1 and S2, use the same model and share most features computed on the source morpheme input. However, the latter extends the former system with additional features.

The input to Lost is the source morpheme s, from which we also deduce the following features: its position p within the word coded as a numerical value (from 0 to n) for complex words, or as ‘F’ for free morphemes, its length l in characters,

[4] When a lemma is both in the translation and dictionary or repeated in the translation, we still create distinct paths in the search space, as these can be associated with different features (e.g. their PoS and position). The search algorithm will then pick the most likely option.

	input	S1 features				S2 features		outputs			S2 features	
i	source	position	length	first 3	last 3	copy	position	reference	GRAM or	PoS	copy	position
	morph. m	(in word) t	l	letters d	letters e	src cs	src ps	gloss g	LEX b	tag p	trg ct	trg pt
0	nesi	0	4	nes	esi	0	1/4	he.OBL	LEX	PRON	0	1/4
1	s	1	1	s	s	0	1/4	GEN1	GRAM	GRAM	-1	-2
2	łˤono	F	5	łˤo	ono	0	2/4	three	LEX	NUM	0	3/4
3	uži	F	3	uži	uži	0	2/4	son	LEX	NOUN	0	4/4
4	zow	0	3	zow	zow	0	3/4	be.NPRS	LEX	VERB	0	2/4
5	n	1	1	n	n	0	4/4	PST.UNW	GRAM	GRAM	-1	-2

Figure 4: Example of input, outputs, and associated features to Lost for the Tsez reference sentence of Figure 1.

and its first and last three letters (d and e respectively). Figure 4 displays an example of input and the associated features.

With all these inputs, we compute unigram and bigram feature functions, detailed in Table 2. On top of the basic unigram and bigram features involving the gloss (top of the table), we also consider the binary category b and PoS tag p to compute more general feature functions (middle of the table). The idea is to capture associations between specific grammatical labels occurring after a given PoS tag (e.g. (VERB, PST.UNW) with the bi-pos-gloss feature).

In the S2 system, we add two more features: first, a binary variable (uni-copy-trg-src), which is True only for lexical glosses that occur letter-for-letter in the source sentence, to account notably for copied words (e.g. proper nouns). Second, we add a categorical feature (uni-pos-src-trg) encoding information about the relative position of the current morpheme with each target word in the translation, to lower the probability of high-distortion source-target associations. This categorical encoding is computed by chunking each sequence into four parts and reporting the chunk numbers: for instance, the value '(1/4, 3/4)' is used when matching a morpheme in the first quarter of the source sentence with a target word in the third quarter of the target sentence. For any unaligned target word, we use -1 as the corresponding position; for grammatical glosses, we assign the value -2 for the corresponding target word.

3 Experimental conditions

3.1 Languages

Our (partial) official submission for S1 considers the following five (out of seven) languages: Tsez (ddo), Gitksan (git), Lezgi (lez), Natugu (ntu; surprise language), and Uspanteko (usp; target translation in Spanish). For our second submission (S2), we could only consider three languages (Tsez, Gitksan, and Lezgi). Since our system relies on the translation to get the lexical glosses, we could not run our models on Nyangbo (nyb), although the corpus has a similar size to other languages we studied. For all submissions, we rely solely on the provided training datasets; no external resource was used.

We have run S2 on Tsez and Uspanteko subsequently and will also report these results below.

3.2 Pre-processing

The PoS tags and lemmas are obtained with spaCy,[5] using the `en_core_web_sm` and `es_core_news_sm` pipelines for English and Spanish translations respectively.

All lemmas from the translation are lowercased except when the associated PoS tag is a proper noun ('PROPN').

3.3 SimAlign settings

Since the glosses and the translation are in the same language, we use the embeddings from the English BERT (bert-base-uncased) (Devlin et al., 2019) when the target language is English and mBERT ('bert-base-multilingual-uncased') when it is in Spanish (for Uspanteko). We can note here that our model is compatible with multiple target languages, SimAlign being an off-the-shelf multilingual (neural) aligner.

Our preliminary experiments showed that the embeddings from the 0-th layer yielded the best alignments, especially compared to the 8-th layer, which seems to work best in most alignment tasks. A plausible explanation is that contextualised embeddings are unnecessary here because lexical glosses do not constitute a standard English sentence (for instance, they do not contain stop words, and their word order reflects the source language word order).

[5] https://spacy.io/.

Feature	Test	Example (cf. Figure 4 $i = 5$)
uni-gloss	$\mathbb{1}(g_i = g)$	PST.UNW
bi-gloss	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(g_{i-1} = g')$	(be.NPRS, PST.UNW)
uni-gloss-morph	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(m_i = m)$	(PST.UNW, n)
uni-gloss-position	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(t_i = t)$	(PST.UNW, 1)
uni-gloss-length	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(l_i = l)$	(PST.UNW, 1)
bi-gloss-morph	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(g_{i-1} = g') \wedge \mathbb{1}(m_i = m)$	(be.NPRS, PST.UNW, n)
uni-gloss-start	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(d_i = d)$	(PST.UNW, n)
uni-gloss-end	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(e_i = e)$	(PST.UNW, n)
uni/bi-bin	$\mathbb{1}(b_i = b)$ $(\wedge \mathbb{1}(b_{i-1} = b'))$	GRAM ((LEX, GRAM))
uni/bi-pos	$\mathbb{1}(p_i = p)$ $(\wedge \mathbb{1}(p_{i-1} = p'))$	GRAM ((VERB, GRAM))
uni-bin-morph/position/length	$\mathbb{1}(b_i = b) \wedge \mathbb{1}(m_i = m)/\mathbb{1}(t_i = t)/\mathbb{1}(l_i = l)$	(GRAM, n) / (GRAM, 1) / (GRAM, 1)
uni-bin-start/end	$\mathbb{1}(b_i = b) \wedge \mathbb{1}(d_i = d)/\mathbb{1}(e_i = e)$	(GRAM, n) / (GRAM, n)
bi-position-bin	$\mathbb{1}(t_i = t) \wedge \mathbb{1}(t_{i-1} = t') \wedge \mathbb{1}(b_i = b)$	(0, 1, GRAM)
bi-bin-gloss	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(b_{i-1} = b')$	(LEX, PST.UNW)
bi-gloss-bin	$\mathbb{1}(b_i = b) \wedge \mathbb{1}(g_{i-1} = g')$	(be.NPRS, GRAM)
uni-pos-morph	$\mathbb{1}(p_i = p) \wedge \mathbb{1}(m_i = m)$	(GRAM, n)
bi-pos-gloss	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(p_{i-1} = p')$	(VERB, PST.UNW)
bi-gloss-pos	$\mathbb{1}(p_i = p) \wedge \mathbb{1}(g_{i-1} = g')$	(be.NPRS, GRAM)
uni-pos-start/end	$\mathbb{1}(p_i = p) \wedge \mathbb{1}(d_i = d)/\mathbb{1}(e_i = e)$	(GRAM, n) / (GRAM, n)
uni-copy-trg	$\mathbb{1}(ct_i = ct)$	-1
uni-copy-trg-src	$\mathbb{1}(ct_i = ct) \wedge \mathbb{1}(cs_i = cs)$	(-1, 0)
uni-posi-ts	$\mathbb{1}(pt_i = pt) \wedge \mathbb{1}(ps_i = ps)$	(-2, 4/4)
uni-gloss-morph-pts	$\mathbb{1}(g_i = g) \wedge \mathbb{1}(pt_i = pt)$ $\wedge \mathbb{1}(m_i = m) \wedge \mathbb{1}(ps_i = ps)$	(PST.UNW, -2, n, 4/4)

Table 2: Unigram and bigram features for our submissions: S1 features about the main gloss label on top, S1 features involving the two other general outputs, and S2 additional features at the bottom.

3.4 Parameter settings

We always use Lost with the default setting, using only the l_1 regularisation penalty $\rho_1 = 0.5$ and keeping the l_2 penalty term to $\rho_2 = 0$. This setting gave the best results on average in our preliminary experiments.

3.5 Metrics

We use the same evaluation metrics as in the Shared Task: morpheme accuracy, word accuracy, BLEU, and differentiated precision, recall, and F1-score for grammatical (gram) and lexical (stem) glosses.

4 Results

Table 3 reports the results for the organiser's baseline[6] and our systems on the development dataset, while Table 4 gives the corresponding test numbers. We only present the word- and morpheme-level (overall) accuracy, which are the two official metrics of the Shared Task results.[7] We also report the results of S2 for Tsez and Uspanteko, which were not available at the time of submission.

model	ddo	git	lez	ntu	usp
baseline	74.2	25.0	32.6	-	75.9
S1	83.6	40.2	84.4	88.2	76.5
S2	84.5*	43.8	85.1	88.5	77.3
baseline	85.0	30.0	50.1	-	81.3
S1	91.0	55.5	87.3	92.1	82.7
S2	91.5*	58.8	88.2	92.4	83.4*

Table 3: Accuracy (overall) at the word (top) and morpheme (bottom) levels for the baseline and our two systems on the *development* dataset. Star-marked values correspond to runs that were not available at the time of submission.

Our systems are consistently better than the baseline, with larger gaps when few training sentences are available (cf. Gitksan or Lezgi). Our second system slightly improves the accuracy on the development set; a similar trend can also be observed on the test set.

Compared to other submitted systems, we reached the best word accuracy for Gitksan and

[6] https://github.com/sigmorphon/2023glossingST/tree/main/baseline.

[7] https://github.com/sigmorphon/2023glossingST/blob/main/results.md.

model	ddo	git	lez	ntu	usp
baseline	75.7	16.4	34.5	41.1	76.6
S1	84.9	28.4	83.4	88.8	76.3
S2	85.5*	31.5	83.0	89.3	76.7*
baseline	85.3	25.3	51.8	49.0	82.5
S1	91.4	50.8	87.2	92.6	82.4
S2	91.8*	51.1	87.0	92.8	82.7*

Table 4: Accuracy (overall) at the word (top) and morpheme (bottom) levels for the baseline and our two systems on the *test* dataset. Star-marked values correspond to runs that were not available at the time of submission.

Natugu and the best morpheme accuracy for Natugu.

5 Discussion

5.1 Impact of training data size

Table 5 displays the evolution of the F1-scores at the morpheme levelCheck this ? broken down by gloss type (lexical and grammatical) for three sizes of the training dataset in Natugu (200, 500, and all 791 sentences). For both settings, the model reaches better scores for grammatical glosses, and, unsurprisingly, lexical glosses benefit more from the increase in training data. While the additional features in S2 were mostly introduced to improve the lexical gloss prediction in the small resource condition, it is noteworthy that they also help improve the prediction of grammatical labels. Similar observations were made for the other test languages.

	S1		S2	
	gram	lex	gram	lex
200	93.3	80.5	93.6	81.3
500	95.3	88.5	95.2	88.3
full	95.7	89.5	95.9	89.6

Table 5: F1-scores for grammatical and lexical glosses with an increasing number of training data in Natugu.

5.2 Number of selected features

Table 6 presents the number of active features (in thousands) selected among all features (in millions) by S1 and S2. We note here that thanks to the l_1-regularisation, most feature weights are set to 0 since less than 1% of the features are actually active. For illustrative purposes, Appendix A lists the features with the largest weight for the Lezgi system.

	ddo	git	lez	ntu	usp
S1	167k (170M)	3k (0.8M)	43k (24M)	60k (39M)	132k (34M)
S2	174k (172M)	3k (0.8M)	46k (24M)	64k (40M)	137k (35M)

Table 6: Number of active features (out of the total number of computed features) for each setting and language.

6 Conclusion

Assuming the lexical glosses can be aligned with words in the target translation, we repurposed a statistical machine translation system based on a globally-normalised model, akin to CRFs, that allows us to dynamically define a local set of labels for the automatic gloss generation task. Using two sets of features, our systems are compatible in low- and very low-resource settings and outperformed the baseline models according to several evaluation metrics.

We plan on further exploring feature functions on both source and target sides. Besides, since our systems rely on automatic alignments, which may contain and project some noise, we will try to remove this dependency modelling alignment as an unobserved variable in a latent variable model. Furthermore, as our submission focused on low-resource data conditions, we did not consider neural methods, which are notably data-intensive; future work would be to integrate word embeddings for better-resourced languages such as Arapaho.

Our code is available at: `https://github.com/shuokabe/gloss_lost`.

Acknowledgements

This work was partly funded by French ANR and German DFG under grant ANR-19-CE38-0015 (CLD 2025). The authors warmly thank Thomas Lavergne for his help and assistance regarding the configuration and exploitation of Lost.

References

Diego Barriga Martínez, Victor Mijangos, and Ximena Gutierrez-Vasques. 2021. Automatic interlinear glossing for Otomi language. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 34–43, Online. Association for Computational Linguistics.

Balthazar Bickel, Bernard. Comrie, and Martin Haspelmath. 2008. The Leipzig Glossing Rules: Conventions for interlinear morpheme-by-morpheme glosses. Leipzig: Max Planck Institute for Evolutionary Anthropology, Department of Linguistics. https://www.eva.mpg.de/lingua/resources/glossing-rules.php.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Ginn, Sarah Moeller, Alexis Palmer, Anna Stacey, Garrett Nicolai, Mans Hulden, and Miikka Silfverberg. 2023. Findings of the SIGMORPHON 2023 shared task on interlinear glossing. In *Proceedings of the 20th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

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, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Thomas Lavergne, Alexandre Allauzen, Josep Maria Crego, and François Yvon. 2011. From n-gram-based to CRF-based translation models. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 542–553, Edinburgh, Scotland. Association for Computational Linguistics.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 761–768, Sydney, Australia. Association for Computational Linguistics.

Angelina McMillan-Major. 2020. Automating gloss generation in interlinear glossed text. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 355–366, New York, New York. Association for Computational Linguistics.

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, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

A Feature weights

Type	Feature	Weight
uni-gloss-start	say ∧ луг	3.22
bi-gloss-morph	say ∧ AOR ∧ лагьа	3.22
bi-gloss-morph	talking ∧ AOC ∧ гафарун	2.80
uni-gloss-morph-pts	. ∧ -1 ∧ . ∧ 4/4	2.75
bi-gloss-morph	fortress ∧ OBL ∧ къеле	2.65
uni-gloss-end	now ∧ ила	2.65
uni-gloss-start	dog ∧ киц	2.65
bi-gloss-morph	newspaper ∧ OBL ∧ газет	2.49
uni-gloss-start	girl ∧ руш	2.49
bi-gloss-morph	SBST ∧ PST ∧ ди	2.49

Table 7: Top 10 (positive) features of S2 for Lezgi.

Table 7 displays the features with the largest weight in the S2 system trained on the Lezgi corpus. We can notice here that some (initial or final) character trigram features (uni-gloss-start and uni-gloss-end) are relevant, corresponding to lexemes that either typically occur with an inflexion mark: 'лугьу' and 'лугьун' for 'say', occurring approximately 200 times together or that combine with a prefix, as 'гила' and 'игила' for 'now' (around 20 co-occurrences).