How Low Resource Can You Go? Unsupervised Morphological Segmentation of Lishan Didan

Samuel Miller
University of Maryland – College Park samm@umd.edu

Abstract

There are over 7000 spoken languages in the world today, but many of them quickly and quietly disappearing. These invisible languages often have little to no digital presence: no data to even train an NLP model on. This paper begins exploring just how much data are needed for one task - morphological segmentation on Lishan Didan. Lishan Didan (Persian Azerbaijani Jewish Neo-Aramaic) is critically endangered and a very low-resource language. This paper documents the first public attempt to train an NLP-related model on Lishan Didan, and likely one the very first attempts for any of the Neo-Aramaic languages. It shows the process of gathering solicited data to form a small, linguistically accurate dataset, which is then used to train state of the art morphological segmentation models. Overall, this paper demonstrates that with a dedicated effort, there may be hope for the world's invisible languages; however, it also reveals that we have still got a frustratingly long way to go on even the relatively "simple" task of unsupervised morphological segmentation.

1 Introduction

Why bother to spend a colossal amount of effort to train a model for a language with only a couple thousand elderly speakers? It may seem asinine, but in truth, it is an excellent question to ask.

With each language that dies, humanity takes one more step towards eliminating our cultural diversity. Members of these linguistic communities often find themselves powerless to keep their language and culture alive - and language death is very often not voluntary (Campbell and Belew, 2018). Creating tools to help document, preserve and ultimately propagate these languages are invaluable resources to those who silently watch as their linguistic heritages fade away.

Linguistic inequity in Natural Language Processing is rampant. Joshi et al. present a 6-bin classification system for languages according to their linguistic resources. They find that 88.38% or 2,191 out of the 2,485 languages they analyzed are placed into the lowest class: The Left-Behinds, who have "virtually" no resources (Joshi et al., 2020).

There is clearly quite a bit of work to do to address this inequity.

This paper focuses on one of these "left-behind" languages: Lishan Didan. Lishan Didan, also known as Persian Azerbaijani Jewish Neo-Aramaic, is a modern Aramaic language, which was estimated to have around 4500 elderly speakers in 2001 (eth).

There are 2 major linguistic documentations of Lishan Didan: *The Jewish Neo-Aramaic Dialect of Persian Azerbaijan: Linguistic Analysis and Folkloristic Texts* by Irene Garbell published in 1965, and *The Jewish Neo-Aramaic Dialect of Urmi* by Geoffrey Khan published in 2008. These represent the only available systematic attempts at documentation of Lishan Didan, and both feature a collection of short stories that were spoken in Lishan Didan, and then transcribed (according to different orthographies). Given that he builds upon Garbell's work, Khan's book was used as a reference to ensure everything was done correctly. For the purposes of this paper, Khan's orthographic system will be used. Of note, Lishan Didan does not have a standardized orthography that enjoys use at the time of this paper's composition (Garbell, 1965) (Khan, 2008).

1.1 Focus

Ultimately, this paper aims to address the following question: are good unsupervised morphological segmentations attainable for Lishan Didan, an acutely low resource language with a rich, ambiguous morphology?

2 Related Work

At the time of writing this paper, there are no publicly available works involving both Natural Language Processing and Lishan Didan.

In terms of Morphological segmentation, the MorphAGram project has been successfully used to segment morphologically rich, low-resource languages. MorphAGram uses Adaptor Grammars to aid in segmentation, and it was shown to be quite effective for many languages, including Mexicanero, Nahuatl, Wixarika, and Yorem Nokki, all low-resource indigenous languages of Mexico (Boundary Precision and Recall F1 of around 70 to around 80) (Eskander et al., 2020). It has also been shown to be able to extrapolate to languages of a syntax/morphology different to those that were trained (Eskander et al., 2016) (Eskander et al., 2018). It has even shown to be effective specifically for low-resource polysynthetic languages, both the 4 indigenous Mexican languages mentioned earlier (Eskander et al., 2019), and for Georgian, Inuktitut, and Adyghe (Khandagale et al., forthcoming). This paper attempts to reproduce their methods for Lishan Didan.

This paper uses Morfessor as a baseline, as in these previously mentioned papers. Much work has been done regarding Morfessor and low-resource languages. It has been shown that Morfessor is quite effective at this for Central Yup'ik (a low-resource, polysynthetic language), though the same paper does show that Byte-Pair Encoding ultimately performs just as well if not better for a machine translation task (Liu et al., 2020).

In contrast to this, it has been shown on Nahuatl, Raramuri, Shipibo-Konibo, and Wixarika (4 low-resource polysynthetic languages) that unsupervised methods (Morfessor, FC, LMVR) outperform byte-pair encodings on a machine translation task. However, it was also shown in the same paper that supervised methods outperform both of them, indicating that supervised methods are still the best methods out there (Mager et al., 2022).

Morfessor has also been shown to be effective on the morphologically-rich agglutinative Nguni languages of Southern Africa (the languages used were: isiZulu, isiXhosa, isiNdebele, & siSwati). Their character-level LSTM language model using a entropy-based approach fails to outperform the Morfessor baseline. Though the results aren't ideal, they do show some promise (Moeng et al., 2022). This paper's discussion on canonical and surface segmentations has direct parallels to some features Lishan Didan also has; further discussion on this is in Section 6.1.

Word	Seg. 1	Meaning 1	Seg. 2	Meaning 2
+qtile	qtil-e	He has killed	qti-le	Cut him!
gebi	geb-i	In my house	g-eb-i	In my shame

Table 1: Examples of morphological ambiguity

Dataset	Words
Train	1,411
Test	1000

Table 2: Data split of the datasets

3 Data Collection

From the stories in end of Khan's book, a list of 2,411 unique words were extracted. In the process, the transcriptions had to be "simplified" by removing various diacritics (vowel accents, pause bars, velarization markers). Any suprasegmental features were removed, though future work may wish to consider keeping some (specifically suprasegmental velarization, as there are minimal pairs with that). Some "non-standard" characters were replaced by their unicode equivalents, as the encodings in the book did not translate over well.

Of the 2,411 words, 1000 were manually segmented by the author (who is a speaker of this language) in concordance with the grammars presented in Khan's book. Occasionally, ambiguities would arise, and multiple options would be provided, as exemplified in Table 1. While relatively rare, this does raise the notion that context may be quite important is languages like Lishan Didan with significant morphological ambiguity.

3.1 Datasets

These 1,411 unsegmented & 1,000 segmented words were split into train and dev sets, as shown in Table 2.

3.2 Statistics

Lishan Didan, like other Semitic languages, has a templatic morphology. However, due to extensive contact with with Farsi, Turkish, and Kurdish, it has absorbed some morphemes and words from these languages, which can cause some words to behave more agglutinatively (Khan, 2008). While 1 dimensional segmentation is limiting for templatic languages due to the inability to separate vowels from consonants when transcribed in a Latin script, Lishan Didan still manages to be quite morphologically rich, with an average of 2.406 morphemes per

Prefixes			Suffixes		
w-	kul-	dowr-	-š	-li	-i
reš-	kud-	be-	-ye	-le	-ex
ki-	qulb-	bar-	-lax	-yawe	-ew
m-	k-	ba-	-xun	-la	-etun
wa-	heč-	b-	-lu	-ox	-et
ya-	ma-	geb-	-en	-ta	-lox
xel-	m-	gal-	-ula	-lxun	-lax
	...			...	

Table 3: Examples of some of the affixes provided in scholarly seeded knowledge

Model	F1	Precision	Recall
MorphAGram-st	.259	.352	.230
MorphAGram-ss	.255	.352	.225
MorphAGram-cas	.244	.337	.215
MorphAGr-cas+ss	.232	.325	.305
Morfessor	**.597**	**.591**	**.605**

Table 4: Results of the models for segmenting the test set: average of 5 runs of each. Bolded values are the best in their column.

Unsegmented	+maraqlile	məndaykorpi
Correct Seg.	mar-aql-il-e	mən-d-ay-korpi
Model	**Seg. A**	**Seg. B**
MorphAGram-st	maraqlile	məndaykorpi
MorphAGram-ss	maraqlile	məndaykorpi
MorphAGram-cas	maraqlile	məndaykorpi
MorphAGr-cas+ss	maraqlile	məndaykorpi
Morfessor	mar-aql-ile	mən-day-korpi

Table 5: Undersegmentation of words

word, and an average morpheme length of 2.576 in the 1000 segmented words.

4 Models

Two main models were used: MorphAGram (`github.com/rnd2110/MorphAGram`) and Morfessor (as a baseline). MorphAGram relies on the use of Adaptor Grammars, specifically the Pitman-Yor Adaptor-Grammar Sampler, to train a linguistically-informed morphological segmenter (Eskander et al., 2016). In contrast, Morfessor uses a probabilistic maximum a posteriori model, and is generally understood to be a baseline that is better for larger amounts of input data (Soricut and Och, 2015).

MorphAGram was run on four settings: standard, scholarly seeded, cascaded, and cascaded + scholarly seeded. For the scholarly seeded models, they were provided with a list of common affixes selected by the author from the affixes documented in Khan's book (See Table 3 for some examples). Cascaded approximates this by automatically adding to the grammar. Morfessor was run unsupervised.

4.1 Parameters

MorphAGram was run with the recommended `-r 0 -d 10 -x 10 -D -E -e 1 -f 1 -g 10 -h 0.1 -w 1 -T 1 -m 0 -R`, and 100 iterations.

Morfessor was run using the standard settings.

5 Results

Each model was run 5 times, and an average of the F1, Precision, and Recall across runs is reported in Table 4.

Morfessor clearly outperformed MorphAGram by a large margin. One main reason for this may be that Morfessor has a lot of automatic fine-tuning, whereas the training for MorphAGram were done with custom code, so there is a far larger risk of user error. It also also possible that MorphAGram truly does need more data than this to perform well. It is interesting to note that the more "kowledge" MorphAGram had, the worse it performed (though they are all so close, it does not appear to be a very significant difference).

Be it morphological ambiguity, low amount of training data, or simply user error, Morfessor was far superior at this task. In fact, Morfessor did well enough that the outlook may still be promising.

6 Error Analysis

The outputs seem to indicate that under-segmentation was the largest issue across the board, as shown in Table 5. Rarely morphemes were split: for the most part morpheme boundaries were respected. This indicates that the tolerance may be set too high - allowing the splits to happen more easily can help bring up the scores (at the risk of over-segmenting). Tailoring a more informed starting context-free grammar for Lishan Didan may also help MorphAGram overcome these hurdles.

There is also the issue that some morphemes are parts of others. For example, take the inflectional suffixes: -etun = 2pl, -et = 2ms, and -e = 3ms. The models may have a difficult time segmenting a word like +paltetune as +palt-etun-e, and rather do it as +palt-et-une.

Further, some words happen to end in a way that looks like a suffix, but actually isn't, especially since morphemes are generally so small (at an average of around 2.5 characters per morpheme). This ambiguity may have made the models err on the side of caution and not segment where it should have.

6.1 Canonical vs Surface Segmentation

Going back to Nguni languages of Southern Africa, Moeng et al. note that a word ngezinkonzo may surface as nge-zin-konzo, but may have a canonical form of nga-i-zin-konzo (Moeng et al., 2022).

Lishan Didan also has this ai –> e lexical ambiguity, and it is quite pervasive. For example, in xačangelet, the surface segmentation – which is what is used – is segmented this way: xa-čang-el-et. But the underlying canonical form is xa-čang-a-il-et. This can obscure what should be segmented by creating alternations, and caused further difficulty for these models. Lishan Didan features several such vowel fusions, most often ə reducing to Ø.

Segmenting using the surface form, as this paper has done, may have increased lexical ambiguity dramatically enough to cause poorer performance.

7 Future Work

With such lackluster results, there is definitely a lot more work do. MorphAGram has been shown to work previously, and it remains an enigma as to why it performed so poorly, if not simply because it needs more data. Exploring exactly how and why MorphAGram failed, and if it can be rectified is an area that definitely needs to be explored further. By identifying issues and mitigating them, we may be able to tweak MorphAGram to get its performance up to where we want it to be.

One major way this can be tested is by increasing the amount of data. A method of data augmentation may help achieve this, perhaps akin to what Xia et al. propose using target-side monolingual data alongside pivots from a high-resource language (Xia et al., 2019).

It is also prudent to see if other critically low resource Semitic languages also suffer the same fate to determine whether or not some semitic morphologies are in general problematic for MorphAGram. This is unlikely, however, because MorphAGram has been demonstrated to work well with Arabic, though it does have far more resources than Lishan Didan (Eskander et al., 2016).

As a whole, exploring how these models perform on a diverse array of languages would be a very good way to figure out what exactly went wrong. Purposefully using small amounts of data to train these models, then extracting the linguistic features that tend to fail may prove fruitful in figuring out how to ensure that these models are performing to their highest capabilities.

Conclusion

Although this paper cannot report amazing results, that was not unexpected. Models do need data to train, and sometimes one simply does not have enough. However, these results were also promising. The models were not haphazardly segmenting words, but rather were simply undersegmenting. This is indicative of a concretely addressable problem, and that there is hope that the models' poor performance can be improved. While this paper cannot definitively demonstrate well-formed unsupervised morphological segmentations, it cannot rule out the possibility that MorphAGram is capable of creating such segmentations.

References

Lishán didán. *Ethnologue (18th ed., 2015).*

Lyle Campbell and Anna Belew. 2018. *Cataloguing the world's endangered languages*, volume 711. Routledge New York, USA.

Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith Klavans, and Smaranda Muresan. 2020. MorphAGram, evaluation and framework for unsupervised morphological segmentation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7112–7122, Marseille, France. European Language Resources Association.

Ramy Eskander, Judith Klavans, and Smaranda Muresan. 2019. Unsupervised morphological segmentation for low-resource polysynthetic languages. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 189–195, Florence, Italy. Association for Computational Linguistics.

Ramy Eskander, Owen Rambow, and Smaranda Muresan. 2018. Automatically tailoring unsupervised morphological segmentation to the language. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 78–83, Brussels, Belgium. Association for Computational Linguistics.

Ramy Eskander, Owen Rambow, and Tianchun Yang. 2016. Extending the use of Adaptor Grammars for unsupervised morphological segmentation of unseen languages. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 900–910, Osaka, Japan. The COLING 2016 Organizing Committee.

I. Garbell. 1965. *The Jewish Neo-Aramaic Dialect of Persian Azerbaijan: Linguistic Analysis and Folkloristic Texts*. Ianua linguarum / Series practica. Mouton.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

G. Khan. 2008. *The Jewish Neo-Aramaic Dialect of Urmi*. Gorgias neo-Aramaic studies. Gorgias Press.

Christopher Liu, Laura Dominé, Kevin Chavez, and Richard Socher. 2020. Central yup'ik and machine translation of low-resource polysynthetic languages.

Manuel Mager, Arturo Oncevay, Elisabeth Mager, Katharina Kann, and Ngoc Thang Vu. 2022. Bpe vs. morphological segmentation: A case study on machine translation of four polysynthetic languages.

Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2022. Canonical and surface morphological segmentation for nguni languages. In *Artificial Intelligence Research*, pages 125–139, Cham. Springer International Publishing.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, Denver, Colorado. Association for Computational Linguistics.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. *CoRR*, abs/1906.03785.