The Dhaka University Journal of Linguistics : Vol. 12, No. 23-24, Published by the Registrar, University of Dhaka, ISSN 2075-3098

How to Process Bangla Linguistic-Data through NLP Pipeline

Mohammad Mamun Or Rashid*

Abstract: Raw texts are the most used applied form of human language, especially stored in electronic and digital media. In this research, we would like to propose a Bangla language processing with annotation guidelines from raw data to value-added data. The main objective of this paper is defining all input-output specification which is pipelined as processing phases. The major processing phases are set on the text understanding part. Test understanding part requires various types of tagging and labeling tasks like PoS tagging, parsing, NER tagging, etc. Moreover, another two steps need to be achieved to comply with the pipeline, that is coreference resolution and word sense disambiguation which define the semantic states of any linguistic input.

Keywords: Corpus Linguistics, Bangla Text Processing, Annotation

1. Introduction

Language processing is generally used for language-understanding and also for developing data-driven real-life applications. It follows some steps as a part of a non-linear pipeline. For English, it has been achieved a well-defined benchmark whether Bangla is still undefined as it is considered 'low-resource language'. Not having enough linguistic data is the initial reason for the low-resource tag. There is no processing standard to analyze the Bangla language scientifically. Both data and processing knowledge are parallelly responsible for backwardness.

2. Previous Literature

Over the last decade, several research papers on various phases of Bangla language processing have been published. These papers have

* Assistant Professor (on lean), Dept. of Bangla, Jahangirnagar University & Consultant, Bangladesh Computer Council.

not been deeply concerned with the processing pipeline, they were centered on a single NLP problem. Though language processing is not only a single phase, it is rather a holistic pipeline. However, the concept of the pipeline has been given importance in the documentation of various NLP libraries and tools in other high resource languages other than Bangla. Because text processing is not possible through any fragmented or isolated phase, it's a part of a pipeline. So, this processing has to go through a continuous process. Since text collection, annotation and distribution have been selected in this study as a part of the processing, the researches related to processing are the focus of today's discussion.

For corpus development, Martin Wynne's edited book (Wynne, 2005) on a corpus, in particular, is worth following. His theoretical discussions are very constructive to ensure the quality of the corpus. A formation of a pipeline was found in Jurafsky and Martin's book (Jurafsky & Martin, 2009) as it is widely accepted for its initial and necessary discussion of understanding the natural language processing. The book covered data cleaning to semantic annotation phases including word embeddings type understanding. However, some processing tools like Stanford OpenNLP and NLTK exercise the idea, moreover implement the pipeline. So, guidelines provided by these libraries should be accepted as standard.

After the development of the corpus, the principal concern is annotation. For annotation Pustejovsky and Stubbs's book (Pustejovsky and Stubbs, 2012) is regarded as the Bible of Annotations. Here, all complexities regarding annotation are discussed in a systemic theoretical presentation. However, the applied aspects of the annotation projected in the documentation (Language Processing Pipelines: n.d.) of Spacy and Prodigy (Annotation interfaces: n.d.). Their correlated guidelines are hands-on and endorse applied solutions. At the same time, their theoretical part is also quite rich. Documentation of other tools like Doccano (Text Annotation for Human: n.d.) etc. also gives an idea about practical processing. Among the discussions on various processing phases for the Bangla language, there are many data-driven discussions. Most of the researcher talked about POS tagging (Sarkar & Basu, 2007; Parts of Speech Tagset: Bengali, 2009; Ekbal, Hasanuzzaman & Bandyopadhyay, 2009; Bali & Choudhury, 2010; Dash, 2013; Paul & Mumin, 2014). In terms of PoS tagging, there

are few discussions on lemmatization and named entity (Chakrabarty, Chaturvedi, & Garain, 2016; Ekbal & Bandyopadhyay, 2009; Banerjee, Naskar & Bandyopadhyay, 2014). Although there are few discussions on the structure of Bangla sentences (Dhar, Chatterji, Sarkar & Basu. 2012), it is easy to say that discussion on a phrase-based parser or dependency parser with computational aspects is rare. Even though, there are very few theoretical discussions about coreference resolution and word sense disambiguation which represent Bangla structure. Nonetheless, most of the aforementioned researches did not build the gold standard dataset, even did not apply annotated big datasets. Therefore, we are proposed to build a data-driven processing pipeline to develop a gold standard dataset and thus, understand the Bangla language scientifically.

In terms of wholistic-processing, there are some well-accepted state-of-the-art pipeline and libraries like NLTK (Loper and Bird, 2002), Stanford CoreNLP (Manning et al., 2014), Spacy (Honnibal and Montani, 2017), AllenNLP (Gardner et al., 2018), Flair (Akbik et al., 2019), Stanza (Qi et al., 2020) and Huggingface Transformers (Wolf et al., 2019). Among them, Stanza uses a neural pipeline for text engineering, including tokenization, multi-word token expansion, lemmatization, part-of-speech and morphological feature tagging, dependency parsing, and named entity recognition. Most of them are following a multi-step pipeline to process textual data and based on probabilistic models, moreover, in most of the cases, they are flawless, efficient and their performance is acceptable. Even, some libraries like Indic NLP library, iNLTK, etc are showing good results for many languages even for 'low-resource' languages. Indic NLP library (Kunchukuttan, 2020) uses some steps as a part of feature engineering tasks such as Text Normalization, Script Information, Tokenization, Word Segmentation, Script Conversion, Romanization, Indicization, Transliteration, Translation for the pipeline. The iNLKT (Arora, 2020) uses tokenization, NER, POS, and Text entailment though it emphasizes embedding vectors and language models.

3. Why Processing

In this age of unsupervised machine learning, a primary question is usually raised that why corpus needs to be processed. There are many possible answers, the first one could be, raw text may contain tags of

many types of markup language, may contain information about the page, sometimes contain unnecessary words from other scripts. All these elements have to be removed or processed as noise. Secondly, a language has to be analyzed to understand it. As required for analysis, the data has to be processed according to the goal. The third purpose is dataset preparation to build machine learning models. To develop models, it is well established that various feature engineering tasks have to be done. As a result, to do this processing work, one needs to have a deep grasp of the language, as well as skills in various disciplines like descriptive linguistics, corpus linguistics, etc. Therefore, in a general sense, there are two short answers, that why textual data is processed: firstly. To understand the language, secondly, to solve machine learning-based problem.

3.1 To understand the nature of natural language

One of the main purposes of linguistic data processing is to understand the complexities of language in a scientific manner. As an example, only big data processing could give an idea of the number and type of verbs in a specific language. Or it can be easily said through processing what is the ratio of the content word to functional word in its corpus. Or, someone wants to know, how many words in different languages in the corpus of a language, or what are the most less occurred or high-frequency words in the language. For these aforementioned cases, data processing is required for every step.

Traditional grammar usually expresses the rules of a certain language from references to previous literature or the personal intuition of the scholars. Language processing works in a reverse way. In this process, the rules of language are extracted through corpus analysis. Traditional grammar, which has been practiced for many years, is the ancestor of this domain. But human languages are so diverse that, the grammar or rule that is found after big data analysis by machine is very different from traditional grammar. The gaps created by human scholars could be filled up through big linguistic data analysis using scientific approaches. Traditional grammar, even descriptive linguistics, has not discussed some of the linguistic problems or issues that can be solved through corpus linguistics. Descriptive linguistics deals with some aspects of text analytics and deals with constituency parsing and word sense disambiguation. However, corpus linguistics analyzes every step

of the language or textual data. At the same time, all the needs of lexicography can be met through corpus linguistics.

As a result, language has to be processed to analyze it with the help of technology. Some processes can be deduced directly through quantitative analysis. Moreover, some analysis demands human intellectual and laborious input. Some preliminary data and answers are obtained by direct quantitative analysis of the text of the corpus. e.g., character frequency in the corpus; Corpus token and type number, type-token ratio; most frequent and less occurred tokens; total sentence number, sentence number according to word count, longest and shortest sentence number, etc.

Some more quantitative analysis can be done if there is metadata. For example, how many sources used, how many topics or domains are there, how much data is there at any given time according to the time of creation, what is the rate of copyright-free content? What is the ratio of oral data to transcribed corpus and return corpus etc?

Some steps can be solved with a formula like regular expressions. For example, noise removal or data cleaning, normalization, stop word removal. In most cases, morphological analysis of the language can be done by extracting the stem or lemma with the help of formulas.

The internal architecture of the language can be found through annotations. But just as annotations are extremely laborious and expensive work, computational knowledge is needed to create its guidelines. That means the annotator has to be told how the data will be analyzed. Data needs to be processed or annotated according to the needs of the data scientist. As a result, many basic language problems can be solved through annotations. i.e., Parts of Speech Recognition, STT Recognition, Shallow Parsing or chunking, Dependency Parsing, Correlation Resolution, Word Sense Disambiguation, etc.

3.2 To solve the ML problem for real-life application

What is the benefit of finding out the features of language through technology? The idea from Alan Turing in 1950, *Can Machine Think* (Turing, 1950: 433-460) will answer this question. The underlying philosophy of the question raised by Mr. Turing has created a revolution in information science, especially in the NLP world. An artificial knowledge graph owned by a machine will be capable of

handling various features of human language. This means that linguistic data and processing of that data is a prerequisite for creating language-centric artificial intelligence.

Real-life applications with artificial intelligence are commercial or popular terms, in technical documents, it is machine learning or just ML. However, this subject is revolutionary though works very simple manner. In traditional software or applications, the input data is given with formulas or instructions, the input data is given following the instructions and output. Such software or systems cannot solve any problem outside of the formula or out of the previous input. But machine learning analyzes the input and output and removes the middle rules. As a result, it learns to solve problems by extracting formulas from inputs and outputs himself, so it can handle problems outside the previous inputs. That is why input-output is very important in any AI application.

4. How to process

In NLP applications, a dataset or corpus is the only raw material for developing applications. Therefore, the process begins with the corpus collection. Though dataset collection is not the only concern, the researchers have to ensure its quality and quantity. To ensure the quality of the corpus, some parameter like balance and representation has to be justified. For quantity, various criteria including deduplication, the type-token ratio has to be checked. In addition to considering the quality and quantity of the corpus, other things need to be ensured for management and like the interoperability and sustainability of the corpus in the future. e.g., source, metadata, and storing and distribution.

No electronic corpus can represent the totality of human language, because collection and enhancement of corpus is a never-ending process. And must be constantly updated. As a result, it is a very complex and almost impossible task to develop a beneficial corpus in all its functions. Yet so much attention has been paid to both qualitative and quantitative standards when creating nationally tagged corpus that they can represent a language. And analyzing these corpus gives an almost accurate idea of the language and is also useful in real-life application development.

There is no limit or standard for how big a corpus should be in terms of quantity. However, a comparative discussion of various internationally recognized corpus collections (The most widely used online corpora, n.d.; Leipzig Corpora Collection, 2017; Mumin, Seddiqui, Iqbal & Islam, 2018; bnWaC: Bengali corpus from the web, n.d.) suggests that the amount must be enormous to get an idea of the nature of the language.

Table 1: Token amount of well-accepted Corpus

Corpus name	# words in million	time period
iWeb: The Intelligent Web-based Corpus	14000	2017-2020
News on the Web (NOW)	6040	2010-2018
Global Web-Based English (GloWbE)	1900	2012-2013
Wikipedia Corpus	1900	2014-2000
Hansard Corpus	1600	1803-2005
Early English Books Online	755	1470s-1690
Corpus of Contemporary American English (COCA)	560	1990-2017
Corpus of Historical American English (COHA)	400	1810-2009
Corpus of US Supreme Court Opinions	130	1790-2020
TIME Magazine Corpus	100	1923-2006
Corpus of American Soap Operas	100	2001-2012
British National Corpus (BYU-BNC)	100	1980-1993
Google Books: American English	155000	1500-2000
Google Books: British English	34000	1500-2000
bnWaC	11.5	-
SUMono	27	-
CIIL Bengali Corpus	03	-
SUPara	.8	-
Lipzip Bengali Community	16	2017-2020

A data repository that initially carries language features is called a General Corpus. There are also different types of the corpus. For example, Purposive Corpus, Monolingual Corpus, Parallel Corpus,

Multimedia Corpus, Learner Corpus, Diachronic Corpus, Specialized Corpus, etc. A corpus is said to be made for a specific application or purpose only. For example, erroneous Corpus, developed for spell checker software, contains a collection of spelling and misspellings. The parallel corpus compares and aligns two or more corresponding units of text with another language or form is called parallel corpus. i.e. machine translation corpus.

Corpus can be collected in a variety of ways; in the past, the corpus was being collected for the lexicographical purpose, and the sources were printed text such as newspapers, magazines, and books. However, with the advent of electronic text in the web world today, large amounts of data are collected with the help of scrapers and crawlers. Though, such types of the corpus are not diversified. However, in this process, it is possible to easily create a large number of corpus obtaining billions of tokens in one command. In many cases, newspapers' websites use structural metadata, topics, tags, and menu-links, so these data can cover the most relevant and useful domains of recent times. But to maintain quality, a corpus needs to be considered in terms of its quality:

1. When fixing the balance or representation depending on the domain or topic, it is necessary to first check the authenticity of the source. Then it is required to check each mother-child domain and their compatibility with sister domains according to the tags. Then, need to make sure that the same document is being counted in multiple domains since normally a single document can represent multiple topics.
2. Since the corpus is based on practical real-life data created by others, all aspects of copyright must be ensured for publication and further use. Closed corpus without copyright is a worthless and wrong investment.
3. Not only the variety of domains or topics but also the general corpus needs to be balanced and representative in terms of text type. Some mandatories categories are mentioned below to ensure the diversity of corpus in nature:

   Structured, non-running vs Unstructured, running text: Text corpus is usually unstructured data, but a small amount of the corpus could be tabular and structured data.

Imaginative vs Informative text: The corpus should contain imaginative text rich in rhetorical and critical thinking from literary sources. However, easily found corpus sources like web news cover informative sentences mainly. So, the ratio should be optimized as per requirement.

Formal vs Informal text: If a corpus represents a formal or classical language, it is often difficult to build a real-life application using it. So, people have to analyze the language used in daily life by collecting it from the field of actual use.

Scripted vs Non-scripted: There is a big difference between text from the written format corpus and transcribed text from the oral corpus. Although the written language corpus represents the standard, the non-scripted corpus represents the actual practical language of man.

Subjective vs Objective: There is a difference between writing from an autobiographical point of view, or writing from the writer's point of view. Both must be present.

Narration vs Conversation: Must have a significant amount of conversion data used in real life and various services. It is also worthy to note that some applications like question-answering system, virtual private assistant, chatbot are based on conversational dataset mainly.

Source: Print, web, and speech: Since corpus is a collection of electronic text, web corpus is the most popular. But there must have searchable data from a printed source. At the same time, there must be a transcribed form of oral or speech corpus.

4. The corpus dataset needs to support the GIF curve, and also should check the similarity and homogeneity of the linguistic features of another established corpus.
5. The composition and publication periods are important to consider. This is because the corpus may be used according to the requirements of the application only, it could be used to visualize the timeline of changes of a specific language also.
6. In all steps and functions of the corpus, the utf-8 and uft-16 must be supported as a common encoding standard to recognize the local languages.

7. The corpus must have a granular level metadata set that contains all the conditions mentioned earlier. However, a brief metadata format is added in appendix a.

4.1 Annotation and human-in-the-loop

Annotations are the most labyrinthine and tedious step in the data processing. Annotations can also be called labeling or tagging. In the supervised machine learning process, the input data is not used as training data. Because it is annotated in such a way that the machine desired result gets a clear idea. That is why it is processed according to the demand of the result or output. Learning is done to create end-to-end models from input-output only without unsupervised machine learning annotations. “Unlike robots in the movies, most of today’s Artificial Intelligence (AI) cannot learn by itself: it relies on intensive human feedback. Probably 90% of Machine Learning applications today are powered by Supervised Machine Learning” (Munro, 2019: para. 01). However, unsupervised type of learning uses a lot of data and a lot of computing resources also. The truth is it is not cost-effective and not possible for the small invested or no fund projects. It is well established that the un-labeled end-to-end model works very well for very simple problems after using adequate computation resources. That is why complex problems require data processing like annotation by the human to develop a gold standard dataset. Artificial intelligence systems often do not provide good accuracy because the annotation process and the process to ensure the gold standard is overlooked. Most of the time the subtle aspects of annotation are avoided in the name of optimization through hand engineering. In doing so, the prerequisite for the creation of artificial intelligence is the creation of faults in the process of human intelligence. The process of directly utilizing human intelligence in the process of machine learning is now called Human-in-the-Loop: “the process of leveraging the power of the machine and human intelligence to create machine learning-based AI models”. (Vikram Singh Bisen, 2020: para. 03). This concept is recommended by all human-machine interaction disciplines: “To change the limitations of present-day technology, machines must be engaged implicitly and indirectly in a world of humans, or: We must put Computers in the Human Interaction Loop” (Waibel and Stiefelhagen, 2009: 03). There are at least three reasons why human in the loop is

essential: "1. There is not a labeled set of data. 2. The data set evolves rapidly. 3. The data is hard to label through automated means.'' (Johnson, 2020: para. 07). Therefore, humans cannot be left out of the problem-making process. Human input has to be taken to create artificial intelligence.

4.2 Type of annotation

There are several types of annotations, such as image labeling, document classification, linguistics annotations, speech labeling, etc. But the annotation process may depend on the target task required by the dataset of developing the application like spell and grammar checker dataset or text summarizer dataset etc. Similarly, the annotation process depends on some types of natural language problems like NER, PoS tagger, etc. Therefore, in terms of application requirements, annotation or labeling could be classified into some categories. i.e. categorization or classification, sequencing labeling and sequence to sequence, entity linking or relation extraction, etc (Pustejovsky, 2012: 88-103; Ambalina, 2019; Infographic – 4 Types of Data Annotation, n.d.). The annotation for classification could be based on both binary class (yes/no) or multiple class. Another type of annotation is sequence labeling where sequential and inline elements of a sentence are labeled; and in this labeling method, the location and collocational data are saved to extract the context. However, sequence to sequence labeling is used to map two sequences especially parallel corresponding sequences. Relation labeling is another type of labeling where the relationship between two or multiple entities are extracted.

Therefore, for NLP problems, we can use four types of labeling method as per use-case. If the problem of text processing is classification or categorization, then annotation could be solved in the same manner. For example, a simple example of a binary class-based text classification problem is spam filtering. Here, it is considered whether a text of a mail is spam or not. Binary class classification uses simple labels yes or not, as two classes. In that case, each label and class of each document will be unique. However, such annotations can also be called metadata-type annotations because they reveal the identity of the document or paragraph.

If the task is an analysis of product review through the comments of an e-commerce site, then it can be called a multiclass classification

problem. For example, if the review category or class number is three, it can be assumed that their labels or tags will be positive, negative, and neutral. Then each class will be categorized or classified through one or another tag. The data can be saved in different places or columns in the same file with different labels in the same line.

In a multiclass classification, each document may have a unique label, moreover, it may have a multi-label. For example, if the purpose is to extract a movie category from a movie summary dataset, then a movie will get a single label initially i.e. comedy. At the same time, the same movie can get more labels like action or drama. So, the movie will get a comedy tag with additional tag action or drama. Such representation should be called a multiclass multilevel annotation.

Sequence labeling is the act of assigning a label to each element of a sequence (in a sense, a sentence is a line of words). In-text processing, a sequence is an order where the word is an element. Sequence labeling can be done in two ways: one is token labeling, the other is spin labeling. Token labeling is where each word is considered a token and each word has added a label. POS tags in English sentences are such tags. Because each token in this language can be determined with the white space tokenizer. Span labeling is the process of marking a certain part of a sentence or one or more parts marking for labeling. For example, in named entity recognition and phrase chunking from a sentence, a span of sentences consisting of multiple words is selected for labeling. It can also be called segmentation as each sequence can be divided.

Sequence to sequence labeling or mapping usually involves two aligned sentences or entities. In this case, a sequence or sentence is used as a label i.e. sentence of the target language in machine translation; sometimes even a paragraph can be a sequence. For example, a large paragraph could be used as input for text summarization, as opposed to a short one or more sentences.

There is another type of annotation, where a link or relation of one or more words with one word in a sentence is annotated. This type of annotation is used to extract links or relations between members of a paragraph instead of a sentence. Relationships are usually extracted for syntactic and semantic dependency parsing, coreference resolution, or discourse analysis.

4.3 Structure and storing

Annotation structures for data arrangement and storage during annotations can be in several ways (Pustejovsky, 2012: 94-99) such as inline or stand-off. Labels and structural marks are stored in sentences in inline annotations. And in the stand-off process, labels and structural marks are separated on the side of the input sentence. As a result, the stand-off annotations have to be stored in the next column by marking the token or character location label in the sentence or paragraph. The advantage of storing words according to the character position in the corpus is that the text can be easily reproduced and this method can be used to save the feature of word collocation or context in the sentence. It is the most efficient way to measure the relation of each word in a corpus. Our recommendation is the hybrid method. Since the linguistic data processing pipeline is running text-based, inline visual representation is very effective for annotations. For this, inline annotations will be there initially. Nevertheless, the final format for storing data exports must be a stand-off. It is necessary to adopt a format that is interchangeable with almost all required formats, starting from tab-separated values, which can include multilevel annotations. Including all these considerations, Spacey has proposed a format with an example structure (JSON input format for training, n.d.) that is compliant in terms of annotations and storage. Therefore, we would like to adapt a format (shown in appendix b.) with little chance for the Bangla dataset processing pipeline:

5. Processing Pipeline

Bangla linguistic data processing pipeline will be started by receiving a raw text as input and after receiving the raw text (appendix C). It is assumed that the raw text will be always clean and follow the same encoding standard. However, the text may or may not be normalized. That means the raw text should be inputted after standard preprocessing tasks. However, the current pipeline will be started from tokenized phase. Therefore, the text will be tokenized under BILUO notion. Then, all lemma in the given text will be extracted and thus the main part of the processing i.e. pos tagging will be implemented to recognize the grammatical categories of text. Then, the parsing phase will be executed. Though there are several types of parsing, we select

chunking or shallow parsing for phrase-structure parsing and deep parsing as dependency parsing as consequently. In chunking, the relationship between phrases are captured. And in dependency Parsing only the relation between words are captured. Then, named entity chunking, coreference resolution and word sense disambiguation phases will be performed in same manner and pipeline. These phases will add some semantic resource to the pipeline. It is worthy to note that the pipeline may not be linier all-time, every step is standalone and user can skip and merge any step as per requirement.

5.1 Tokenization

From running text, white-space based tokenization is a basic step of NLP pipeline. Tokenization is an NLP task where shorter form as tokens are detected from longer strings of text. Generally, text can be segmented or tokenized into sentences, in the same manner, sentences can be split or tokenized into words, etc. So, using tokenizer, a paragraph would be segmented into sentences and a sentence or data can be split into words. In this current pipeline, both tokenizers will be applied, but initially, the word tokenizer is a more relevant concern. For Bangla text, question mark, exclamatory marks, and Dari as period define sentence boundary. But sometimes sentences can be presented in a list or table could be ended without any periods! However, word tokenization is a more complex situation for Bangla, it is not splitting words following white-spaces.

In our approach, we would like to include hyphenated words as a single token i.e. নৌ-ফাঁড়ি, নৃ-গোষ্ঠী, দাঁড়ি-কমা, নামে-বেনামে, টিআরইএম-২. Moreover, the system could be able to detect inter-dependent words which are not visibly joined by a hyphen i.e. ভারপ্রাপ্ত কর্মকর্তা, দক্ষিণ-পশ্চিম, ২৪ পরগনা, আবহাওয়া অধিদপ্তর, সিটি কর্পোরেশন। Additionally, tokenization as entity recognition (though a separate stand-alone process) could be a better and useful task for any ML problem.

Input: ব্যবসায়ীদের সূত্রে জানা গেছে, মিয়ানমারের পাশাপাশি এ মুহূর্তে চীন, তুরস্ক ও মিসর থেকে পেঁয়াজ আমদানি করতে চাইছেন ব্যবসায়ীরা। সড়ক পরিবহণ আইনের দাঁড়ি-কমাও পরিবর্তন হয়নি: সড়ক পরিবহন সচিব। সেলিম প্রধানের নামে-বেনামে শত কোটি টাকার সম্পদ, নিশ্চিত হয়েছে দুদক। নারীর শান্তি-নিরাপত্তা সূচকে বাংলাদেশ তলার দিকে। লালমিনরহাটে আ. লীগের দুই পক্ষে সংঘর্ষে আহত ১২।

Output: [‘১২’, ‘আইনের’, ‘আমদানি’, ‘আ.’, ‘লীগের’, ‘আহত’, ‘এ’, ‘ও’, ‘করতে’, ‘চাইছেন’, ‘চীন’, ‘জানা’, ‘গেছে’, ‘টাকার’, ‘তলার’, ‘তুরস্ক’, ‘থেকে’, ‘দাঁড়ি’, ‘কমাও’, ‘দিকে’, ‘দুই’, ‘পক্ষে’, ‘দুদক’, ‘নামে’, ‘বেনামে’, ‘নারীর’, ‘নিশ্চিত’, ‘পরিবর্তন’, ‘পাশাপাশি’, ‘পেঁয়াজ’, ‘ব্যবসায়ীদের’, ‘ব্যবসায়ীরা’, ‘বাংলাদেশ’, ‘মিসর’, ‘মিয়ানমারের’, ‘মুহূর্তে’, ‘লালমিনরহাটে’, ‘শত’, ‘কোটি’, ‘শান্তি’, ‘নিরাপত্তা’, ‘সম্পদ’, ‘সচিব’, ‘সড়ক পরিবহণ’, ‘সেলিম’, ‘প্রধানের’, ‘সূচকে’, ‘সূত্রে’, ‘সংঘর্ষে’, ‘হয়নি’, ‘হয়েছে’]

5.2 BILUO notion for MWE

White-space tokenizer has some limitations like it omits the joint, collocated, and dependent tokens. So, some conjunct tokens lose their bonding with the collocated member. Sometimes white-space tokenizer split the hyphenated words. Therefore, in this pipeline, we would like to propose the BILUO notion first to handle multi-token and multi-word-expressions (MWE) as a single unit. It will be an intelligent tokenizer that will carry the underlying behavior of the Bangla language. Here, BILOU stands for beginning, Inner, and last tokens of multi-token chunks, and ‘U’ is used for unique or single token and ‘O’ for outside of consideration.

Assigning the correct label for each token (Ratinov & Roth 2009) is essential for entity chunking. As all tokens are not single-token, so the system should have the handling capability of multi-token. The best way of multi-token labeling is using the BILUO scheme (Named Entity Recognition, n.d). It could make the text rather value-added though it contains a compound labeling system.

Input: সড়ক পরিবহণ আইনের দাঁড়ি-কমাও পরিবর্তন হয়নি: সড়ক পরিবহণ সচিব। সেলিম প্রধানের নামে-বেনামে শথ কোটি টাকার সম্পদ, নিশ্চিত হয়েছে দুদক। নারীর শান্তি-নিরাপত্তা সূচকে বাংলাদেশ তলার দিকে লালমিনরহাটে আ. লীগের দুই পক্ষে সংঘর্ষে আহত ১২।

Input: [(সড়ক,'B'), (পরিবহণ,'L'), (আইনের,'U'), (দাঁড়ি-কমাও,'U'), (পরিবর্তন,'U') (হয়নি,'U'), (:,'O'), (সড়ক,'B'), (পরিবহণ,'I'), (সচিব,'L'), (I ,'O'), (সেলিম,'B'), (প্রধানের,'L'), (নামে-বেনামে,'U'), (শত,'B'), (কোটি,'L'), (টাকার,'U'), (সম্পদ,'U'), (নিশ্চিত,'U'), (হয়েছে,'U'), (দুদক,'U'), (I ,' O'), (নারীর,'U'), (শান্তি-নিরাপত্তা,'U'), (সূচকে,'U'), (বাংলাদেশ,'U'), (তলার,'U'), (দিকে,'U'), (I ,'O'), (লালমিনরহাটে,'U'), (আ. লীগের,'U'), (দুই,'U'), (পক্ষে,'U'), (সংঘর্ষে,'U'), (আহত,'U'), (১২,'O'), (I ,'O')]

5.3 Lemmatization

Reducing vocabulary using stemmer is a common NLP task. As Bangla is an inflectional and fusional language so stemmer can reduce the dataset size but cutting the suffix or prefix from the word using a stemmer could lose the quality of a grammatical category. Real-world Bangla words are inflected, input-output texts are inflected so that reducing the suffix or prefix can hamper the grammatical relations. That is why we are avoiding the stemming process though it is helpful for morphological analysis and also for some ML problems.

But lemmatization could be rather helpful for text processing where it keeps the grammatical quality of each word. It returns the lemma as a unit, which is the root word, and doesn't cut off the inflectional features of that word, so it reduces unnecessary computational costs. Also, this step minimizes ambiguity, so that dealing with homonyms and homographs is quite easier than in the past. Therefore, we prefer lemmatization as it is an intelligent operation.

Detection of the lemma is a difficult task. The outlook of some words could be as inflected words but actually, they are not inflected. i.e. the token নামে-বেনামে looks inflected but it is not.

Input: ব্যবসায়ীদের সূত্রে জানা গেছে, মিয়ানমারের পাশাপাশি এ মূহুর্তে চীন, তুরস্ক ও মিসর থেকে পেঁয়াজ আমদানি করতে চাইছেন ব্যবসায়ীরা। সড়ক পরিবহণ আইনের দাঁড়ি-কমাও পরিবর্তন হয়নি: সড়ক পরিবহন সচিব। সেলিম প্রধানের নামে-বেনামে শত কোটি টাকার সম্পদ, নিশ্চিত হয়েছে দুদক। নারীর শান্তি-নিরাপত্তা সূচকে বাংলাদেশ তলার দিকে। লালমিনরহাটে আ. লীগের দুই পক্ষে সংঘর্ষে আহত ১২।

Output: ['১২', 'আইনের', 'আমদানি', 'আ. লীগের', 'আহত', 'এ', 'ও', 'করতে', 'চাওয়া', 'চীন', 'জানা', 'টাকা', 'তলা', 'তুরস্ক', 'থেকে', 'দাঁড়ি-কমা', 'দিক', 'দুই পক্ষ', 'দুদক', 'নামে-বেনামে', 'নারী', 'নিশ্চিত', 'পরিবর্তন', 'পাশাপাশি', 'পেঁয়াজ', 'ব্যবসায়ী', 'বাংলাদেশ', 'মিসর', 'মিয়ানমার', 'মুহুর্ত', 'লালমিনরহাটে', 'শত কোটি', 'শান্তি-নিরাপত্তা', 'সম্পদ', 'সচিব', 'সড়ক পরিবহণ', 'সেলিম প্রধানের', 'সূচক', 'সূত্র', 'সংঘর্ষ', 'হওয়া']

5.4 Parts of Speech Tagging

PoS tagging is a process where the grammatical category of words from a sentence is assigned by a label or tag. PoS tagging is the most important feature engineering phase for real-life text-based

applications. It gives the machine logic how the elements are categorized in a sequence. Using pos tag the machine creates an artificial mapping of any grammar of a certain language. It is worthy to note that, pos tagging is highly tag-set dependent, which means, the number and the format (i.e. flat or hierarchical; with or without attribute) can control both annotation and trained model performance. Rich tag-set means granular label representation of language, but it means more classes or categories for the model as well as a long learning curve for the annotator too. Therefore, the universal tag-set (Universal POS tags, n.d.) is more acceptable than the Penn tag-set (Penn Treebank, n.d.). However, here we would like to propose a sample representation of Bangla pos tagging based on Penn POS tags.

Input: রহিম করিমকে বাজার থেকে ভালো মানের চাল ডাল ও তেল কিনতে বলল। সকাল থেকে সন্ধ্যা পর্যন্ত পুলিশ ও গ্রামের সচেতন লোকজন চোরদের ধরার চেষ্টা করল, কিন্তু ব্যর্থ হয়ে ফিরে আসল।

Output: [(‘রহিম, 'NNP'), (‘করিমকে’, 'NNP'), (‘বাজার’, 'NNC'), (‘থেকে’, 'PP'), (‘ভালো’, 'AJ')), (‘মানের’, 'NNC') (‘চাল’, 'NNC'), (‘ডাল’, 'NNC'), (‘ও’, 'CC'), (‘তেল’, 'NNC'), (‘কিনতে’, 'VBIF'), (‘বলল’, 'VB3') , ('। ', '.')]

[(‘সকাল’, 'NNC'), (‘থেকে’, 'PP'), (‘সন্ধ্যা’, 'NNC'), (‘পর্যন্ত’, 'PP'), (‘পুলিশ’, 'NNC'), (‘ও’, 'CC'), (‘গ্রামের’, 'NNC'), (‘সবেতন’, 'AJ'), (‘লোকজন’, 'NNC'), (‘চোরদের’, 'NNC'), (‘ধরার’, 'VBIF'), (‘চেষ্টা’, 'NNV'), (‘করল’, 'VB3'), (‘কিন্তু’, 'CC')]

5.5 Shallow Parsing/Chunking

Shallow parsing, or chunking, is the process of extracting phrases from unstructured text. Chunks is a group of relevant words or part of a sentence. There are some standard well-known chunks such as noun phrases, verb phrases, and prepositional phrases. A noun phrase is a phrase that has a noun as its head. It could also include other kinds of words, such as adjectives, ordinals, determiners. A verb phrase is a syntactic unit composed of at least one verb. In this phase, phrases of a sentence will be chunked following some steps:

Input: “রহিম করিমকে বাজার থেকে ভালো মানের চাল ডাল ও তেল কিনতে বলল।”

Output after Split as token with PoS tag: [(‘রহিম’, 'NNP'), (‘করিমকে’, 'NNP'), (‘বাজার’, 'NNC'), (‘থেকে’, 'PP'), (‘ভালো’, 'AJ'), (‘মানের’, 'NNC') (‘চাল’,

'NNC'), (‘ডাল’, 'NNC'), (‘ও’, 'CC'), (‘তেল’, 'NNC'), (‘কিনতে’, 'VBIF'), (‘বলল’, 'VB3') , ('। ', '.')

Chunking Graph as output after Chunking: (S (NP রহিম/ NNP করিমকে/NNC) (বাজার/'NNC থেকে/PP) (ভালো/AJ মানের/NNC চাল/NNC ডাল/NNC ও/CC তেল/NNC) (VP কিনতে/VBIF বলল/VV3))

Output for VP: “কিনতে বলল”

Output for NP: “রহিম করিমকে বাজার থেকে’, “ভালো মানের চাল ডাল ও তেল”

5.6 Dependency Parsing

In dependency parsing, a link is found between the elements of a sequence or a poem. Such parsing easily reveals the complex relationship of language formation. There are acceptable tag sets for this which are known as Universal Dependencies. “The general philosophy is to provide a universal inventory of categories and guidelines to facilitate consistent annotation of similar constructions across languages while allowing language-specific extensions when necessary” (Universal Dependencies, n.d.: para. 01). The main idea is any sentence of a specific language, all elements like words, have some relationship or dependency on other words in the sentence, except one. Generally, the word without dependency is considered as root, most of the time, the verb is declared as root. The rest of the dependent of a sentence are directly or indirectly related to the root. So, it’s a structure with try-part. It has a head, relation, and dependent. Usually, a dependent depends on a head according to a certain relation, and more than one dependent can depend on the same head. but, one dependent can only have one head. In this phase, the raw text will act as input and word should be recognized as head or dependent with grammatical relation as output shown below:

[পাখি সব করে রব]

Text	Dep	head text	head pos	child
পাখি	nusbj	করে	verb	সব
সব	amod	পাখি	adjective	
করে	root	করে	verb	পাখি, সব
রব	dobj	করে	verb	

5.7 Entity Chunking

Entity Chunking is a process to detect various entities from unstructured text, especially NER or named entity recognition is a major application built by using the entity chunking process. Generally, this entity chunking is based on some class which is considered a category also. There are minimum 3-class based entity recognition, even the class number could be extended to 18 as per the design of tag-set. In our approach, we would like to propose a 7-class system which is a modified version of popular and person names, organizations, locations, etc.

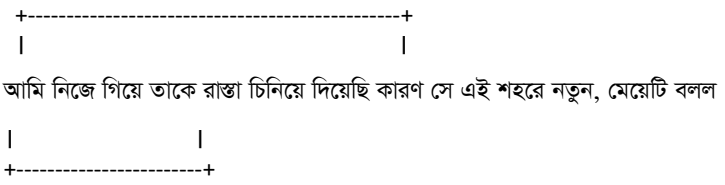
Here we use PER label for Person, groups; ORG label for organizations, companies, institutions, LOC for any types of location, place, geo-political entities like country-name, city-name and non-geo-political like river name, mountain name, etc.; OBJ and EVN label is used for expressing object and event naming. A separate label is coined for any type of title like a book title, song title by using TIT; At last, IDN is used for expressing identity name like religious and linguistical identity. Here, a sample annotated sentence with the BILUO notion has been represented:

Input: "গতকাল মঙ্গলবার রাতে ঢাকার বসুন্ধরা আন্তর্জাতিক কনভেনশন সেন্টারের বসুন্ধরা রাজদর্শন হলে মাইলসের ৪০ বছর পূর্তি উদ্‌যাপনের সমাপনী অনুষ্ঠানে মাইলসের গানে গানে মেতে ওঠেন নানা বয়সের ভক্তরা।"

Output: [(গকাল, 'B'), (মঙ্গবার, 'O'), (রাতে, 'O'), (ঢাকার, 'U-LOC'), (বসুন্ধরা, 'B-LOC'), (আন্তর্জাতিক, 'I-LOC'), (কনভেনশন, 'I-LOC'), (সেন্টারের, 'L-LOC'), (বসুন্ধরা, 'B-LOC'), (রাজদর্শন, 'I-LOC'), (হলে, 'L-LOC'), (মাইলসের, 'U-ORG'), (৪০, 'O'), (বছর, 'O'), (পূর্তি, 'O'), (উদ্‌যাপনের, 'O'), (সমাপনী, 'O'), (অনুষ্ঠানে, 'O'), (মাইলসের, 'U', 'ORG'), (গানে, 'O'), (গানে, 'O'), (মেতে, 'O'), (ওঠেন, 'O'), (নানা, 'O'), (বয়সের, 'O'), (ভক্তরা, 'O'), (। , 'O')]

5.8 Coreference Resolution

Coreference resolution is a task of referencing among relational words or phrases mainly. The process shows the intra-sentence and inter-sentence pronoun-noun relationship. There are several types of references like anaphoric reference, cataphoric reference, split antecedents, noun phrase refereeing, etc. All of these referencing extract pronouns and their corresponding reference. Here we coin a complex sentence to display the relations.



In this above-mentioned sentence, the word 'মেয়েটি' is a reference and its coreference is 'তাকে' and 'সে'; moreover, the word 'আমি' is narrator, here it's a different entity. When a word refers to an entity that is mentioned earlier is called an anaphoric reference. If the word is mentioned later, then it will be considered as a cataphoric reference. This input-output of coreference notions are illustrated as follows:

Input	Output
Anaphora: আমার ছোটভাই পাড়ার ছেলেদের সঙ্গে ক্রিকেট খেলছে। এরা ভালো কিন্তু ওর চেয়ে ওদের বয়স বেশি। এ নিয়ে আমি চিন্তায় থাকি।	[আমার: [আমার, আমি]], [ছোটভাই: [ছোটভাই, ওর]], [পাড়ার ছেলে: [পাড়ার ছেলে, এরা, ওদের]], [খলা: [খেলছে, এ]]
Anaphora: করিম গতকাল কনসার্ট দেখতে গিয়েছিল। সে বলল, এটা অসাধারণ।	[করিম: [করিম, সে]]
Catephora: তার কথা কেউ বলে না, সে প্রথম প্রেম আমার নীলাঞ্জনা।	[তার: [তার, সে, নীলাঞ্জনা]]
Split antecedents: করিম ও রহিম চাকুরি করতে ঢাকা গেল। তারা দুজনেই ভালো বেতন পাচ্ছে।	[রহিম করিম" [রহিম করিম, তারা]]
Noun phrase refereeing: দেশের সবচেয়ে মেধাবীরা বিদেশে ডিগ্রি করতে যায়। কিন্তু এই শ্রেণির লোকজন কমই ফেরে।	[দেশের সবচেয়ে মেধাবীরা: [দেশের সচেয়ে মেধাবীরা, এই শ্রেণির লোকজন]]

6. Representation of pipeline as Result

The below-mentioned (Table 02) processing pipeline has been started with receiving raw Bangla text, then it will follow some well-defined phases. It is assumed that all inputted text should be noise-fee and follow a similar encoding standard. After processing the data will be represented in both inline and stand-off format. And also, data will be exported with the aforementioned structure keeping possibilities of interoperability:

Table 2: Representation of Processing Pipline with Sample data

Pipeline phase	Annotation type	Annotation format
Raw text	NA	[পাখি সব করে রব] [রহিম সাহেব তার ছেলে করিমকে নিউ মার্কেট থেকে ১২টার মধ্যে গরম গরম তেহারি বা কাচ্চি বিরিয়ানি কিনতে বললেন। কিন্তু সে খাতুনগঞ্জ থেকে বাসি খিচুরি কিনল। তা দেখে তার মাথা গরম হয়ে গেল]
Tokenization (white Space)	Regular expression	[পাখি সব করে রব] [রহিম, সাহেব, তার, ছেলে, করিমকে, নিউ, মার্কেট, থেকে, ১২টার, মধ্যে, গরম, তেহারি, বা, কাচ্চি, বিরিয়ানি, কিনতে, বললেন, কিন্তু, সে, খাতুনগঞ্জ, থেকে, বাসি, খিচুরি, কিনল, তা, দেখে, তার, মাথা, গরম, হয়ে, গেল।]
BILUO notion for MWE	Sequence labeling, Inline, Stand-off storing	[(‘পাখি’, 'O'), (‘সব’, 'L'), (‘করে’, 'O'), (‘রব’, 'O')] [(‘রহিম’, 'B'), (‘সাহেব’, 'L'), (‘তার’, 'O'), (‘ছেলে’, 'O'), (‘করিমকে’, 'U'), (‘নিউ’, 'B'), (‘মার্কেট’, 'L'), (‘থেকে’, 'O'), (‘১২টার’, 'U'), (‘মধ্যে’, 'O'), (‘গরম’, 'O')), (‘গরম’, 'O') (‘তেহারি’, 'O') (‘বা’, 'O'), (‘কাচ্চি’, 'B'), (‘বিরিয়ানি’, 'L'), (‘কিনতে’, 'O'), (‘বললেন’, 'O'), ('I '), (‘কিন্তু’, 'O'), (‘সে’, 'O'), (‘খাতুনগঞ্জ’, 'U') (‘থেকে’, 'O'), (‘বাসি’, 'O'), (‘খিচুরি’, 'U' (‘কিনল’, 'O'), ('I '), (‘তা’, 'O'), (‘দেখে’, 'O'), ('I '), (‘তার’, 'O'), (‘মাথা’, 'O'), (‘গরম’, 'O') (‘হয়ে’, 'O'), (‘গেল’, 'O') ('I ')]
Lemmatization	Rule based	[পাখি সব করে রব] [রহিম সাহেব, তার, ছেলে, করিম, নিউ মার্কেট, থেকে, ১২টার, মধ্যে, গরম গরম, তেহারি, বা, কাচ্চি বিরিয়ানি, কিনতে, বললেন, কিন্তু, সে, খাতুনগঞ্জ, থেকে, বাসি, খিচুরি, কিনল, তা, দেখে, তার, মাথা, গরম, হয়ে, গেল।]
Parts of Speech Tagging	Sequence labeling, Inline, Stand-off storing	[(‘পাখি’, 'NNC'), (‘সব’, 'AJ'), (‘করে’, 'VB1'), (‘রব’, 'NNP') [(‘রহিম-সাহেব’, 'NNP'), (‘তার’, 'PRS'), (‘ছেলে’, 'NNP'), (‘করিমকে’, 'NNP'), (‘নিউ মার্কেট’, 'NNC'), (‘থেকে’, 'PP'), (‘১২টার’, 'NNT'), (‘মধ্যে’ 'PP'), (‘গরম-গরম’, 'AJ')), (‘তেহারি’, 'NNC'), (‘বা’, 'CC'), (‘কাচ্চি-বিরিয়ানি’, 'NNC'), (‘কিনতে’, 'VBIF'), (‘বললেন’, 'VB3'), ('I '), (‘কিন্তু’, 'CC'), (‘সে’, PR3), (‘খাতুনগঞ্জ’, 'NNC') (‘থেকে’, 'PP'), (‘কিনল’, 'VB3'), ('I '), (‘তা’, 'PRN'), (‘দেখে’, 'VBN'), (‘তার’, 'PR$'), (‘মাথা’, 'NNC') (‘গরম’, 'AJ'), (‘হয়ে’, 'VBIF'), (‘গেল’, 'VB3') ('I ')]

Pipeline phase	Annotation type	Annotation format
Shallow Parsing/ Chunking	Sequence labeling, Span, Inline, Stand-off storing, inline storing	[(S (NP পাখি/NNC সব/ AJ) (VP করে/VB1 রব/NNP)) [(S (NP রহিম-সাহেব/NNP তার/NP ছেলে/NNP করিমকে/NNP) (NP নিউমার্কেট /NNC থেকে/PP ১২টার/NNT মধ্যে/PP) (NP গরম-গরম/AJ তেহারি/NNC বা/CC কাচ্চি-বিরিয়ানি/NNC) (VP কিনতে/VBIF বললেন/VV3))। (S (NP কিন্তু/CC সে/PR3 খাতুনগঞ্জ/NNC থেকে/PP) (VP কিনল/ VB3))। (S (NP তা/PRN দেখে/VBN তার/ PR$ মাথা/NNC) (VP গরম/AJ হয়ে/VBIF গেল/ VB3)]
Dependency Parsing	Entity linking labeling, Inline, Stand-off storing, inline visualizing	[পাখি সব করে রব] Text Dep head texthead pos child পাখি nusbj করে verb সব সব amod পাখি adjective করে root করে verb পাখি, সব রব dobj করে verb
Entity Chunking	Sequence labeling, Inline, Stand-off storing	[(‘রহিম’. 'B-PER'), (‘সাহেব’, 'L-PER'), (‘তার’, 'O'), (‘ছেলে’, 'O'), (‘করিমকে’, 'U-PER'), (‘নিউ’, 'B-LOC'), (‘মার্কেট’, 'L-LOC'), (‘থেকে’, 'O'), (‘১২টার’, 'U-DNT'), (‘মধ্যে’, 'O'), (‘গরম’, 'O')), (‘গরম’, 'O') (‘তেহারি’, 'U') (‘বা’, 'O'), (‘কাচ্চি’, 'B-OBJ'), (‘বিরিয়ানি’, 'L-OBJ'), (‘কিনতে’ 'O'), (‘বললেন’, 'O') , ('। '), (‘কিন্তু’, 'O'), (‘সে’, 'O'), (‘খাতুনগঞ্জ’, 'U-GPE') (‘থেকে’, 'O'), (‘বাসি’, 'O'), (‘খিচুরি’, 'U-OBJ' (‘কিনল’, 'O'), ('। '), (‘তা’, 'O'), (‘দেখে’, 'O'), ('। '), (‘তার’, 'O'), (‘মাথা’, 'O'), (‘গরম’, 'O') (‘হয়ে’, 'O'), (‘গেল’, 'O') ('। ')]
Coreference Resolution	Entity linking labeling, Inline, Stand-off storing	[পাখি: [সব]] [রহিম সাহেব: [তার]], [করিম: [বাসি কিচুরি: [তা]]
Word Sense Disambiguation	Sequence labeling, span, Inline, Stand-off storing	[রব: [sense 1: chirping] [গরম: [sense 1: warm], গরম [sense 2: hot tempered]

7. Limitation and future researches

Each step of this study needs to be discussed in detail. In this ongoing research, only a brief idea of the processing phases and some

sentences or paragraphs are mentioned as examples to visualize the annotation format. In particular, some phases such as POS tagging, named entity recognition, dependency parsing, etc. depend on the tag-set. Each step though has a few tag-sets which have been internationally accepted. But the question is how compatible all the internationally prevalent tag-sets are with the Bangla language. The processing pipeline will not be complete unless they are verified and determined. As an extension of this ongoing research, it will be said that each of these phases will be discussed in detail, which is essentially an inevitable part of this processing.

There are some real-life constraints in the propped method. The main challenge is annotating big data keeping the quality and standard. Linguistic data annotation requires high labor from intellectually sharped humans. In terms of Bangla language, a clear understating of the language is a mandatory prerequisite to perform this task. However, finding such types of dedicated and educated people even native speakers or linguists is very rare. Therefore, deploying the right people for the annotation is the main non-technical challenge of language processing for supervised ML. Secondly, Anntatiing big data is a task that is complex in nature and enormous in scope size. So, there are always some investment issues as well as a management concern that could not be ignored.

8. Conclusion

This processing pipeline can play an effective role in the development of Bangla corpus and processing of that corpus properly. This study will serve as a general guideline, especially on how to annotate Bangla running text. And as a result of a cluster of gold standard datasets created through human annotation and validation using the human-in-the-loop concept, a powerful knowledge graph can be achieved, based on that knowledge graph one or more language models will be developed. Which will help to make artificial replicas of the Bangla language in computers and machines.

References

A. Chakrabarty, A. Chaturvedi & U. Garain (2016). A Neural Lemmatizer for Bengali. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016).

A. Ekbal, M. Hasanuzzaman, & Samir Bandyopadhyay (2009). Voted approach for part of speech tagging in Bengali. 1. 120-129.

A. Ekbal and S. Bandyopadhyay (2009). "Bengali Named Entity Recognition Using Classifier Combination," 2009 Seventh International Conference on Advances in Pattern Recognition, Kolkata, 2009, pp. 259-262, DOI: 10.1109/ICAPR. 2009.86.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Waibel, Hartwig Steusloff, Rainer Stiefelhagen, Kym Watson (2008). Computers in the human interaction loop, Springer-Verlag London.

A. M. TURING (1950). COMPUTING MACHINERY AND INTELLIGENCE, Mind, Volume LIX, Issue 236, Pages 433–460, https://doi.org/10.1093/mind/LIX.236.433.

Anoop Kunchukuttan [Web page]. (2020). The IndicNLP Library. Retrieved from https:// github.com/anoopkunchukuttan/indic_ nlp_library/blob/master/ docs/indicnlp.pdf.

"Annotation interfaces" [Web page]. (n.d.). Retrieved from https://prodi.gy/docs/ api-interfaces.

Arnab Dhar, & Sanjay Chatterji, Sudeshna Sarkar, Anupam Basu (2012). A Hybrid Dependency Parser for Bangla. 55-64.

Arun Krishna Paul & Md. Abdullah Al Mumin (2014). A Fine-Grained Tagset for Bengali Language, SUST Journal of Science and Technology, Vol. 21, No. 1,

"bnwac-bengali-corpus" [Web page]. (n.d). Retrieved from www.sketchengine.eu/ bnwac-bengali-corpus/

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Dan Jurafsky & James H. Martin (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. CoRR, cs.CL/0205028.

George-Bogdan Ivanov (2018). Natural Language Processing for Hackers: Learn to build awesome apps that can understand people, Manning Publications.

"Infographic – 4 Types of Data Annotation" [Web page]. (n.d.). Retrieved from https://innodata.com/infographic-4-types-of-data-annotation/#

James Pustejovsky & Amber Stubbs (2012). Natural Language Annotation for Machine Learning: A Guide to Corpus-Building, O'Reilly Media.

Jonathan Johnson (2020). "What Is Human in The Loop (HITL) Machine Learning?" Retrieved from https://www.bmc.com/blogs/hitl-human-in-the-loop/, 2020

"JSON input format for training" [Web page]. (n.d.). Retrieved from https://spacy.io/api/annotation#vocab-jsonl

Kalika Bali, Monojit Choudhury, and Priyanka Biswas (2010). Indian Language Part-of-Speech Tagset: Bengali LDC2010T16. Web Download. Philadelphia: Linguistic Data Consortium.

"Language Processing Pipelines" [Web page]. (n.d.). Retrieved from https://spacy.io/usage/processing-pipelines.

Leipzig Corpora Collection (n.d). Retrieved from https://corpora.uni-leipzig.de?corpusId=ben_community_2017.

Limarc Ambalina (2019). https://lionbridge.ai/articles/an-introduction-to-5-types-of-text-annotation/

L. Ratinov & D. Roth (2009). Design challenges and misconceptions in named entity recognition. In Proceedings of the Thirteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics.

M. A. A. Mumin, M. H. Seddiqui, M. Z. Iqbal, M. J. Islam (2018). SUPARA0.8M: A BALANCED ENGLISH-BANGLA PARALLEL CORPUS, Retrieved from https://ieee-dataport.org/documents/supara08m-balanced-english-bangla-parallel-corpus

Martin Wynne (Editor) (2005). Developing Linguistic Corpora: A Guide to Good Practice (AHDS Guides to Good Practice) Oxbow Books.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. CoRR, abs/1803.07640

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Niladri Sekhar Dash (2013). Part-of-speech (POS) Tagging of Bengali Written Text Corpus, Bhasa Bijnan o Prayukti: An International Journal on Linguistics and Language Technology, Vol. 1, No. 1.

"Parts of Speech Tagset: Bengali" [Web page]. (2009). Version 0.3, IDCIL CENTRAL INSTITUTE OF INDIAN LANGUAGES.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. arXiv preprint arXiv:2003.07082

Robert Munro (2019). Human-in-the-Loop Machine Learning: Active learning, annotation, and human-computer interaction, Manning, MEAP.

S. Banerjee, S.K. Naskar, S. Bandyopadhyay (2014). Bengali Named Entity Recognition Using Margin Infused Relaxed Algorithm. In: Sojka P., Horák A., Kopeček I., Pala K. (eds) Text, Speech and Dialogue. TSD 2014. Lecture Notes in Computer Science, vol 8655. Springer, Cham. https://doi.org/10.1007/978-3-319-10816-2_16

Steven Bird, Ewan Klein and Edward Loper (2009). Natural Language Processing with Python. O'Reilly Media.

Sudeshna Sarkar & Anupam Basu (2007). Part-of-Speech Tagging for Bengali. 10.3115/1557769.1557833.

"Text Annotation for Human" [Web page]. (n.d.) Retrieved from https://doccano.herokuapp.com.

"The most widely used online corpora" [Web page]. (n.d.). Retrieved from https://www.english-corpora.org

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. ArXiv, abs/1910.03771.

"Universal Dependencies" [Web page]. (n.d.). Retrieved from https://universaldependencies.org/introduction. html

Vikram Singh Bisen (2020). "What is Human in the Loop Machine Learning", Retrieved from https://medium.com/vsinghbisen/what-is-human-in-the-loop-machine-learning-why-how-used-in-ai-60c7b44eb2c0

Appendix

a. Brief Metadata format

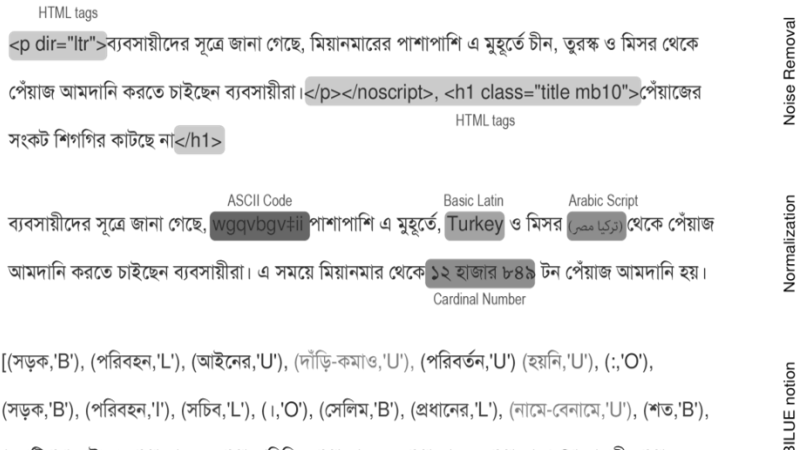
```
{
    "Corpus": {
        "items": [{
                "Doc_title": "A Special Event",
                "Author/s": "Single or more authors",
                "wrote": "2011",
                "published": "2012",
                "Source_type": "Newspaper/Textbook/Youtube",
                "Source_medium": "Web/Transcribed/Printed",
                "Link":"http://www.yourdomain.com/events.htm",
                "Topic": "Politics/Autobiography/feedback",
                "Formality": "Casual/Formal",
                "Essay_style": "Narration/Conversation",
                "Essay_PoV": "First/Second/Third",
                "Genre": "Informative/Imaginative",
                "Copyright": "Free/restricted",
                "Annotator": "name"
        },{
                "Doc_title": Ò†cv÷gv÷viÓ,
                "Author/s": Òiex›`ªbv_ VvKziÓ,
                "wrote": "1298",
                "published": "1298",
                "Source_type": "Textbook",
                "Source_medium": "Web",
                "Link":"https://www.tagoreweb.in/Stories/galpoguchchho-
                84/postmastar-1700",
                "Topic": "Creative writings",
                "Formality": "Formal",
                "Essay_style": "Both",
                "Essay_PoV": "Third",
                "Genre": "Imaginative",
                "Copyright": "Free",
                "Annotator": "Rashid"
        }]
    }
}
```

b. Structure of annotated data

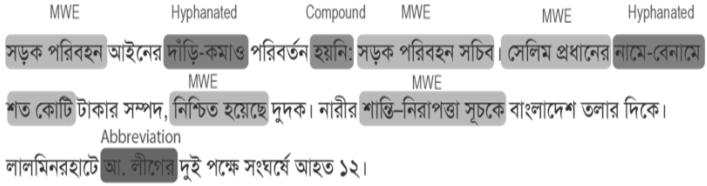
```
[{
  "doc_id": int,                # ID of the document within the corpus
  "paragraphs": [{              # list of paragraphs in the corpus
```

```
    "raw": string,              # raw text of the paragraph
    "sentences": [{             # list of sentences in the paragraph
      "tokens": [{              # list of tokens in the sentence
        "BILUO": string,        # BILUO label, e.g."O" or "U"
        "token_id": int,        # index of the token in the document
        "dep": string,          # dependency label
        "head": int,            # offset of token head relative to token index
        "tag": string,          # part-of-speech tag
        "orth": string,         # verbatim text of the token
        "ner": string           # Named entity label, e.g."PER" or "ORG"
      }],
      "brackets": [{            # phrase structure
        "first": int,           # index of first token
        "last": int,            # index of last token
        "label": string         # phrase label
      }]
    }],
    "cats": [{                  # categories for text classifier
      "label": string,          # text category label
      "value": float / bool     # label applies (1.0/true) or not (0.0/false)
    }]
  }]
}]
```

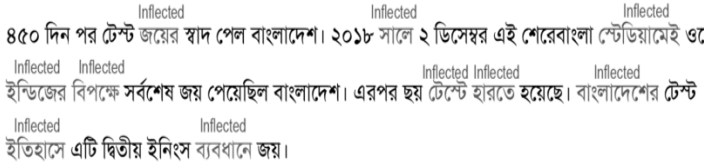
C. Visual Representation of Annotation with sample data



MWE Tokenization



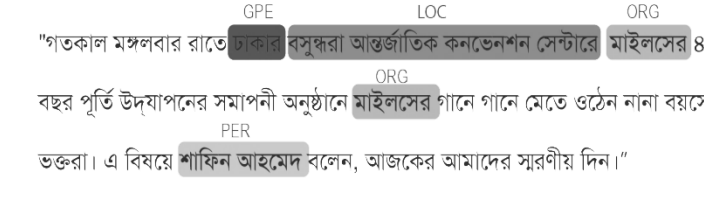
Lemmatization



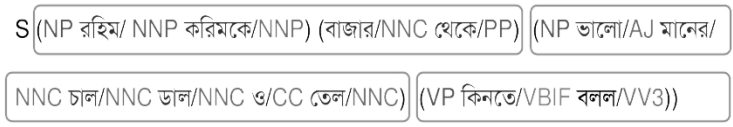
PoS Tagging

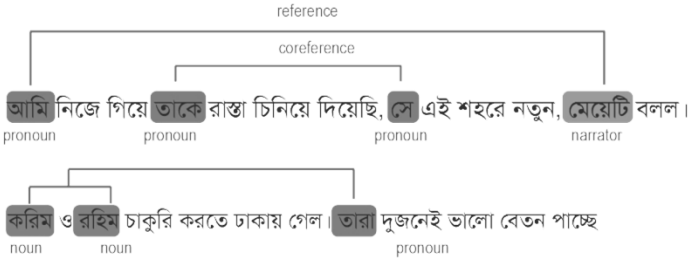
[('রহিম', 'NNP', 0, 2), ('করিমকে', 'NNP', 4, 7), ('বাজার', 'NNC', 9, 11), ('থেকে', 'PP', 13,14), ('ভালো', 'AJ', 16,17), ('মানের', 'NNC', 19, 21) ('চাল', 'NNC', 23,24), ('ডাল', 'NNC', 26,27), ('ও', 'CC', 29, 29), ('তেল', 'NNC', 31,32), ('কিনতে', 'VBIF', 34,36), ('বলল', 'VB3', 38,40) , ('।', '.')]

NER recognition

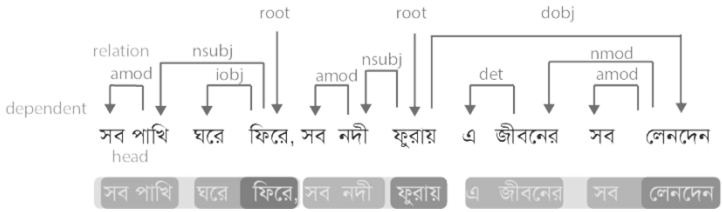


Chunking





Coreference



Deep Parsing