Ship-LemmaTagger: Building an NLP Toolkit for a Peruvian Native Language

José Pereira-Noriega[1], Rodolfo Mercado-Gonzales[2], Andrés Melgar[2],
Marco Sobrevilla-Cabezudo[2], and Arturo Oncevay-Marcos[2]

[1] Facultad de Ciencias e Ingeniería, Pontificia Universidad Católica del Perú, Lima, Peru
jpereira@pucp.pe

[2] Departamento de Ingeniería, Research Group on Pattern Recognition and Applied Artificial Intelligence, Pontificia Universidad Católica del Perú, Lima, Peru
{rmercado,amelgar,msobrevilla,arturo.oncevay}@pucp.edu.pe

Abstract. Natural Language Processing deals with the understanding and generation of texts through computer programs. There are many different functionalities used in this area, but among them there are some functions that are the support of the remaining ones. These methods are related to the core processing of the morphology of the language (such as lemmatization) and automatic identification of the part-of-speech tag. Thereby, this paper describes the implementation of a basic NLP toolkit for a new language, focusing in the features mentioned before, and testing them in an own corpus built for the occasion. The obtained results exceeded the expected results and could be used for more complex tasks such as machine translation.

Keywords: Part-of-speech tagging, Lemmatization, Low resource language, Shipibo-konibo

1 Introduction

Traditionally, both Part-of-speech tagging (POS-tagging) and Lemmatization were made by the use of hand-crafted rules [6]. However, there are several recent studies showing that machine learning approaches are suitable to solve these tasks without taking any effort in defining all the rules and exceptions needed for a particular language.

Specifically, in the case of an agglutinative language like Shipibo-konibo, the labor of building rules is not feasible due to all of the possible combinations of affixes. Also, due to the lack of an established order in the words of a sentence in this language, the labor of developing rules for POS-tagging is particularly time-consuming.

Nevertheless, in order to use machine learning approaches, it is necessary to have an annotated corpus. In this way, since it is easier to build those datasets than the rules, it was decided to follow this learning approach for the develop of our NLP tools for this low resource language.

The paper is organized as follows: in Section 2 are presented some works related to lemmatization and POS-tagging for agglutinative and low-resourced languages. After that, Section 3 describes the case study of this research: the Shipibo-konibo language.

Later, the corpus annotation process is presented in Section 4. Then, Section 5 explains the functionalities developed in this work. Finally, the experiments performed are included in Section 6, and Section 7 presents some conclusions and potential future works.

2 Related Works

In the case of the Shipibo-konibo language, there have not been any direct attempts to solve the problem of POS-tagging or lemmatization. Moreover, this language does not even have an annotated corpus or any computational tool. However, there are some studies for similar agglutinative and low-resourced languages that show some progress in solving these tasks.

For the POS-Tagging task, in languages like Bhojpuri [11] and Bengali [3], the supervised learning approach had a great performance (between 86% and 90% of accuracy). The experiments made for these languages were performed with Support Vector Machines trained models. Also in similar languages like Nepali, approaches based in Hidden Markov Models were used with a little lower results [10].

Regarding the lemmatization task, in languages like Urdu [4] or Mongol [7], it is shown that a rule-based approach can be really effective in solving this problem. However, these studies used manually generated rules, a big corpus, and dictionaries of words to deal correctly with exceptions.

Although, due to the particular agglutinative characteristics of the Shipibo-konibo language, the labor of making manual derivation rules is not feasible. Therefore, it is also possible to develop rules automatically, like it is shown for the Afrikaans [2] and for some European languages [6]. However, since the corpus built for this study is currently smaller than the ones used for those languages, lower results were expected for this work.

3 The Shipibo-konibo Language

Shipibo-konibo (SHP) is the sixth language with highest number of native speakers in Peru. It is a language spoken by about 150 communities (mainly in the Amazon region) and is taught in almost 300 public schools in Peru (schools with a bilingual education program) [8] [1]. However, it does not have any own computational-linguistic resources yet, and this is the reason why it is considered a low-resourced language from a computational perspective, like most of the peruvian native languages.

SHP is an agglutinative language which relies in the use of around 114 suffixes plus 31 prefixes [12] and their combinations for word derivation. However, there is not an official grammar established, so, in order to develop computational-linguistic resources it was a must to relied on the assistance of linguistic experts and bilingual speakers.

4 Corpus: Building and Annotation

Because there is no annotated corpus for SHP, a new one was built with the required data for the job. This task was achieved with the development of an annotation tool called ChAnot[3], the help of linguists with a vast knowledge of the language and some native speakers. It is important to note that they had no experience in annotation tasks. The final corpus for this study is available in a project site[4].

ChAnot is an annotation tool that allows to process a text by sentences and perform morphological (lemma and affixes), morpho-syntactic (POS-tag) and named-entity annotation. This tool was developed to make easier the process of creating an annotated corpus for peruvian low-resourced languages. Unlike annotators tools that allow highlighting parts of the document to annotate some information, the focus of this tool is to process a sentence sequentially word for word, allowing the splitting of its affixes and an specific annotation.

On the other side, a suitable tagset for the language was needed, and since Shipibo-konibo and most native languages in Peru do not have an official tagset, a linguist team defined a new one based on the standard tagset of Universal Dependencies [9] and linguistic studies regarding the language [12]. The tagset match with the UD standard names can be seen in the tool website. With the help of this tool, it was possible to develop a corpus of 219 annotated sentences, where each word of the sentence contains: an annotation of the lemma, POS-tag, sub-POS-tag, and a list of all the affixes that appears in the word.

This corpus was used entirely for the training of the POS-tagger. The distribution of the amount of words per tag in the Shipibo-konibo tagset is shown in Table 1.

Table 1: Structure of the corpus used in the POS-tagging task

POS Category	Quantity
Adjective	66
Adverb	40
Particle	1
Conjunction	38
Determiner	53
Interjection	6
Noun	368
Proper Noun	15
Numeral	6
Interrogative Word	59
Adposition	14
Pronoun	65
Punctuation	311
Verb	361
Auxiliary	95

Table 2: Structure of the corpus used in the Lemmatization task

POS Category	Quantity
Adjective	309
Adverb	130
Particle	1
Conjunction	29
Determiner	4
Interjection	11
Noun	1474
Proper Noun	0
Numeral	6
Interrogative Word	30
Adposition	22
Pronoun	32
Verb	1490
Auxiliary	6

[3] Available in: `chana.inf.pucp.edu.pe/chanot`

[4] Available in: `chana.inf.pucp.edu.pe/resources`

Furthermore, with the help of a Shipibo-konibo dictionary (which entries included POS-tags information), it was possible to identify the derived wordforms of lemmas that were presented in the examples of the use of each entry. In that way, the corpus of the lemmatization task could get more annotated examples. At the end, the corpus achieved a total of 3544 unique input words (with their correspondent lemma and POS-tag) distributed by word category as it is shown in Table 2.

5 Ship-LemmaTagger

5.1 Part-of-speech Tagging

Part-of-speech (POS) tagging is the process of assigning a part of speech to each word in a corpus [5]. For this process, it was defined a tagset aligned to the standard tagset of Universal Dependencies, and after that a supervised learning approach was considered.

The workflow for this step is shown in Figure 1. Firstly, a sentence is received as an input. Then, a tokenization step is performed, where the sentence is split in tokens (words, numbers, symbols or signs).Each token in a sentence is checked to observe whether it is a numeral, a symbol or a punctuation sign. If the token is one of the three possibilities mentioned before, the POS-tag is assigned directly, otherwise the trained supervised model comes into action.

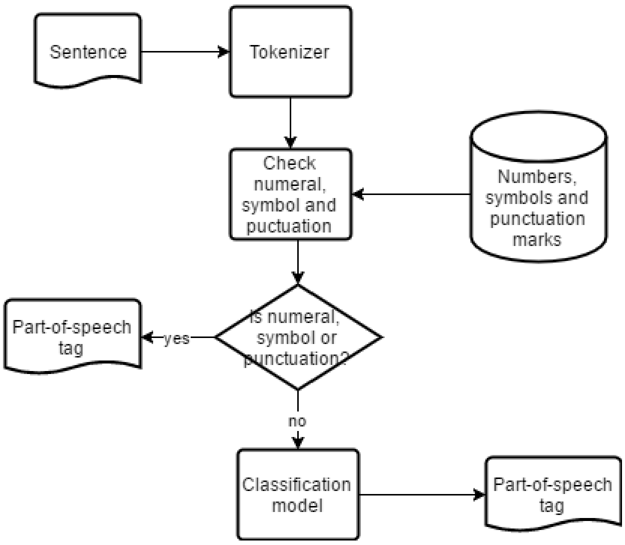
Sentence
Tokenizer
Check numeral, symbol and puctuation
Numbers, symbols and punctuation marks
Part-of-speech tag
yes
Is numeral, symbol or punctuation?
no
Classification model
Part-of-speech tag

Fig. 1: Part-of-speech tagging process

For the classification task, the approach of Ekbal et al. [3] was taken in consideration, since they trained a Support Vector Machine (SVM) algorithm using different features such as word information (initial and final substring of the word, which could be called prefixes and suffixes) and contextual information (previous word, previous POS-tags and next word). The complete list of the generated features is as follows:

- Current token
- Previous token
- Next token
- A binary value that indicates whether it is the first token of sentence or not.
- A binary value that indicates whether it is the last token of sentence or not.
- A binary value that indicates whether the first character of the token is capitalized.
- A binary value that indicates whether all characters of the token are capitalized.
- Prefixes (initial characters) of length 1, 2, 3 and 4.
- Suffixes (last characters) of length 1, 2, 3 and 4.
- Two previous POS-tags.

For instance, in the sentence ”Manaxawe betan chaxo iní” (that means ”The motelo and the deer”), the features regarding the information of word ”Manaxawe” are 1 (first token), 0 (not last token), 1 (first character capitalized), 0 (some characters are not capitalized), ”m”, ”ma”,”man”, ”mana” (prefixes) and ”e”,”we”,”awe”,”xawe” (suffixes). Meanwhile, the features for the contextual information of the word ”chaxo” are ”betan” (previous token), ”iní” (next token), conjunction (previous POS-tag) and noun (POS-tag before previous POS-tag).

5.2 Lemmatization

The lemmatization process follows the workflow shown in Figure 2. First, an individual input token is analyzed in order to determine if it could possess a suffix. This is done by contrasting the end of the word versus a list of all the existent suffixes identified in the Shipibo-konibo language.

In case there is a potential suffix present in the token, a possible rule is inferred with the use of a trained classification model. Once the potential rule is obtained, it is analyzed whether it could be applied for the input token to get the lemma. If the rule could no be used (there is no match) we retrieve the same word as the final lemma.

Regarding the rule prediction task, the approach of W. Daelemans [2] was followed, training a K-NN classification model using a number of features corresponding to the size of the biggest word of the corpus. In this feature vector, each character of the word is mapped to a dimension according to the position of the character in the word. Furthermore, since the language is highly agglutinative on the side of the suffixes, it was decided to reverse the order of the characters in a word to get an alignment between suffixes. On the other side, the derived rules of transformation were considered as the classes of the model.

The lemmatization rules are composed similarly to the ones shown in the previous work: two-elements tuples with (1) the string to be removed and (2) the string to be added to get the final lemma. In both cases, if there is no need to add or remove a string from the input word, the corresponding part of the tuple is left blank.

Additionally, since there are some particular suffixes that only appear in certain words categories, it was decided later to include the POS-tag as an additional feature.

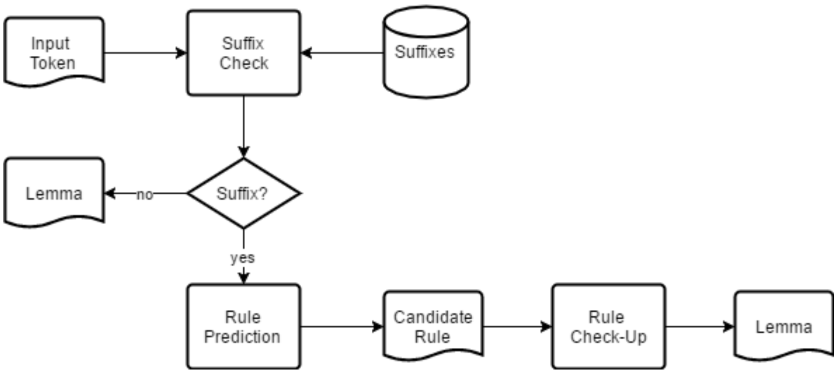

Fig. 2: Lemmatization process

For instance, for the word “ainbobo”, that means “women” in Shipibo-konibo, and its lemma “ainbo”, the features vector would be: [’o’ , ’b’ , o’ , ’b’ , ’n’ , ’i’ , ’a’ , ’noun’] and the rule of transformation would be [bo -] because we need to remove the substring ”bo” from the input word and add a null string (” ”) to get the lemma.

6 Experimentation and Results

Regarding the POS-tagger experiments, two methods were used: an SVM and a Decision Tree model. For the validation step, the corpus was split in sub-datasets for training (70%) and testing (30%). After repeating this split process 10, 50 and 100 times, the average accuracy of our part-of-speech tagger was obtained. The results are presented in Table 3.

Additionally, experiments with ensemble learning methods were tested, but the scores were lower than the expected. Finally, the best overall accuracy was 0.848, obtained with the SVM algorithm (kernel=RBF, C=1, gamma=0.1).

Table 3: Accuracy for part-of-speech tagging experiments

Iterations / **Algorithm**	**10**	**50**	**100**
SVM	0.847	0.847	**0.848**
Decision Tree	0.808	0.810	0.811

For the lemmatization task using the K-NN algorithm, the performance was validated by splitting the corpus in two equal parts for training and testing (50-50). This division was made by stratifying every class of the corpus in two parts, in order to avoid the disproportion of some word categories with little data. This process was performed

100 times with random divisions each time, and the average accuracy obtained is presented in Table 4.

The experiment was fulfilled using different numbers of neighbors and distance metrics in order to find the optimal result. In this way, the best parameters configuration (neighbors=5, distance=Manhattan) achieved an overall accuracy of 0.593. This is caused by the presence of high number of features for this task, and with the Manhattan measure, the relation between near features is isolated and the alignment of the characters obtained more relevance. Also, it is important to notice that the number of neighbors needed for the optimal result should not be too high, since that configuration could bias the results towards the rules with higher appearances in the corpus.

However, this result was not completely satisfactory in itself, but considering that only half of the corpus was used for training, it was a good step to then test it together with the POS-tagger.

Table 4: Accuracy results for the lemmatizer

# of k Metric	1	3	5	7	9	11	13
Euclidean	0.482	0.531	0.558	0.536	0.557	0.534	0.521
Chebyshev	0.486	0.514	0.543	0.539	0.558	0.539	0.541
Manhattan	0.502	0.539	**0.593**	0.562	0.547	0.556	0.551

Finally, both procedures were merged by using the best trained model of the POS-tagging step as an additional feature for the lemmatization. This new lemmatizer model was trained with the whole corpus obtained from the dictionary, and it was tested on the annotated sentences with ChAnot, that include a set of different words. With this procedure, it was obtained a new accuracy value of 81.4% for the trained lemmatizer as it is shown in Table 5.

Table 5: Accuracy results for the joint process

# of k Metric	1	3	5	7	9	11	13
Euclidean	0.805	0.565	0.507	0.486	0.474	0.483	0.476
Chebyshev	0.762	0.531	0.498	0.492	0.474	0.471	0.465
Manhattan	**0.814**	0.574	0.525	0.492	0.489	0.492	0.487

7 Conclusions and future work

This study focus on the developing of a basic NLP toolkit for a new language. As this language (SHP) is an agglutinative one, some approaches in similar contexts were taken in consideration in order to build a solid feature vector to fit learning models for the POS-tagger and Lemmatizer tasks.

The first results were uneven, highlighting the good performance of the POS-tagger. However, despite having achieved an individual low result for the lemmatization task, the integration with the POS-tagging process (as an input feature) led to very promising results in general. Likewise, since the approach used was a corpus-based, the continuous growth of the annotated corpus could lead to better accuracy results for both tasks.

As future work, semi-supervised learning methods will be considered for upcoming experiments. This approach could take advantage of the large unannotated corpus available and, with the integration of the predictive models in the annotation tool, it could support the development of more linguistic resources for this language.

Acknowledgments For this study, the authors appreciate the linguistic team effort that made possible the corpus annotation, and also acknowledge the support of the "Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica" (CONCYTEC Perú) under the contract 225-2015-FONDECYT.

References

1. Acosta, S., Natalia, K., Huamancayo Curi, E., Mori Clement, M., Carbajal Solis, V.: Documento nacional de lenguas originarias del perú (2013)
2. Daelemans, W., Groenewald, H.J., Van Huyssteen, G.B.: Prototype-based active learning for lemmatization (2009)
3. Ekbal, A., Bandyopadhyay, S.: Part of speech tagging in bengali using support vector machine. In: Information Technology, 2008. ICIT'08. International Conference on. pp. 106–111. IEEE (2008)
4. Gupta, V., Joshi, N., Mathur, I.: Design and development of a rule-based urdu lemmatizer. In: Proceedings of International Conference on ICT for Sustainable Development. pp. 161–169. Springer (2016)
5. Jurafsky, D., Martin, J.H.: Speech and language processing, vol. 3. Pearson (2014)
6. Juršic, M., Mozetic, I., Erjavec, T., Lavrac, N.: Lemmagen: Multilingual lemmatisation with induced ripple-down rules. Journal of Universal Computer Science 16(9), 1190–1214 (2010)
7. Khaltar, B.O., Fujii, A.: A lemmatization method for mongolian and its application to indexing for information retrieval. Information Processing & Management 45(4), 438–451 (2009)
8. Ministerio de Educación del Perú: Minedu oficializa alfabetos de 24 lenguas originarias a ser utilizados por todas las entidades públicas. http://www.minedu.gob.pe/n/noticia.php?id=33082, accessed: 2016-31-03
9. Nivre, J., de Marneffe, M.C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C.D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al.: Universal dependencies v1: A multilingual treebank collection. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016). pp. 1659–1666 (2016)
10. Paul, A., Purkayastha, B.S., Sarkar, S.: Hidden markov model based part of speech tagging for nepali language. In: Advanced Computing and Communication (ISACC), 2015 International Symposium on. pp. 149–156. IEEE (2015)
11. Singh, S., Jha, G.N.: Statistical tagger for bhojpuri (employing support vector machine). In: Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on. pp. 1524–1529. IEEE (2015)

12. Valenzuela, P.: Transitivity in shipibo-konibo grammar. Ph.D. thesis, University of Oregon (2003)