

# KU-CST at the SIGMORPHON 2020 Task 2 on Unsupervised Morphological Paradigm Completion

Manex Agirrezabal & Jürgen Wedekind

## Abstract

We present a model for the unsupervised discovery of morphological paradigms. The goal of this model is to induce morphological paradigms from the bible (raw text) and a list of lemmas. We have created a model that splits each lemma in a stem and a suffix, and then we try to create a plausible suffix list by considering lemma pairs. Our model was not able to outperform the official baseline, and there is still room for improvement, but we believe that the ideas presented here are worth considering.

## Introduction

We built a model that gets raw text and a list of lemmas, and returns the set of paradigms for each of those lemmas. The raw text could be the following:

*The aircraft landed at the JFK airport. Other pilots decided to land in Philadelphia. As you may imagine, landing a plane is not an easy job, but imagination can help.*

From there, the model should be able to extract morphological paradigms. Some of them are shown in the table below.

land	decide	imagine
land	decide	imagine
landed	decided	imagined
landing	deciding	imagining

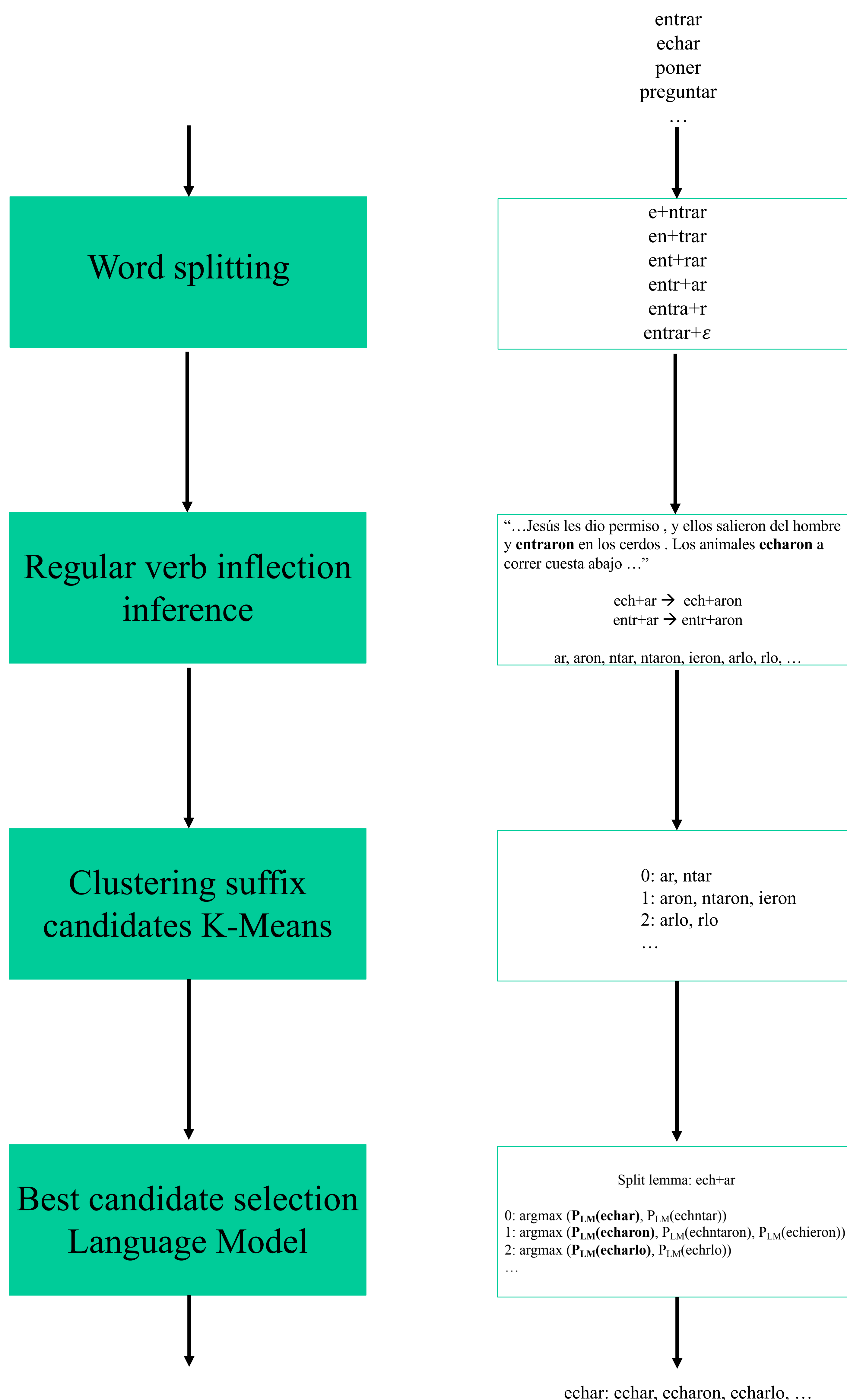
We present a pipeline that assumes that all morphological realizations in a paradigm (for each language) follow a fixed structure:

stem+suffix

Based on that logic, we look for the best candidates to compose the suffix inventory, we cluster them using K-means and after that, we join stems and suffixes. We employ language models to get the most natural outputs.

## Dataset

As one of the most widely extended resources is the bible, this was used as the raw text input data. Together with the bible, a list of verbal lemmas was given. The languages for development were Maltese, Persian, Portuguese, Russian and Swedish. The languages for testing included the following: Basque, Bulgarian, English, Finnish, German, Kannada, Navajo, Spanish and Turkish.



In the first step, for each lemma in the lemma list and each word in the corpus/dictionary, all possible splits are generated.

First, we determine for each splitted lemma the number of potential inflections of the hypothesized stem. Then, in order to find regularly inflecting lemmas, we consider lemma pairs and we look for the biggest intersection of possible suffix sets for each lemma pair and for each split.

In this step we group different realizations of the same suffix with K-Means clustering using a modified version of Minimum Edit Distance as the distance metric, which tries to punish changes that are made at the end of the suffix. Changes of characters of different classes are also considered worse than those of the same class (consonants and vowels).

We check how often a lemma is associated with different splits. The split that happens most frequently will be used as the stem.

Each stem will be joined with one suffix from each cluster. In order to decide which is the best suffix, we use a bigram character-level language model to estimate the probability of the output sequences, trained on the input bible.

Language	Development languages							
	Gold		Baseline		Non-flexible model		Flexible model	
	no. of slots	no. of slots	macro	no. of slots	macro	no. of slots	macro	
Maltese	32	17	0.2029	2	0.013	254	0.0022	
Persian	136	31	0.0605	2	0.0074	11	0.0155	
Portuguese	76	34	0.3964	70	0.1275	1104	0.0109	
Russian	16	19	0.4132	10	0.0706	387	0.0035	
Swedish	11	15	0.4167	17	0.2282	588	0.0093	

Language	Test languages							
	Gold		Baseline		Non-flexible model		Flexible model	
	no. of slots	no. of slots	macro	no. of slots	macro	no. of slots	macro	
Basque	1658	27	0.0006	2	0.0001	30	0.0002	
Bulgarian	54	34	0.3169	13	0.0415	138	0.0299	
English	5	4	0.662	7	0.1729	51	0.0353	
Finnish	141	21	0.055	108	0.0208	1169	0.0039	
German	20	9	0.29	40	0.0498	425	0.007	
Kannada	57	172	0.1512	1	0.0169	44	0.0427	
Navajo	30	3	0.0327	2	0.002	38	0.0013	
Spanish	70	29	0.2367	40	0.1084	225	0.0352	
Turkish	120	104	0.1553	502	0.0071	1772	0.0011	

## Expansion of the lemma list

In order to increase the recall of the model, we decided to extend the lemma list. We obtain new lemmas by training a very simple verb classifier. We create a simple dataset with the input lemmas and some random words from the corpus. We, then, train a simple Logistic Regression model, using character uni-, bi- and trigrams for representing each word. We also include word boundary symbols in the representations. The model that uses the extended list of lemmas for extracting suffixes is called the **Flexible model**, and on the other hand, the initial model (the one that uses only the initial lemmas as input) is called the **Non-flexible model**.

## Discussion and Future work

We assumed each inflected form to be decomposable into a stem and a suffix. This could be, for example, sufficient for English or Spanish, but not for languages such as German that follow a two splits pattern.

Apart from that, a much more straightforward estimate of the morphological richness  $r_m$  could, for example, be obtained by just considering the triple  $l^1 = r^1 + s$ ,  $l^2 = r^2 + s$ ,  $l^3 = r^3 + s$  of optimally splitted distinct lemmas with the maximum number of common suffixes. Because these lemmas are most likely to be frequently used lemmas with regular inflection, the size of the union of their inflections would presumably yield a good estimate of  $r_m$ . Clustering of these triples could also help in identifying verb classes with distinct but regular inflection.

As we have not used any neural network based component, and these would be very useful for learning the morphophonological changes that commonly happen when inflecting words, we would like to incorporate a Sequence-to-Sequence model at the end of our pipeline.