Predictive Text for Agglutinative and Polysynthetic Languages

Sergey Kosyak
School of Linguistics,
Higher School of Economics
Moscow, Russian Federation
skosiak@hse.ru

Francis M. Tyers
Department of Linguistics,
Indiana University
IN, United States of America
ftyers@iu.edu

Abstract

This paper presents a set of experiments in the area of morphological modelling and prediction. We test whether morphological segmentation can compete against statistical segmentation in the tasks of language modelling and predictive text entry for two under-resourced and indigenous languages, K'iche' and Chukchi. We use different segmentation methods – both statistical and morphological – to make datasets that are used to train models of different types: single-way segmented, which are trained using data from one segmenter; two-way segmented, which are trained using concatenated data from two segmenters; and finetuned, which are trained on two datasets from different segmenters. We compute word and character level perplexities and find that single-way segmented models trained on morphologically segmented data show the highest performance. Finally, we evaluate the language models on the task of predictive text entry using gold standard data and measure the average number of clicks per character and keystroke savings rate. We find that the models trained on morphologically segmented data show better scores, although with substantial room for improvement. At last, we propose the usage of morphological segmentation in order to improve the end-user experience while using predictive text and we plan on testing this assumption by doing end-user evaluation.

1 Introduction

Nowadays text prediction is widely used in different applications such as autocomplete tools, smart keyboards, etc. The used language models are limited by resources, so they can store only the top-N highest frequency words, which may work well with analytic languages, but when it comes to the synthetic languages the out-of-vocabulary (OOV) problem becomes more and more noticeable. In order to deal with this problem, words are usually segmented in constituent parts, so that more of them can be saved in the model vocabulary. Segmentation is almost always done using statistical methods, such as BPE (Gage, 1994). In this paper, we test whether morphological segmentation can improve language modelling and whether it can compete against statistical segmentation methods in predictive text entry task.

The reason to suggest morphological segmentation is that we want text prediction to be both effective and *ergonomic*. By ergonomic we mean that predictions should be linguistically sound and intelligible for the end user. For example, imagine an English word *antidisestablishmentarianism*. An ergonomic segmentation will split the word into its constituent morphs [anti, dis, establish, ment, arian, ism], or an alternative [anti, dis, establishment, arianism]. An unergonomic segmentation might be [antid, isestab, lishme, ntarianism] or [an, tidises, tablishm, entarianism]. One of the issues with many current methods is that while they can produce segments that are meaningful units, in many cases the segments are not linguistically meaningful. We argue that for the task of predictive text entry producing non-linguistic units creates more cognitive load and so will result in slower text entry than predicting the same amount (or a greater number of) linguistic units.

The remainder of the paper is laid out as follows: in Section 2 we overview the languages we experiment on, in Section 3 we discuss the works that were an inspiration for this paper, in Section 4 we describe the experiments we are doing, in Section 5 we review the used segmentation methods, in Section 6 we provide results of language modelling, in Section 7 we speak about language modelling evaluation task, in Section 8 we discuss our thoughts on the results, in Section 9 we announce the planned future experiments. Examples in this paper will be mostly given in K'iche', Chukchi and English. English examples, while English being neither an agglutinative or polysynthetic language, are given in

Proceedings of the first workshop on field linguistics, pages 77–85
October 16, 2022.

order for the reader to better understand the examples.

2 Languages

We perform the experiments using two languages: K'iche' (ISO-639: `quc`), a Mayan language of Guatemala that is of the agglutinating type, and Chukchi (ISO-639: `ckt`), a Chukotko-Kamchatkan language of Siberia of the polysynthetic type. Both of these types are characterised by words consisting of a large number of individual morphs, surface representations of morphemes.

The following examples in K'iche' (1) and Chukchi (2) demonstrate this tendency.[1]

(1) X-in-e'-ki-k'am-a'
CP-B1SG-MOV-A3PL-receive-DEP
'They went to take me'

Both languages exhibit polypersonal agreement (both the subject and object arguments of transitive verbs are encoded on the verb), and Chukchi, in addition, exhibits noun incorporation. As it can be seen in example 2, the object *манэ* /mane/ 'money' is incorporated, rendering intransitive the transitive root *ваӈля* /wanła/ 'ask'.

(2) Нэмыӷэй ны-манэ-ваӈля-сӷэв-ӷэна-т.
nemәqej nә-mane-wanła-sqew-qena-t
also ST-money-ask-MCP-ST.3SG-PL
'They also came to ask for money'

Languages of these types are widespread across the Americas but infrequent in Europe and, as a result, were less researched in terms of predictive text input.

2.1 Data

As K'iche' and Chukchi are low-resource languages, the availability of large corpora is limited. We use data annotated for morphological segments and unannotated text as well. For Chukchi, the annotated data comes from the ChukLang[2] corpus, we use a version that was extracted and converted to Cyrillic orthography to make it compatible with the unannotated corpus. The unannotated data comes from a collection of folklore and texts from the internet.

For K'iche' we also use annotated and unannotated texts. The annotated texts are a hand-segmented set of sentences used in constructing a morphologically and syntactically annotated corpus of K'iche', these sentences come from a range of sources including grammar-book and dictionary examples, stories and legal texts. This corpus is well described in Tyers and Henderson (2021).

The second, unannotated, portion of the data is obtained from the *An Crúbadán* project done by Scannell (2007), that collected corpora from the web for indigenous and marginalised languages.

Table 1 shows the amount of data available for both languages.

	Unannotated		Annotated	
	Sents	Words	Sents	Words
K'iche'	24,254	275,265	1,299	8,789
Chukchi	33,322	151,585	1,006	4,417

Table 1: Dataset sizes for the two languages measured in sentences and words. Unannotated and annotated datasets do not intersect. Annotation was done manually.

2.2 Preprocessing

In order to segment the raw data using a morphological segmentation model the annotated data is split into two disjoint subsets: train (50 percent) and test (50 percent). This ratio is chosen due to low annotated data volume – we suppose that a choice of a disbalanced ratio like 80 percent/20 percent can lead to unreliable results. The automatically segmented corpus is then used for language modelling, while the test split of annotated data is used for predictive text.

3 Related work

Being one of the latest works on language modelling of indigenous languages, Schwartz et al. (2020) proposed the usage of morphological segmentation in order to improve metrics of language modelling. The authors compared different segmentation methods, such as single words, dividing into characters, BPE, Morfessor, Finite-state transducers (FST). Unfortunately, the authors could not do the end-task evaluation of the trained models but suggested doing predictive text as evaluation.

Boudreau et al. (2020), devoted to Mi'kmaq language modelling evaluation, gave us ideas on

[1] Glossing symbols are from the original sources: CP 'completive', B1SG 'absolutive 1st person singular', MOV 'movement prefix', A3PL 'ergative 3rd person plural', DEP 'dependent status suffix, ST 'stative', MCP 'goal-oriented movement', ST.3SG '3rd person singular stative', PL 'plural'.

[2] `https://chuklang.ru/`

how to approach the language modelling task. Mi'kmaq (ISO-639: `mic`), an Eastern Algonquian low-resourse polysynthetic language, is spoken primarily in Eastern Canada and has around 8700 speakers. Not only did the authors work with indigenous language, but they also did the keystroke savings evaluation, which is pretty similar to the predictive text evaluation described in the previous work.

There are other works – Suhartono. et al. (2014); Yu et al. (2017) – that described keystroke savings evaluation. What is more important, the authors worked with agglutinative languages, Bahasa (ISO-639, `ind`), the official language of Indonesia, and Korean (ISO-639, `kor`), official and national language of both North Korea and South Korea (originally Korea). Though we do not want to use the same language modelling technics as were described in the papers, we still find it inspiring there are works dedicated to this task.

As we mentioned before, we assume that the usage of morphs while doing text prediction will make it both effective and ergonomic; in the same time, morphological segmentation brings new challenges. Lane and Bird (2020), devoted to Kunwinjku, a polysynthetic language of northern Australia, and Turkish, showed that morph-based autocomplete for polysynthetic languages can be troublesome due to long words and sparse vocabularies of such languages. Moreover, dialectal variations and dealing with input errors using edit distance makes the next-morpheme predictioning even harder, so, as it is shown in the paper, Turkish may be a more attractive language for morph-based predictioning than Kunwinjku.

4 Tasks

As mentioned previously, our experiments are split into four distinct tasks, from the more fundamental to the more application-specific. In the following sections we describe the methodology for these tasks and the results obtained.

Segmentation We use several segmentation methods in order to compare morphological segmentation and statistical one.

Language modelling We do 10-fold cross-validation in order to train models for end-task evaluation. The evaluation metric is word and character level perplexity. Although the model we use allows both character and word level training, in this paper we do word level training with subwords serving as words.

Predictive text entry We take the trained models from the former task and compare their performance in the predictive text task. The task is to predict the next linguistic unit of output for a given input looking at the top-3 predictions. The evaluation measure is average number of clicks per character and keystroke savings rate. The fewer clicks per character the less the end-user has to type. It is important to mention that the first segment of each word is always typed character by character; this is caused by the model not having token `<bos>` (beginning of the sentence) in its design and the fact that we are doing word level training. As mentioned above, we use the cross-validation models for this task.

Significance testing As the main tasks – language modelling and predictive text – are done using cross-validation, we have sets of results for each model. These results are tested in order to say if some models are significantly better than the others. To do this, first, we do the one-way ANOVA[3] with the null hypothesis being "all the means are the same". In case the null hypothesis is rejected, we then do pairwise Least Significant Difference test (LSD-test)[4] to group the models so that we can find the best performing ones which are not significantly different from each other. The LSD values are given in the appendix.

5 Segmentation

The idea to compare statistical and morphological segmentation was already tested by other researchers; for example, Pan et al. (2020) showed that the usage of morphological segmentation significantly improves the BLEU and ChrF3 metrics in neural machine translation (NMT).

In this paper we want to compare statistical segmentation, presented by Unigram (Kudo, 2018) and WordPiece (Schuster and Nakajima, 2012), and morphological segmentation[5]. We choose NeuralMorphemeSegmentation (NMS; Sorokin and Kravtsova, 2018) for morphological segmentation

[3](2008) One-Way Analysis of Variance. In: The Concise Encyclopedia of Statistics. Springer, New York, NY. https://doi.org/10.1007/978-0-387-32833-1_297

[4](2008) Least Significant Difference Test. In: The Concise Encyclopedia of Statistics. Springer, New York, NY. https://doi.org/10.1007/978-0-387-32833-1_226

[5]We also tried BPE but as the results did not surpass the other systems we exclude them for matters of space and clarity of presentation.

Variant	Example
Input text	*Xke'x ri nukinaq'*
Canonical	x# ke'x$ ri nu# kinaq'$
NMS	x# ke'x$ ri nu# kinaq'$
Unigram	xke# '# x$ ri nuki# na# q'$
WordPiece	xk# e'x$ ri nuk# inaq'$

Table 2: Segmentation variants for the K'iche' sentence *Xke'x ri nukinaq'* "My beans were ground". The canonical segmentation corresponds to /CP-grind.PASS the POSS.1SG-bean/. The hash symbol, #, indicates that there is a segment after the current one and the dollar symbol, $, indicates the last segment in a multi-segment word.

as we have already used it before and it showed good results.

As an output format, as a base we use the stem with singular suffix strategy mentioned in Pan et al. (2020). We modify the strategy, so that all of the subwords are treated the same way: single-morpheme words remain unchanged, in composite words every morpheme except the last one ends with #, the last morpheme ends with $. Table 2 demonstrates the format.

6 Language modelling

Merity et al. (2017) proposed the usage of an AWD-LSTM model for language modelling, showing that it achieves state-of-the-art word level perplexities on Penn treebank and WikiText-2. This model was applied in Schwartz et al. (2020) to several indigenous languages, including Chukchi, and showed good performance. The model trains fast, allows to be trained both on character level and word level, and also is good dealing with overfitting, which is essential while working with low-resource languages.

Although BERT (Devlin et al., 2019) has been successfully used for low-resource languages, Ngoc Le and Sadat (2020) and Wang et al. (2020) showed that models based on BERT models usually have hundreds of millions of parameters and as such are not efficient enough in terms of space for existing mobile phones. This is not suitable for us as our main goal is to use the model for a phone keyboard in order to do predictive text. For all the mentioned reasons we use the described above AWD-LSTM as our model.

The data for language modelling is at first split into modelling (80 percent) and test (20 percent) subsets. Then for the 10-fold cross-validation the modelling subset is split into train (75 percent) and validation (25 percent) subsets. The folds are made using ShuffleSplit[6] with the same seed as the one used while language modelling. The dictionaries for the embeddings consist of all the subwords of train dataset plus the `<unk>` token; the validation subset is used to calculate perplexity in the end of each epoch. The models are trained until 5 epochs without perplexity improvement on a validation subset.

The training hyperparameters are included in the appendix.

6.1 Modelling type

All the models we train can be divided into three types: single-way segmented, two-way segmented and finetuned models.

In order to distinguish a language model from a segmentation method the model names will be given in **bold** e.g. Unigram is a segmentation model while **Unigram** is a model trained on data processed by the corresponding segmentation model.

6.1.1 Single-way segmented

Models of this type – **NMS**, **Unigram**, **Wordpiece** – are trained using datasets from Section 5.

6.1.2 Two-way segmented

Models of this type – **NMS+Unigram**, **NMS+Wordpiece** – are trained using two datasets from Section 5 concatenated together. The idea behind this modelling type is that we want to see if having data processed by different segmentation methods can help us solve both tasks on a high level.

6.1.3 Finetuning

As it was proposed in one of the related works (Boudreau et al., 2020), pretrained embeddings can be used in order to improve the performance of the language models. We check if finetuning will allow us to get better scores both for language modelling and predictive text.

Models of this type – **Unigram2NMS**, **Wordpiece2NMS** – are at first trained using the Unigram/Wordpiece data and then we use morphologically segmented data to finetune the model. Looking ahead we should also mention that it turned out there is no need to lower the learning rate of the

[6] https://scikit-learn.org/0.24/modules/generated/sklearn.model_selection.ShuffleSplit.html

model while finetuning it as it only lengthens the training.

It is worth mentioning that not only embeddings, but also RNN layers are being pretrained.

6.2 Results

All the results are tested as described in Section 4, Significance testing and as we can see in Table 3, the best models for K'iche' and Chukchi according to perplexity are **NMS** and finetuning models.

	K'iche'		Chukchi	
	Wd	Ch	Wd	Ch
NMS	**32.59**	**7.57**	**176.56**	**27.04**
Uni	35.29	8.20	464.43	71.13
WP	148.24	34.45	2745.33	420.48
Uni2NMS	34.32	7.97	**163.58**	**25.05**
WP2NMS	**32.06**	**7.45**	**165.90**	**25.41**
NMS+Uni	34.10	7.92	265.67	40.71
NMS+WP	54.27	12.61	524.28	80.34

Table 3: Word (Wd) level and character (Ch) level perplexities for the models (mean scores of 10-fold cross-validation). **NMS** stands for **NeuralMorphemeSegmentation**, **Uni** stands for **Unigram**, **WP** stands for **Wordpiece**. We do not give subword level perplexities as they are not comparable. The best scores are in **bold** being significantly better according to ANOVA than the others but not outperforming each other.

The two-way segmented models show lower scores than **NMS** ones, though they are better than the models trained on data of their statistical origin (Unigram, Wordpiece segmenters). It does seem like the usage of morphologically segmentated data allows us to improve the performance of the models.

It is worth saying that perplexity scores for different segmentations can not be compared to each other as is due to the dictionary sizes of all the models being different. In order to do so we need to use not subword, but word and character perplexity. Mielke (2019) describes a method of computing them from subword perplexity, so we decide to use the given formulae.

The normalization of scores is done in a following way: at first, the negative log-likelihood of the strings is computed:

$$\text{nll} = \log \text{ppl}^{\text{sw}} * (C_{\text{sw}} + k) \quad (1)$$

where nll is negative log-likehood, ppl^{sw} is the computed subword level perplexity, C_{sw} is the total count of subwords in the set and k is the total count of lines in the set that stands for the count of `<eos>` tokens, which are also predicted by the model.

Then word level and character level perplexities are calculated using the negative log-likehood we get on a previous step:

$$\text{ppl}^{w} = \exp \frac{\text{nll}}{C_w + k} \quad (2)$$

$$\text{ppl}^{c} = \exp \frac{\text{nll}}{C_c + k} \quad (3)$$

where ppl^{w} is word level perplexity, ppl^{c} is character level perplexity, nll is negative log-likehood, C_w is the total count of words in the set, C_c is the total count of characters in the set and k is the total count of lines in the set.

7 Predictive text input

In order to evaluate the models we do predictive text input. The idea is that we automatically emulate a person using a smart keyboard while it is offering some predictions, which have to be meaningful. The meaningfulness is important because we assume that the typing person would like to choose from real words/morphs and not some artificial subwords that make at best no sense and in a worst case scenario they may mean something totally wrong (3). The example is given in Turkish because it illustrates the problem well.

(3) a. araba-m-a
car-POSS.1SG-DAT
'into my car'

b. arab-am-a
arab-*vulgar.word*-DAT
'arab into *vulgar word*'

While evaluating, we look through top 3 model predictions and compare them to the subword we are currently predicting. If they are equal, that prediction is chosen, otherwise we look at the next one. If none of the predictions were correct, we consider that the user will have to finish the word character by character. Thus, a total number of clicks for a word is computed to measure clicks per character metric:

$$\text{CpC} = \frac{\text{keys}_{\text{prediction}}}{\text{keys}_{\text{normal}}} \quad (4)$$

,

where CpC is clicks per character, $\text{keys}_{\text{prediction}}$ is the count of predicted clicks (spaces are included),

$\text{keys}_{\text{normal}}$ is the count of clicks needed to input the word character by character.

We also include the keystroke savings rate used in Boudreau et al. (2020) so that we can compare our results with theirs:

$$\text{KSR} = \frac{\text{keys}_{\text{normal}} - \text{keys}_{\text{prediction}}}{\text{keys}_{\text{normal}}} * 100 \quad (5)$$

,

where KSR stands for keystroke savings rate.

7.1 Results

All the results are tested as described in Section 4, Significance testing and as we can see in Table 4, for K'iche' the best model is **NMS+Wordpiece** and for Chuckhi the best ones are **NMS**, **Wordpiece2NMS** and **NMS+Unigram** – the same group is second best for K'iche'.

Predictive text metrics do correlate with language-modelling metrics; even though **NMS+Wordpiece** performs the best for K'iche', the group of **NMS** and **Wordpiece2NMS** has both best perplexity and clicks per character scores. We suppose that the models that use morphologically segmented data perform better in this task because the used evaluation data, while not being used in language modelling, resembles the training data, as both these sets are morph-based.

The results for Chukchi are worse than the results for K'iche'. The reason may be that gold standard for Chukchi is in Telqep Chukchi, while the corpus used for training is in standard Chukchi. Another reason may be that words in K'iche' evaluation data are shorter both segmentwise and characterwise than the Chukchi words, as shown in Table 5. In case a model can not predict a correct morph, we penalise it by making the whole word be typed character-by-character, so the longer the word is, the more significant mistakes become.

8 Discussion

As we can see, the evaluation shows that there is no single model that outperforms the others in both languages, but models that use morphologically segmented data generally show higher scores. Thus we recommend to try morphological segmentation as it can be used with a statistical one. It is important to mention is that there is no need in training models using morphologically segmented data from scratch, the existing models can be finetuned and the results will not differ significantly from the ones of **NMS**.

	K'iche'		**Chukchi**	
	CpC	KSR	CpC	KSR
No prediction	1.00	0.00	1.00	0.00
NMS	0.96	3.03	**0.99**	**0.78**
Unigram	0.98	1.46	0.99	0.26
Wordpiece	0.97	2.35	0.99	0.20
Unigram2NMS	0.96	3.49	0.99	0.69
Wordpiece2NMS	0.96	3.53	**0.99**	**0.79**
NMS+Unigram	0.96	3.53	**0.99**	**0.73**
NMS+Wordpiece	**0.95**	**4.26**	0.99	0.68

Table 4: Predictive keyboard metrics, the number of clicks per character (CpC) and keystroke savings rate (KSR) for each of the methods. 'No prediction' means that the user has to input all the words character by character including spaces, serving as baseline. The best scores are in **bold** being significantly better according to ANOVA than the others but not than each other.

	SpW	**CpW**
Chukchi	2.54	8.83
K'iche'	1.56	5.20

Table 5: Segments per word (SpW) and characters per word (CpW) metrics of the evaluation datasets.

K'iche' models in all the tasks have better performance than Chukchi models. While we do not know the particular reason for this, we assume that the polysynthetic language complexity may be hindering the model from training. In the mentioned above Lane and Bird (2020) the authors also reported that polysynthetic languages have their special challenges such as high word length, complexity, etc.

As we reference Boudreau et al. (2020), it seems reasonable to compare the results of their experiments with the results of ours. As our task was to predict *linguistic* units, not any kind of units, while in the Mi'kmaq paper words and BPE segments were being predicted, comparison of the results may seem not really correct; though if we do compare the results, we can see that the best KSR score for Mi'kmaq is **3.81**, while the best score for K'iche' is **4.26**. At the same time, the best Chukchi KSR (**0.79**) is much worse that the Mi'kmaq score.

Alongside the metrics we compute there is also a metric which requires end-user testing – the sanity check. As mentioned before, the issue with statistical segmentation is that subwords predicted and offered to the user may have no sense for the user

or, what is much worse, may carry the wrong meaning. We do suppose that this alone can be a reason to choose morphological segmentation over the regular one because segmentation task is not done just for itself – it serves a purpose in a larger scheme of things. We think that in case the language model will be used in predictive text setting, where the user experience and user reaction is highly relevant, morphological segmentation should be chosen as a subword tokenisation method, while statistical segmentation may be chosen for machine translation, for example.

9 Future work

We plan to test several other language models and language modelling metrics in order to find out what correlates best with text prediction scores.

We find it reasonable to experiment on other languages, for example, Nahuatl and Yupik, in order to get a better understanding when the use of morphological segmentation is reasonable.

Another task to do is to run an end-user evaluation of multiple segmentations and determine which units are preferred. In order to do this, we also need to solve the problem of predictive text evaluation that the user has to input the first word character by character – to do this, we will possibly have to combine word level and character level based models.

Acknowledgements

We thank Robert Pugh for his comments and suggestions on an earlier version of this manuscript.

References

Jeremie Boudreau, Akankshya Patra, Ashima Suvarna, and Paul Cook. 2020. Evaluating the impact of subword information and cross-lingual word embeddings on mi'kmaq language modelling. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2736–2745, Marseille, France. European Language Resources Association.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates.

William Lane and Steven Bird. 2020. Interactive word completion for morphologically complex languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4600–4611, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and Optimizing LSTM Language Models. *arXiv preprint arXiv:1708.02182*.

Sabrina J. Mielke. 2019. Can you compare perplexity across different segmentations?

Tan Ngoc Le and Fatiha Sadat. 2020. Revitalization of indigenous languages through pre-processing and neural machine translation: The case of Inuktitut. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4661–4666, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. 2020. Morphological word segmentation on agglutinative languages for neural machine translation.

Kevin Scannell. 2007. The Crúbadán Project: Corpus building for under-resourced languages. In *Proceedings of the 3rd Web as Corpus Workshop*, pages 5–15.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *International Conference on Acoustics, Speech and Signal Processing*, pages 5149–5152.

Lane Schwartz, Francis Tyers, Lori Levin, Christo Kirov, Patrick Littell, Chi kiu Lo, Emily Prud'hommeaux, Hyunji Hayley Park, Kenneth Steimel, Rebecca Knowles, Jeffrey Micher, Lonny Strunk, Han Liu, Coleman Haley, Katherine J. Zhang, Robbie Jimmerson, Vasilisa Andriyanets, Aldrian Obaja Muis, Naoki Otani, Jong Hyuk Park, and Zhisong Zhang. 2020. Neural polysynthetic language modelling.

Alexey Sorokin and Anastasia Kravtsova. 2018. Deep convolutional networks for supervised morpheme segmentation of russian language. In *Artificial Intelligence and Natural Language*, pages 3–10, Cham. Springer International Publishing.

Derwin Suhartono., Garry Wong., Polim Kusuma., and Silviana Saputra. 2014. Predictive text system for bahasa with frequency, n-gram, probability table and syntactic using grammar. In *Proceedings of the 6th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,*, pages 305–311. INSTICC, SciTePress.

Francis Tyers and Robert Henderson. 2021. A corpus of K'iche' annotated for morphosyntactic structure. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 10–20, Online. Association for Computational Linguistics.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual bert to low-resource languages.

Seunghak Yu, Nilesh Kulkarni, Haejun Lee, and Jihie Kim. 2017. Syllable-level neural language model for agglutinative language.

A Hyperparameters

Here we provide hyperparameter values for the various models to aid in reproduction of the results.

A.1 Morphological segmentation

In this section we describe the best hyperparameter settings that we found for the various tasks.

A.1.1 NeuralMorphemeSegmentation

The best results for morphological segmentation are achieved with this hyperparameters:

Parameter	K'iche'	Chuckhi
convolutional layers	3	3
window size	3 – 4	4–6
filters	96	96
dense output users	64	20
context dropout	0.3	0.3
memorize morphemes	no	no
memorize ngram counts.	no	no

Table 6: NMS hyperparameters.

A.2 Least Significant Deviation values

The LSD-test results for language modelling and predictive text tasks (this value is used to arrange the tasks results into groups where all the values have no significant difference):

Task	K'iche'	Chukchi
language modelling	1.494	17.806
predictive text	14.22e-4	6.779e-4

Table 7: LSD values

A.3 Language modelling

All the models based on Merity et al. (2017) are trained with the hyperparameters in Table 8.

Parameter	Value
LSTM layers	3
embedding dim	256
hidden units per layer	3000
use regularization	no
layers dropout	0.4
RNN layers dropout	0.1
embeddings dropout	0.1
remove words from embeddings dropout	0.0
sequence length	100
optimizer	Adam
learning rate	1e-3
weight decay	1.2e-6
seed	1111

Table 8: AWD-LSTM hyperparameters.

B Evaluation

System	Sentence
Raw	*ri tapa'l kub'an k'ax we man ch'ajom taj*
Gloss	'When the *nance*[1] is not washed, it can cause a lot of damage.'
NMS	r i _ t a p a ' l _ k u **b'an** _ **k'ax** _ w e _ m a n _ c h ' a j o m _ **taj** _
Unigram	r i _ t a p a ' l _ k u b ' a n _ k ' a x _ w e _ m a n _ c h ' a j o m _ **taj** _
Wordpiece	r i _ t a p a ' l _ k u b ' a n _ k ' a x _ w e _ **man** _ c h ' a j o m _ **taj** _
Raw	*jawi xkib'ij wi chi ke'e wi*
Gloss	'Where did they say that they would go?'
NMS	j a w i _ x **ki b'ij** _ **wi** _ **chi** _ k e ' e _ w i _
Unigram	j a w i _ x k i b ' i j _ w i _ c h i _ k e ' e _ w i _
Wordpiece	j a w i _ x k i b ' i j _ w i _ c h i _ k e ' e _ w i _
Raw	*kamik kewa' pa taq ri b'e*
Gloss	'Today they will eat on the way.'
NMS	k a m i k _ k e w a ' _ **pa** _ **taq** _ **ri** _ b ' e _
Unigram	k a m i k _ k e w a ' _ p a _ t a q _ **ri** _ b ' e _
Wordpiece	k a m i k _ k e w a ' _ p a _ **taq** _ r i _ **b'e** _

Table 9: Examples of text prediction by single-way segmented models for K'iche' (see Section 6). Underscores indicate word boundaries. Segments in **bold** were correct morph or word guesses. [1] *Byrsonima crassifolia*, a species of flowering plant.