The Pashto Corpus and Machine Learning Model for Automatic POS Tagging

Ijazul Haq (<hanjie@sjtu.edu.cn>)  
Shanghai Jiao Tong University  
Weidong Qiu  
Shanghai Jiao Tong University  
Jie Guo  
Shanghai Jiao Tong University  
Tang Peng  
Shanghai Jiao Tong University

Research Article

Keywords: Corpus Creation, CRF, NER, NLP, Pashto, POS tagging, Word Segmentation

Posted Date: March 24th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2712906/v1

License: ☝️  This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License
Abstract

In this study we developed a corpus of the low-resource language – Pashto. The corpus consists of 5 million words, labeled for explicit word-boundaries. And at the time of writing this paper, around 2 million of the words are tagged for Part-of-speech (POS) information. Pashto has no explicit word-delimiter like whitespace in English. The word-boundary markers added to the corpus are not only useful in this study for splitting the text into words, but can also be used in the development of a specialized word segmenter for Pashto. The process of POS tagging was carried out in several rounds where each Round has two sub phases, the automatic POS assignment and manual correction. A specialized web application is developed for manual correction and quality control. The tagset used for tagging is very concise and pragmatic that is developed on the guidelines compatible with the previous standard corpora. In the first Round, the baseline Lexicon-based approach was used for tagging a chunk of 230K word and then manually corrected. Using these tagged words as training data, a Machine Learning (ML) model was trained for tagging the remaining corpus. The purpose of incorporating ML was to improve the accuracy of automatic POS assignment, thus to speed-up the tagging process and reduce the manual effort. Tagging results of the final ML-based model are very satisfactory, which yields an accuracy of 99% and F1-score of 98%. Besides building an automatic POS tagger, the proposed corpus is aimed to be used in countless open research areas in Pashto NLP such as homographs disambiguation, NER, word segmentation, text proofing, constituency and dependency parsing and language modeling etc.

1. Introduction

The recently developed ChatGPT is one of the best Artificial Intelligence (AI) chatbots. Based on the generative pre-trained transformer (GPT) (Brown, Mann et al. 2020), it is designed to generate human-like responses to natural language inputs. The model performs quite well on the major languages such as English, Spanish and German but its results are very disappointing for Pashto. One of the reasons is non-availability of large-scale corpora and rich datasets of this low-resource language. Despite being the language of more than 50 million people, research on Pashto NLP is still in infancy. Of-the-shelf tools are hardly available even for the basic NLP tasks, such as tokenization, word segmentation, proofing and NER etc. This study is aimed to fill this research gap by developing an NLP friendly corpus of 5M segmented words, where 2M words are tagged with POS labels. These figures are huge for a language with such scarce digital resources.

Annotated corpora constitute a very useful tool for research, and POS information are very helpful for the model to better understand the language. This information can later be used in text analysis algorithms to capture the syntactic and semantic relations between words. To understand the in-depth meaning of a sentence, build a knowledge graph or extract relationships, POS tagging is a very important step. POS tagging is the process of assigning a grammatical category to each word in a sentence, such as noun, verb, adjective, or adverb. With POS information added to each word, AI models would be able to learn more about the grammatical structure of the Pashto text which could improve their ability to generate coherent and grammatically correct text. A corpus can be annotated with different kinds of information
such as grammatical, semantic, phonetic, lexical, or prosodic annotation. Grammatically tagged corpus is the most common form of annotated corpora, it is one of the first types of annotation to be performed on corpora and is still used, e.g., (Bonneau-Maynard, Quignard et al. 2009, Camacho-Collados, Delli Bovi et al. 2019, Szymanik and Kieraś 2022). The British National Corpus (Aston and Burnard 1998), Brown Corpus (Francis and Kucera 1979), and LCMC (McEnery and Xiao 2004) are examples of grammatically annotated corpora.

POS tagging is seemed to be intuitive but in fact, it is not a straight-forward task, as a particular word can have different tags based on the context in which the word is used. For example: In the sentence “he is walking”, “walking” is a Verb in the present continuous tense but in the sentence “he enjoys walking”, “walking” is a Gerund. Similarly, in Pashto in the sentence “<it's snowfall on the mountain>”, the word <snow> is a Noun but in the sentence “<listen to me>”, the same word <listen> is an Imperative Verb. This type of situation is known as homograph ambiguity where the same-looking words can have more than one meaning. POS annotation can solve these types of issues by giving an “added value” to the corpora and increasing the specificity of data retrieval.

Pashto is a low resource language and digital content is not very abundant. Large-scale corpora and well-organized datasets can hardly be found, therefore, researchers of Pashto NLP waste a lot of time and resources just in data collection and preprocessing. The previous work on Patho POS tagging is almost negligible. In (Rabbi, Khan et al. 2009) a rule-based technique is used for POS tagging where the author has claimed an accuracy of 88% on a corpus of 100K words. In another study (Khan, Ali et al. 2020), the ML technique is used that achieved 97% accuracy, but the corpus is very small, has 18K words, and another limitation is the small tagset that has only 14 tags.

The Pashto POS tagged corpus proposed in this study is aimed to be used in countless NLP tasks, however, in this study we have discussed only the very basic use-cases. Pashto is not a standardized language and there are no golden rules for the proper use of whitespace in the writing system. The inconsistent use of whitespace leads to two typical spelling errors, the space-omission and space-insertion error. The issue of space-omission arises when the whitespace between two words is ignored and the words stuck together, thus two or more words technically become a single string. On the other hand, space-insertion issue occurs when a (useless) whitespace is inserted within a word and split a single word into two or more sub-parts. These errors introduce enormous amount of noise in the text and make it hard for NLP application to identify the start and end of the word. The problem of space-omission and space-insertion is not limited to Pashto only but other sister languages also face the same issues. Discussion and proposed solution for Persian and Urdu language can be found in (Panahandeh and Ghanbari 2019) and (Lehal 2010, Rashid and Latif 2012) respectively, however for Pashto language, to the best of our knowledge, no such study is previously available. The proposed corpus can be used in the development of AI models for identifying the correct position of whitespace in the text and removal of the space-omission and space-insertion issues. Another use-case of this corpus is to develop a specialized word segmenter for Pashto. The current NLP research is dependent on the baseline whitespace tokenization, but whitespace is not a reliable word delimiter in Pashto. Around 3% of the words in Pashto
are compound-words – words having more than one parts separated by whitespace, i.e., New York. A whitespace tokenizer will break-down these compound-words into two or more (probably meaningless) parts. The corpus can be used to train an AI model that will capture the morphological and contextual features of the tokens to identify the word-boundary. Apart from these basic use-cases mentioned here, the corpus can be used in countless NLP applications such as Named Entity Recognition (NER), homographs disambiguation and dependency parsing etc. The contributions we made in this study can be summarized as follows:

- We developed a general-purpose Pashto text corpus of 5M words, manually corrected for space-omission and space-insertion errors, marked with explicit word-boundaries, and annotated 2M of the words for POS information.
- Using the corpus, we developed a ML-based automatic POS tagger for Pashto words.
- We explained some basic use-cases of the corpus

2. Pashto Language

Pashto is written in Perso-Arabic script, it is also cursive in nature and written right-to-lift (RTL). And like Arabic and Persian there is no upper-case and lower-case distinction in letters. Pashto uses a modified version of the Arabic alphabet with some additional letters to indicate certain sounds that Pashto has and Arabic doesn’t. And therefore, Pashto has a total of 44 alphabets as shown in Fig. 1, while standard Arabic has 28.

Grammatically, Pashto is a SOV (Subject -> Object -> Verb) language with split ergativity. The SOV order is not very strict and the Object can sometimes come before Subject, becoming OSV without changing the voice from active to passive, as shown in Fig. 2. Pashto has a rich morphology and intricate inflection mechanism where prefixes and suffixes add a lot of information to the meaning of the word. Adjectives and Nouns are inflected for number (singular/plural), gender (feminine/ masculine) and case (direct/oblique). The use of singular and plural in Pashto is similar to that of English, the nouns denoting one object are singular and those denoting more than one are plural. The verb system in Pashto is very complex that is basically divided into two tenses on morphological basis, the present tense and the past tense. The future tense is the same as the present tense but preceded by the particle “”. Every tense has two aspects, imperfective that indicates continuous actions and perfective that indicated completion of an action. Present and imperative forms are based on present tense while past, optative, and infinitive forms are based on past tense. Further reading about Pashto and grammatical rules on which this study is based can be found in (Penzl 1962, David 2013, Khan and Wazir 2020).

1.1 USE OF WHITESPACE

In English language, a whitespace is essential between words, where in languages like Chinese and Japanese a whitespace is not used for word separation. In Pashto language however, a whitespace is used for word separation but not consistently. Pashto is not standardized and there are no golden rules
for the proper use of space especially in typing (writing with keyboard). The improper use of whitespace leads to two common spelling errors, the space-omission and space-insertion. To understand these errors, we should first know about the two categories of Pashto alphabets, (i) the Breakers, i.e., “ ” and (ii) Non-breakers – all the remaining 33 alphabets. Breakers are the characters that will not join to the following character and therefore if a word ends with a breaker character the space-insertion after the word is not compulsory. For example, the phrase has two words,  <my> and  <friend>, that are not separated by whitespace but still perfectly readable, because the word “ ” ends with a breaker character “ ”. In English, if “myfriend” is written instead of “my friend”, it's considered an obvious typo. In Pashto however, this is not considered as typo, and a human reader may not even notice anything wrong. But, in NLP domain it leads to a very serious issue, known as space-omission, where the phrase will be considered as a single word by computer. On the other hand, when a useless space inserted within a word it leads to the space-insertion issue, e.g., writing  instead of  <advancement>. A space-insertion error can be compared to the English typo of writing “advance ment” or “adva ncment” instead of “advancement”. Figure 3 show an example sentence highlighting the two types of spelling errors.

1.2 PASHTO WORD

There is no consensus among linguists on the definition of word and the criterion of the concept remain controversial. But in this document, “word” refers to the basic element of the Pashto language that communicates a meaning and can exist independently. A word is semantically complete and several words can be put together to form a sentence. A Pashto word may consist of a single string of characters with no whitespaces – called simple-word i.e.,  <fast>,  <sobbing> and  <to examine> etc. Two or more simple-words may join to form compound-words having whitespaces between the sub parts, i.e.,  <hardworking> and  <confused>. Majority of the words in Pashto are simple-words that makes around 80% of the lexicon but the repetition frequency of compound words is much lower than simple-words and thus the corpus has around 3% compound-words. Most of the compound words are Proper Nouns (Named Entities) i.e.,  <Elbert Einstein> or  <Saudi Arabia>. The more the number of sub-parts in a compound word the greater is the possibility it's a Proper Noun.

3. Human Annotators And Their Background

In the creation of corpus, we tried our best to reduce manual effort, but the involvement of human annotators in corpus annotation is inevitable. The Pashto language is not yet explored very well in the area of NLP, of-the-shelf tools and resources are not available and computer readable data can hardly be found. The rule of the human annotators will be explained in the coming sections but here we want to just give an introductory note about their background. We were supplemented by four volunteers, all of them are native Pashto speakers and graduated in Pashto language. One of the authors himself is a native Pashto speaker and have gained enough understanding of Pashto grammar and morphology before starting this project. For correct annotation with reasonable speed in both the word segmentation and POS tagging, understanding the structure of the sentences is essential. Even with the best sets of
guidelines provided to the annotators, we made sure that the annotators have received considerable training in linguistics, particularly in grammar and syntax.

4. Data Collection

Digital content of Pashto is not very abundant and researchers usually waste a reasonable amount of time and resources in data collection alone. And as usual, this task was very challenging in this study also. For this project we needed two types of textual data, (i) Pashto sentences, and (ii) Pashto Lexicon (or Dictionary)

4.1 COLLECTION OF PASHTO SENTENCES

The text for the corpus is sampled from three sources, news, books and Wikipedia articles. The largest portion is dedicated to news which make 85% of the corpus. Two news sources BBC and VOA are dominant, making around 50% of the corpus while the other half consists of Taand, Khyber News, Khabaryal, Khyber.org, Lekwal and Toolafghan. The books included in the corpus belong to different categories, i.e., religion, poetry, fairytales, politics, novels, current affairs, health and academic dissertations. In data sampling we were biased towards those sources where the content was written on best practice i.e., have fewer space-omission and space-insertion errors. All the text was converted to sentences where two types of delimiters (sentence boundaries) were used for splitting the text, the full stop “.” and the Pashto full stop “-”. Initially, a raw corpus of 260K sentences (around 6M) was prepared. Sentences having words from other languages were removed and included only those sentences that are written in pure Pashto. Too long and very short sentences are excluded. The average length of sentences in the final corpus is around 106 characters. The histogram in Fig. 4 shows the number of words in sentences.

4.2 LEXICON CREATION

For corpus creation a comprehensive lexicon is a prerequisite. A lexicon is a well-formatted list of words with grammatical information. To build such a large-scale lexicon from scratch was quite challenging, thus we initially obtained a digital Pashto dictionary (CreativeXLab). It is one of the best and well-organized digital lexicons available at the time of writing this paper. The lexicon was scanned word by word with semi-automatic tools and converted to the desired format. The size of Lexicon was initially relatively small, having 25K words, however, it gave us a foundation and then we added more words manually. Most of the newly added entries were the frequent words from the corpus and commonly used people and places names. As the Pashto is spoken mainly in Afghanistan and Pakistan, therefore, detailed gazetteer of both the countries were added. Finally, a lexicon of 45K words was prepared, that was quite enough for this study.

5. Tagset
A Tagset is a list of POS tags/labels used to indicate the part-of-speech of each word in a text corpus, examples are Brown Corpus Tagset (Francis and Kucera 1979) and CLAWS4 (Leech, Garside et al. 1994) tagset of the British National Corpus (BNC). The tagset we have selected is derived from the one proposed by I. Rabbi et.al (Rabbi, Khan et al. 2009) for Pashto language. But some customizations are done in order to avoid redundancy, make the boundary between word categories clearer and achieve high annotators agreement. The first thing we changed is that we used the naming convention of Brown Corpus Tagset where the same is used for the Penn Treebank also (Marcus, Santorini et al. 1993). The reason to follow a commonly known convention is just to make the tagset more comprehensible for readers who are already familiar with the previous corpora. The Brown Corpus Tagset lacking some categories that are necessary in Pashto, for that we have used our own self-explanatory tags i.e., “INF” for infinitive and “IMP” for imperative. Personal pronouns are further categorized into 1st, 2nd and 3rd person pronouns for which Romanized numbers are used. Ambipositions are merged with postpositions and assigned PT tag. For negation marker i.e., “not”, “un” etc., various symbols are used in previous tagsets but we have assigned them the “NG” tag. The tagset of I. Rabbi et. al has a total of 54 tags where 17 tags are assigned to punctuations but we have assigned one “PU” tag to them. The tagset we have used is very concise and pragmatic, and the minimum possible tags are included that covers all the words and sufficient for future syntactic parsing and semantic field annotation. The final tagset has 38 tags, as shown with description in Table 1. Disagreement of the linguists is respected, and a non-tagged version of the corpus is also included that can be tagged using any preferred tagset.
### Table 1
Tagset of 38 tags developed for tagging the corpus

<table>
<thead>
<tr>
<th>#</th>
<th>Tag</th>
<th>Detail</th>
<th>#</th>
<th>Tag</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AB</td>
<td>Abbreviation</td>
<td>20</td>
<td>PRPi</td>
<td>Personal Pronoun – 2nd Person</td>
</tr>
<tr>
<td>2</td>
<td>BA</td>
<td>Future-maker Particle</td>
<td>21</td>
<td>PRPiii</td>
<td>Personal Pronoun – 3rd Person</td>
</tr>
<tr>
<td>3</td>
<td>CC</td>
<td>Conjunction</td>
<td>22</td>
<td>PRW</td>
<td>Interrogative Pronoun</td>
</tr>
<tr>
<td>4</td>
<td>DT</td>
<td>Determiner</td>
<td>23</td>
<td>PT</td>
<td>Postposition</td>
</tr>
<tr>
<td>5</td>
<td>FW</td>
<td>Foreign Word</td>
<td>24</td>
<td>PU</td>
<td>Punctuation</td>
</tr>
<tr>
<td>6</td>
<td>FX</td>
<td>Affix (Prefix or Suffix)</td>
<td>25</td>
<td>RB</td>
<td>Adverb</td>
</tr>
<tr>
<td>7</td>
<td>JJ</td>
<td>Adjective</td>
<td>26</td>
<td>RP</td>
<td>Particle</td>
</tr>
<tr>
<td>8</td>
<td>NB</td>
<td>Number</td>
<td>27</td>
<td>UH</td>
<td>Interjection</td>
</tr>
<tr>
<td>9</td>
<td>NG</td>
<td>Negation Particle</td>
<td>28</td>
<td>VBD</td>
<td>Verb - Past Tense</td>
</tr>
<tr>
<td>10</td>
<td>NNF</td>
<td>Singular Common Feminine Noun</td>
<td>29</td>
<td>VBDC</td>
<td>Past Cupola</td>
</tr>
<tr>
<td>11</td>
<td>NNM</td>
<td>Singular Common Masculine Noun</td>
<td>30</td>
<td>VBDX</td>
<td>Past Auxiliary Verb</td>
</tr>
<tr>
<td>12</td>
<td>NNP</td>
<td>Proper Noun (Named Entity)</td>
<td>31</td>
<td>VBG</td>
<td>Gerund</td>
</tr>
<tr>
<td>13</td>
<td>NNS</td>
<td>Plural Common Noun</td>
<td>32</td>
<td>VBH</td>
<td>Subjunctive Verb</td>
</tr>
<tr>
<td>14</td>
<td>IN</td>
<td>Preposition</td>
<td>33</td>
<td>VBIMP</td>
<td>Imperative Verb</td>
</tr>
<tr>
<td>15</td>
<td>PRC</td>
<td>Clitic Pronoun</td>
<td>34</td>
<td>VBINF</td>
<td>Infinitive Verb</td>
</tr>
<tr>
<td>16</td>
<td>PRDEM</td>
<td>Demonstrative Pronoun</td>
<td>35</td>
<td>VBN</td>
<td>Past Participle</td>
</tr>
<tr>
<td>17</td>
<td>PRDIS</td>
<td>Distributive Pronoun</td>
<td>36</td>
<td>VBP</td>
<td>Verb - Present Tense</td>
</tr>
<tr>
<td>18</td>
<td>PRP$</td>
<td>Possessive Pronoun</td>
<td>37</td>
<td>VBPC</td>
<td>Present Cupola</td>
</tr>
<tr>
<td>19</td>
<td>PRPi</td>
<td>Personal Pronoun – 1st Person</td>
<td>38</td>
<td>VBPX</td>
<td>Present Auxiliary Verb</td>
</tr>
</tbody>
</table>

### 6. Methodology

Corpus annotation is an ongoing process, however, at the time of writing this document, around 2M words are assigned with POS tags. For the tagging process, an iterative approach was adopted where the process was carried out in several rounds. In each Round a chunk of 10K sentences (around 230K words) was tagged and added to the corpus. Each Round has two sub-phases, automatic tag assignment, and manual correction. The first Round is different from the subsequent rounds in the sense that the automatic POS tagging phase in the first Round is lexicon-based while in the subsequent rounds it is
Machine Learning (ML) based. In lexicon-based annotation, the words are tagged from the lexicon, regardless of the context. The first chunk was tagged using this approach and then manually corrected. To speed-up the tagging process and reduce the burden on human taggers, a ML approach was then incorporated. The purpose of ML was to exploit the information added by human taggers to the corpus in the form of manual correction. Using the labeled data obtained from Round 1, a ML model was trained using the Conditional Random Fields (CRF) algorithm. The model captures the morphological features and the neighboring information of the target token and predict its label/tag. The error rate of the automatic POS assignment phase was reduced by replacing the baseline lexicon-based technique with the ML model. With each Round of tagging, the amount of training data was increased and accordingly the accuracy of the model was also increased, resulted in a quicker manual correction and high-quality corpus. The flowchart diagram of the methodology is given in Fig. 5.

7. Word Segmentation

Word Segmentation or tokenization is the process of chunking/splitting a text into words. It is considered to be the first step in many NLP applications. Word segmentation of the Pashto text is not intuitive and it's still an open area of research. The proposed corpus can be used for the development of a specialized word segmenter, though this study itself needs segmented text. This project dependent on the two baseline segmentation techniques that are commonly used for other languages, i.e., Whitespace segmentation and Lexicon-based approach. Besides these two techniques, several steps of noise removal and manual correction were also carried out as shown in the Flowchart diagram in Fig. 5. Whitespace tokenizer is an intuitive technique in which a whitespace is considered as the word boundary and each space-delimited token is treated as a word. This technique works well in Latin-based languages, where whitespace is used as explicit word boundary, but not useful for Asian languages like Chinese, Japanese and languages written in Arabic script such as Urdu and Pashto. Though we used it in preprocessing, a whitespace tokenizer alone is not enough for Pashto word segmentation and its output was then given to a Lexicon-based word segmenter. Lexicon-based segmentation is one of the simplest techniques used to convert a text into words, a description can be found in (Rashid and Latif 2012, Long and Boonjing 2018). In this technique the input text is scanned character by character from the beginning and looked-up for sequences of characters in the lexicon. If the sequence of characters found in the lexicon, a word-breaker is inserted after the sequence. To ensure matching the longest length the Longest Matching (LM) variant is used. LM is a greedy algorithm that try to find the longest matches first and then gradually decrease the window size, and if the sequence is not found the last character left is considered as a word. But we used a slightly modified version of the LM, where instead of character sequences we looked up for sequences of tokens obtained from whitespace tokenizer. Tokens that were not found in the lexicon were considered as OOV (Out of Vocabulary). Sentences having no OOV were labeled for word-boundaries, where the whitespaces that were also word-boundaries marked with “B” and the remaining whitespaces that were part of the compound words marked with “S”. Sentences having OOV undergone another phase of cleaning. It was a kind of manual proofing where the OOV tokens were extracted and manually analyzed. The OOV tokens that were valid Pashto words but not present in the lexicon were simply added.
to the lexicon. Tokens that were just random strings, the whole sentence was discarded. The third category of OOV was the most interesting one, where the tokens were valid Pashto words but because of noise i.e., space-omission and space-insertion errors, the lexicon-based tagger didn’t recognize them. Some examples of these noisy tokens are, , , , , , , , , and etc. These noisy tokens were too abundant that discarding them would significantly affect the size of the corpus and manual correction of each token was also not feasible. One solution was to use a semi-automatic technique where a dictionary of the most frequent noisy tokens was created with their corresponding correct substitutes. A find-&-replace operation was then performed on the whole corpus where each noisy OOV token was replaced with its correct substitute according to the dictionary. This operation was performed several times, with updated Errors Dictionary each time, and a total of nearly 200K corrections were made. This technique removed the space-omission errors quite well but lifted some space-insertion errors, the remaining errors were corrected in-parallel with the correction of POS tags. Figure 6 is a snapshot of the corpus after word segmentation and manual correction and Fig. 7 is the length of words with their frequencies of repetition in the lexicon as well as in the corpus.

8. Pos Tagging

The process of POS tagging was carried out in several rounds, where each Round was completed in two sub phases, (i) The Automatic POS assignment, and (ii) Manual Correction. The automatic POS assignment in Round 1 is Lexicon-based where in the following rounds it is ML-based – discussed in the later sections.

8.1 AUTOMATIC POS ASSIGNMENT (LEXICON-BASED)

In lexicon-based method, words of the sentence are looked-up one by one in a lexicon/dictionary and if found, the corresponding tag is assigned. In the first Round of POS tagging, we used this approach for automatic POS assignment. Generally, the larger the size of the lexicon the greater are the chances that a word will be assigned with a POS tag. But in Round 1, we selected only those sentences from the corpus where all of the words were present in the lexicon, thus each of the word was assigned with a tag from the lexicon. This is a context-free approach, in which the surrounding information of the word did not taken into consideration, but in real world, words change their meaning according to the context. In Fig. 8 two sentences are given as an example where the tags are assigned form the lexicon. In both sentences the word “” has assigned the Adjective (JJ) tag, though in the first sentence it means < dam, reservoir > which is a Common Singular Masculine Noun (NNM). After the Lexicon-based annotation, the whole corpus undergone another phase of annotation that is the manual correction by human annotators.

8.2 MANUAL CORRECTION

For manual correction, a specialized web application is developed that is available for human taggers everywhere and anytime. The application is secured by an authentication system and can only be accessed by the authorized individuals. Sentences from the database are presented one by one to the annotator where each word has the most probable tag already assigned to it (during the automatic phase
mentioned earlier). A snapshot of front-end interface of the annotation module is given in Fig. 9. Some tags in the figure look different from our tagset but these are just converted to a more readable format according to our annotators’ understanding, on the back-end these are converted back to the format of tagset. For example, the tag for “Common Plural Noun” is “NNS” in the tagset and displayed as “NN.C.2”, this format may look more descriptive, but we tried to stick to the convention followed in previous corpora. The application has also the functionality of tracking the number of changes made to the tags, which are useful for evaluation of the automatic POS tagging phase. Some basic features of the application are:

- APPROVED – if the annotator agrees with the tagging of automatic phase, the sentence is just marked as approved
- SAVE & NEXT – this option is used to save the changes made to the tag.
- WHITESPACE REMOVAL – this option is actually related to word segmentation, used to remove space-insertion errors from the sentence if any.
- REMOVE – if the sentence has some obvious mistakes i.e., have space-omission errors or incorrect words, the sentence is removed from the corpus.

The application is designed in such a way that one sentence is presented to one tagger only once, and a sentence can be retrieved from the database two times only. Thus, each sentence was checked twice by two different taggers. The taggers made corrections on average speed of 50 seconds per sentence. For five annotators, the correction of 10K sentences took around 20 days by working three hours daily. Figure 10 is snapshot of the corpus after manual correction. From the results recorded by the application, we analyzed that out of the total 230K words checked, the tags of 11.5K words were changed, which makes around 5% of the total. It means that around 95% of the words did not change their meaning in the sentence and the tags assigned from the lexicon were correct. In other words, the lexicon-based POS tagger tagged with 5% error rate. Our aim hover is to reduce this error rate to speed-up the annotation process and reduce human effort, in order to do that we incorporated ML-based approach in the subsequent rounds.

9. Pos Tagging Using Machine Learning

In the first Round of POS tagging around 230K words were tagged to a 100% (theoretical) accuracy. But due to the context-free approach, the lexicon-based tagger resulted in a high error rate and led to a laborious manual correction phase that took a very long time – despite using a handy tool. To reduce the error rate in the automatic phase of tagging, reduce human effort and speed up the annotation process without compromise on the quality, we incorporated a supervised machine learning technique. For many NLP tasks, systems that use supervised learning better generalize over unseen data than rule-based systems. In supervised ML, an algorithm is fed with the training data, which instruct the system what output is related to each specific input value, called training. The trained model is then presented with data that the model has never seen; the aim is to measure how accurately the model will predict the
output. Supervised learning technique is very useful in situations where sufficient amount of data is available for training the model. In our case, we already have labeled/tagged a reasonable number of words in the first Round that is enough to train the model. Besides data preparation, the selection of appropriate algorithm for training the model is also very crucial. Natural language exists in sequential form and assigning POS tags to a sequence is a sequence labeling task where a label (tag) is assigned to each element (word) in the sequence. A classification decision needs to be made at every position in the sequence and classification decisions at different positions influence each other. For sequence labeling tasks several algorithms are available but the one that is commonly used and recommended by many researchers is the Conditional Random Fields.

### 9.1 CONDITIONAL RANDOM FIELDS

Proposed by Lafferty et. al. in 2001 (Lafferty, McCallum et al. 2001), CRF is probabilistic classifier that compute probability distribution over the possible label sequence and model the conditional probability of the entire sequence. This model assumes that features are dependent on each other, and also considers future observations while learning a pattern in the sequence. In POS tagging, context information such as neighboring words and their morphological features are taken into account that significantly improves the performance. CRF allow us to integrate such context features easily, and several studies show that the CRF tend to outperform other methods, as discussed in (Tran, Le et al. 2009, Zhang, Huang et al. 2009, AlKhwiter and Al-Twairesh 2021). A special case of CRF, known as linear-chain CRF has been widely used for sequence labeling tasks which takes the following form:

$$P(y|x) = \frac{1}{Z(x)} \exp \left( \sum_t \sum_k \lambda_k \cdot f_k(t, y_{t-1}, y_t, x) \right)$$

where $x = (x_1, \ldots, x_n)$ is the observation sequence, $y = (y_1, \ldots, y_n)$ is the label sequences $f_k$ and $\lambda_k$ are the feature functions and their corresponding weights respectively, and $Z(x)$ is the normalization factor. Normalization is performed since the output is a probability and the weight estimation is performed by maximum likelihood estimation using the feature functions we define, $Z(x)$ can be expressed as:

$$Z(x) = \sum_{y \in \mathcal{Y}} \exp \left( \sum_t \sum_k \lambda_k \cdot f_k(t, y_{t-1}, y_t, x) \right)$$

### 9.2 FEATURES SELECTION

The POS tagger is essentially a CRF sequence classifier. In CRF, the input is a set of features (also called state features) derived from the input sequence using feature functions. Some examples of feature functions are: what are the suffix and prefix of the word, is it a number, what is the previous and the next word etc. This way the data is converted to a feature's dataset and CRF then try to determine the weights of different feature functions that maximize the likelihood of the labels in the training data. The contexts
that are used to predict the POS tag in Pashto are roughly similar to that used for English. These are the surrounding words and word components. Pashto has similar or may be richer morphology than English, words are enriched by affixes and inflected to communicate various meanings. These morphological gestures are very useful for predicting the category of the word. For training the POS tagger, we considered the following features:

- The word
- If the word is a digit
- If the word is a punctuation mark
- If the word is at the beginning of the sentence or the end of the sentence or neither
- Length of the word – number of characters in the word, since shorter words are more likely to be Particles or Pronouns and the words that are very long are likely to be Proper Nouns.
- Prefixes and suffixes of the word up to three characters long
- Features mentioned above for the surrounding words up to two places before and after the word

### 9.3 TRAINING THE MODEL

For implementation and training the model we used Python `sklearn-crfsuite`\(^1\) wrapper. The c1 and c2 parameters that are the coefficients of L1 and L2 regularization respectively were tweaked for better results and both were finally set at 0.1. Gradient descent using the L-BFGS method was selected as the training algorithm. The dataset was divided on a ratio of 80 to 20 for training and testing respectively. The device used for the experiment was a PC with Intel Core i5 processor and 16GB memory. And training of the model took around 350 seconds.

### 9.4 RESULTS AND EVALUATION

The model yields an accuracy of 98% and F1-measure of 93% on the test data. Though these figures are not very useful indicators to measure the actual improvement we achieved from the Round 1. The real measure is the one obtained from the manual correction phase. In Round 2, ML based POS tagger replaced the lexicon-based tagger and another chunk of 10K sentences (around 230K words) was annotated. After manual correction of the tags, the results recorded by the application were analyzed again. This time the tags of 6.5K words were changed that is 2.8% of the total. It means, the ML-based tagger assigned the POS tags with 97.2% accuracy – an increase of 2.2% from the Round 1 (lexicon-based tagger). Though the size of training data was not very large but we still achieved some improvement. The tagging procedure of Round 2 was repeated again and again and all of the remaining corpus was annotated in this fashion. In each Round the amount of training data was increased by and amount 230K words and consequently the accuracy of the model was increased as well. The Graph in Fig. 11 shows the increase in Test Accuracy as well as the Accuracy measure obtained from the manual correction phase with the increasing size of dataset.

\(^1\) [https://sklearn-crfsuite.readthedocs.io/en/latest/](https://sklearn-crfsuite.readthedocs.io/en/latest/)
10. The Final Pos Tagger Model

One objective of this study was to create a general-purpose POS tagger for Pashto that can be used as an off-the-shelf tool in future research. Therefore, after tagging all the corpus and manual correction, a CRF model was trained once again, keeping the parameters similar to that discussed earlier. Though the dataset used this time was all the POS-tagged corpus of 2M words. 80% of the dataset was used for training and 20% for testing. On test dataset, the accuracy of the model was 99% with the F1-score of 98%. The results of the model are quite satisfactory and we hope that it will be useful in practice. Figure 13 is the confusion metrix obtained from the POS tagger on test data.

11. Use-cases Of The Pashto Corpus

Research in Pashto NLP is still in infancy, and apart from POS tagging there are countless open research areas where the corpus can be used. However, some of the very basic use-cases are discussed here.

11.1 PROOFING TOOL

Pashto is not a standardized language and there are no rules to impose the proper use of whitespace in writing system. Though whitespace is used for words separation but not consistently and thus it cannot be treated as an explicit word-boundary like English or other western languages. On the other hand, some languages like Chinese and Japanese ignore a whitespace and it is a not considered as a part of writing. But, in Pashto a whitespace cannot be ignored as well, because it is a part of writing – especially typing with the modern keyboards. The improper and inconsistent use of whitespace in Pashto leads to two typical spelling errors, commonly known as space-omission and space-insertion errors. A brief explanation with examples is already given in Section 2. These two errors have made the task of word segmentation very challenging and unique that cannot be performed only by using the available baseline segmentation techniques, such as whitespace tokenization or Lexicon-based approach. Ignoring these errors will negatively affect the performance of any NLP algorithm. The proposed corpus can be used to solve the issue of space-omission and space-insertion errors. An AI model can be trained on character-level information that will capture the context information of each character to predict the proper position of whitespace in the sentence.

11.2 WORD SEGMENTER

The whitespace tokenizer may be sufficient for many NLP applications, but, some scenarios demand for a word segmenter. For example, a machine translation system may need to capture the whole word e.g., “New York” not “New” and “York”. The quality of word segmentation affects the performance of all downstream tasks, such as NER, homographs disambiguation and Machine Translation. One such example is a recent study on sentiment analysis in Pashto, where the classic ML algorithms like Random Forest and Naïve Bayes have outperformed sophisticated Deep Learning (DL) algorithms, RNN and LSTM (Iqbal, Khan et al. 2022). One of the reasons is that these DL models are based on word embeddings i.e., Word2Vec and GloVe, where most of the available word embeddings are trained on noisy content, and
therefore fail to efficiently capture the syntax information. The capability of whitespace tokenizer is limited to simple-words, and cannot capture compound words. To efficiently overcome the limitations of whitespace tokenizer and capture the compound words, one traditional solution is the Lexicon-based Longest Matching approach. But this approach has some limitations as well, one is the so-called OOV errors, and secondly, it’s hard to find publicly available digital lexicon of Pashto language. Using the proposed corpus, a specialized AI model can be trained for Pashto word segmentation. Keeping in view this use-case, the corpus is labeled for word-boundaries information where simple whitespaces and word boundaries are explicitly marked with “S” and “B” tags respectively, as shown in Fig. 6.

11.3 NER

NER is the process of identifying and classifying named entities (NEs) in text. NEs are words or phrases that refer to specific entities, such as locations, people, organizations and so on. NER is an important task in NLP as it can help in many downstream applications such as information extraction, machine translation and question answering. NER is still an open research area in Pashto language, and a rich corpus is the prerequisite for an efficient NER system. From NER perspective a rich corpus means a corpus with many NEs, and NEs are usually the Proper Nouns. In the proposed corpus, around 65% of the sentences have Proper Nouns that are assigned with NNP tags, this portion of the corpus can easily be extracted. For NER system, the dataset is first labeled with POS tags e.g., (Sang and De Meulder 2003), these tags can then be used as features for prediction of NEs. In our corpus, 2M words are already assigned with POS tags, and besides that we have developed the general-purpose automatic POS Tagger that can be used to tag more words with state-of-the-art accuracy.

11.4 HOMOGRAPH DISAMBIGUATION

Homographs are words that have the same spelling but different meanings and sometimes different pronunciations. For example, the word “black” can be Adjective referring to “black” or Common Singular Masculine Noun (NNM), means “blame”, similarly the word “sheep” can be Adjective means “mixed” or Common Singular Feminine Noun (NNF), means “sheep”. Homograph disambiguation is the process of determining the correct meaning of a homograph in a given context. Homograph disambiguation is an important task in NLP, as it helps to improve the accuracy of language models. However, it is a challenging, and POS tags can all play an important role in determining the correct meaning of a homograph.

11.5 CONSTITUENCY AND DEPENDENCY PARSING

Constituency Parsing is the process of analyzing a sentence by breaking it down into sub-phrases (known as constituents). These sub-phrases belong to a specific grammatical category, like VP (verb phrase) and NP (noun phrase). Output of the constituency parser is typically a parse tree that represents the hierarchical structure of the sentence. The parser uses a set of grammatical rules and a grammar model to analyze the sentence and construct a parse tree. Dependency Parsing is a technique used to identify the syntactic structure of a sentence by analyzing the relationships between words. It focuses on
identifying the grammatical relationships between words i.e., subject, object and verb of the sentence. The output of dependency parser is typically a dependency tree/graph representing the relationships between words. Dependency parsing focuses on the linear structure of the sentence while constituency parsing focuses on the hierarchical structure. Both techniques have their own advantages and often used together to better understand the sentence. POS tagging is essential for both constituency and dependency parsing as it provides important information about the grammatical function of a word in a sentence e.g., whether it is a noun, verb, or adjective, and this information is used to determine its relationship with other words in the sentence. For example, consider the sentence "the cat ate the milk," to parse this, the parser needs to know that "the" is a noun, "cat" is a noun, and "ate" is a verb. With this information, the parser can identify that "the" is the subject of the sentence and "cat" is the object of the verb "ate." The proposed corpus can thus provide a foundation for further types of parsing.

12. Conclusion

In this study we developed the first large-scale corpus of Pashto language, consists of 5M words. The corpus is labeled for explicit word-boundaries and at the time of writing this paper, 2M of the words are tagged for Part-of-speech information. An iterative approach was adapted for corpus annotation, where the first Round was tagged using lexicon-based approach while the following rounds were ML based. Each Round was followed by manual correction for that we developed a specialized web application to ensure quality. The purpose of incorporating ML was to speed-up the annotation process and reduce the manual effort. Based on the corpus, we trained a CRF based ML model for automatic POS tagging. The results of the POS tagger model are very satisfactory and we believe that it can be used in practice. But the corpus is created on general-purpose guidelines and its use-cases are not limited to this study only. Apart from POS tagging, the corpus is aimed to be used in various NLP application and the most basic use-cases are briefly explained in this study. Besides the mentioned use-cases, there are countless open research areas in Pashto NLP where the corpus can be helpful such as identifying private information, censorship and disinfection, threat intelligence, clickbait detection, text forensics, plagiarism detection, copyrights protection and much more.

13. Future Work

The development of this corpus is a sub-part of our ongoing project – the Pashto Treebank (PTB). Similar to the English counterpart Penn Treebank, the objective of PTB project is to create a large-scale annotated parsed treebank that can serve as a gold standard and single point of entry for further research in Pashto NLP. This study is our first milestone that is the foundation for further forms of analysis i.e., syntactic parsing and semantic field annotation. The main focus of this study was on data collection – not too much on algorithms, and all the experiments were based on the CRF algorithm only. Having this corpus in hand, sophisticated algorithms will be practiced in future to get more improved results.
Declarations

Funding

No funding was received for conducting this study.

Conflict of Interest

The authors have no competing interests to declare that are relevant to the content of this article.

AUTHORS CONTRIBUTION

The 1ST author wrote the main manuscript. The 2nd and 3rd authors supervised the project and 4th author provide technical assistance and helped in experiments. All the authors reviewed the manuscript.

References


Figures
Figure 1

Pashto alphabets/letters

<table>
<thead>
<tr>
<th>Pashto Alphabet</th>
<th>English Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>او خوره (ate)</td>
<td>V</td>
</tr>
<tr>
<td>پوئی (meal)</td>
<td>O</td>
</tr>
<tr>
<td>هغه (he)</td>
<td>S</td>
</tr>
</tbody>
</table>

Figure 2

Order of Subject, Verb and Object in Pashto sentence

> Afghanistan won the first match also by ten wickets
> well-written sentence

> Afghanistan لومری لوبه هم ېه لسو وبکتو ګټلي وه
> noisy sentence

Figure 3

Example of a well-written and a noisy sentence
Figure 4

Length of sentences in the corpus and their frequency

Figure 5

Flowchart diagram of the word segmentation and tagging process
Figure 6

Snapshot of the corpus after noise removal and word segmentation

Figure 7

Length of words in the Lexicon as well as in the corpus
Figure 8

Example sentence after lexicon-based POS assignment

Figure 9

Front-end of the specialized web-application developed for manual correction of the POS tags

Figure 10

Snapshot of the corpus after manual correction
Figure 11

Improvement in accuracy after each Round of POS Tagging

Figure 12

Frequency of tags in the corpus
**Figure 13**

Confusion Matrix of the final POS Tagger Model