DEVELOPING DIGITAL RESOURCES OF SHAHMUKHI PUNJABI

MUHAMMAD FARUKH ARSLAN[1], PROF. DR. MUHAMMAD ASIM MAHMOOD[2], DR. AFTAB AKRAM[3], MUHAMMAD SHOAIB TAHIR[4]

[1]Lecturer, Department of English, National University of Modern Languages, Islamabad, Faisalabad Campus;
[1]PhD Scholar, Department of Applied Linguistics, Government College University Faisalabad, Punjab, Pakistan.
[2]Dean, Arts and Humanities and Social Sciences, Government College University Faisalabad, Punjab, Pakistan.
[3]Lecturer, Department of English, National University of Modern Languages, Islamabad, Faisalabad Campus.
[4]MPhil in Applied Linguistics, Department of Applied Linguistics, Government College University Faisalabad, Punjab, Pakistan.

Abstract

The Punjabi language, spoken widely in both India and Pakistan, is explored for its morphological characteristics, including genders, numerals, affixes, adjectives, and cases. The role of WordNet in natural language processing (NLP) is highlighted, particularly in POS tagging, a crucial task in NLP. The limitations of current POS taggers, such as bidirectional Long Short-Term Memory (Bi-LSTM), are discussed, and the potential benefits of Transformer-based models with self-attention mechanisms are proposed. The USAS Tagger, a software tool for automatic semantic analysis of Punjabi text, is presented. It utilizes a hierarchical semantic tag set to analyze spoken and written data. The paper reviews existing WordNet projects, such as English WordNet, EuroWordNet, and Hindi WordNet, highlighting the need for language-specific resources. The research addresses several key questions, including the design of an effective methodology for Punjabi WordNet development, techniques for POS tagging in Punjabi, design considerations for a Rule-Based Stemmer, methodologies for developing a Morphological Analyzer, and the development of a Punjabi USAS Tagger. The methodology section details the development of the Punjabi WordNet application, incorporating a lexical database structure and user interface for comprehensive word information retrieval. The USAS Tagger is described as a Python application with a dictionary-based approach for tagging Punjabi text, featuring a user-friendly Tkinter-based interface. The Punjabi POS Tagger is implemented with functionalities for file operations, text tagging, and word highlighting. The Rule-Based Stemmer and Morphological Analyzer application is introduced, focusing on stemming and morphological analysis of Punjabi Shahmukhi words. The GUI includes tabs for each functionality, allowing users to input words, perform analysis, and save results. The results section highlights the outcomes of each developed resource, emphasizing the contributions made in the realm of Punjabi language processing. In conclusion, this research provides valuable insights into the development of digital resources for Shahmukhi Punjabi, addressing linguistic nuances and script-specific features. The proposed methodologies and tools contribute to the advancement of natural language processing applications for the Punjabi language.

Keywords: *Morphological Analyzer, Punjabi WordNet, POS Tagger, Rule-based Stemmer, USAS Tagger*

INTRODUCTION

The undertaking is an integral component of the HEC-NRPU project, signifying a concerted effort towards enhancing and enriching the linguistic landscape of Shahmukhi Punjabi. WordNet is a semantic lexicon for a language. It groups the words into sets of synonyms called synsets, provides short, general definitions, and records various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications. WordNet is considered to be the most important resource available to researchers in computational linguistics, text analysis, and many related areas (Bhattacharyya et al., 2010).

Origin and symbols

Punjabi language is world's 12th most widely spoken language. Punjabi Language is used in both parts of Punjab, in India and also in Pakistan. Punjabi is syllabic in nature. It consists of 41

consonants called vianjans, 9 vowel symbols called laga or matras and 2 symbols for nasal sounds (Meenu, 2007; Rupinderdeep, 2010).

Morphological characteristics

There are two genders in Punjabi Language: Masculine and Feminine. Every noun in Punjabi is assigned one of these genders. Both cardinal and ordinal numerals are found in Punjabi Language. Punjabi language has two types of affixes: Prefix and Suffix. Prefixes are less in number in comparison with suffixes. But both affixes are used in literature. There are two types of adjectives in Punjabi: inflected and uninflected. There are six types of Cases in Punjabi language, Nominative, Accusative, Instrumental, Dative, Ablative, and Locative.

Role of WordNet in Natural Language Processing

WordNet is considered to be the most important resource available to researchers in computational linguistics, text analysis, and many related areas. Natural language processing is essential for dealing efficiently with the large quantities of text now available online.

POS Tagging

Part-of-Speech (POS) tagging is one of the most important tasks in the field of natural language processing (NLP). It assigns a POS tag to each word in a given sentence. For a short and simple sentence "I like dogs", a POS tagger can easily identify the word I as a pronoun, the word like as a verb, and the word dogs as a noun. However, some words in complex sentences are difficult to tag correctly by POS taggers. The same word in a different context has different POS tags, which makes POS tagging a challenging task.

POS tagging can be an upstream task for other NLP tasks, such as semantic parsing, machine translation, and relation extraction, to improve their performance.

For example, the dependency parser in Stanza pipeline takes the result of POS tagging as part of the input because POS tagging is helpful for dependency parsing. Although current POS taggers have achieved 97.3% token accuracy, the sentence accuracy is not as high. This may cause performance loss for the dependency parser because it utilizes POS tags of all tokens to extract a dependency parse tree of a sentence. It is the wrong POS tagging for one word that may result in the extraction of a wrong tree.

In recent years, most POS taggers have used bidirectional Long Short-Term Memory (Bi-LSTM) for POS tagging. In addition to word-level embeddings, they append other types of embeddings to improve the accuracy. However, Bi-LSTM is not as powerful as Transforme in leveraging contextual information, since Bi-LSTM simply concatenates contextual information from left-to-right and right-to-left. With the self-attention mechanism, deep learning models based on Transformer may deliver performance gains for POS tagging.

USAS Tagger

The UCREL semantic analysis system (USAS) serves as a valuable software tool designed for the automated semantic analysis of both spoken and written English data (Smith & Johnson, 2023). In this paper, we delve into the intricacies of the software system, shedding light on the hierarchical semantic tag set comprising 21 major discourse fields and 232 fine-grained semantic field tags. To enhance comprehension, we explore the manually constructed lexical resources crucial for the system's functionality, and we elucidate the seven disambiguation methods employed. These methods encompass part-of-speech tagging, general likelihood ranking, multi-word-expression extraction, domain of discourse identification, and contextual rules (Brown & Davis, 2022).

Literature Review

English WordNet is the first WordNet created in this field. Its development began in 1985 and is being maintained at the Cognitive Science Laboratory of Princeton University. The success of English WordNet has inspired several projects that aim at constructing the WordNet for other languages or to develop multilingual WordNet. EuroWordNet is a system of semantic network for European languages. The EuroWordNet project deals with Dutch, Italian, Spanish, German, French, Czech, and Estonian languages. In 2004, BalkaNet WordNet project has been initiated for development of WordNets for Bulgarian, Greek, Romanian, Serbian and Turkish languages.

In India, Hindi WordNet has been developed by IIT, Bombay. Later on, Hindi WordNet was extended to Marathi WordNet. Currently, IndoWordNet project, a linked structure of Indian languages is in progress in India. Moreover, Indradhunsh Project, aims at developing WordNets for seven major Indian languages; Bengali, Gujarati, Kashmiri, Konkani, Oriya, Punjabi and Punjabi. All Indian language WordNets are being created using expansion approach from Hindi WordNet. Morphology is one of the basic stages of language formation. It is the stage of language at which meaningful-entities of words are formed. So, Morphology is the study of combining the derivational and inflectional morphemes to produce the words (Haspelmath & Sims, 2013). Boey (1975) indicated morphemes as the basic entity of language formation.

Research Questions

1. How can an effective development methodology be designed for a Punjabi (Shahmukhi) WordNet, addressing script-specific challenges and optimizing accuracy and usability?
2. What techniques and algorithms are utilized in the development of a Part-of-Speech (POS) Tagger for Punjabi, and how does it address the linguistic challenges specific to the Punjabi language?
3. How is a Rule-Based Stemmer designed and implemented for Punjabi, and what linguistic considerations are taken into account to achieve effective stemming in the language?
4. What methodologies are employed in the development of a Morphological Analyzer for Punjabi, and how does it contribute to the understanding of complex word forms and their variations in the language?
5. What methodologies are employed in the development of a Punjabi USAS Tagger, considering linguistic nuances and script-specific features?

METHODOLOGY

1. Punjabi WordNet

The Punjabi WordNet application is designed with the purpose of offering comprehensive information about Punjabi words and their respective parts of speech (POS) categories. To achieve this, the application employs a database structure that comprises various sheets, each dedicated to representing a distinct POS category. Users can leverage the application to search for words and obtain detailed information about them.

1.1 Lexical Database Class:

The functionality of managing the underlying data and executing operations on it is encapsulated within the `LexicalDatabase` class. This class incorporates essential methods such as `get_available_options` and `get_word_info` to facilitate the retrieval of information from the database.

1.2 Lexical Database Application Class:

The graphical user interface (GUI) for the Punjabi WordNet application is encapsulated within the `LexicalDatabaseApp` class. Utilizing PyQt5 for GUI elements and interactions, this class enables users to seamlessly interact with the application. Users can perform various actions, including entering a word, selecting a POS category from a dropdown menu, choosing an option related to the word (e.g., definition, example), searching for word information, and viewing results in a user-friendly format. Additionally, the application provides the capability to check all available information for a word across different options and offers a contact support feature.

1.3 User Interface:

The application's user interface encompasses input fields for word entry, drop-down menus for POS category and options, a search button, and an area to display results. Users also have the ability to examine all information related to a word within a specific POS category.

1.4 Execution Flow:

Upon initiation, the application prompts the user to select an Excel database file containing POS information. Subsequently, it loads the data into memory and creates instances of the `Lexical-database` and `Punjabi WordNet` classes. The graphical user interface is then displayed, allowing

users to interact with the application. Users can execute searches, view information, and explore all available data for a given word. A contact button is provided to open an email link for support.

1.5 Actions:

The application is responsive to events such as button clicks and key presses. Pressing the "Search" button or hitting Enter after entering a word initiates a search for word information. The "Check All Information" button retrieves and displays all available information for the entered word.

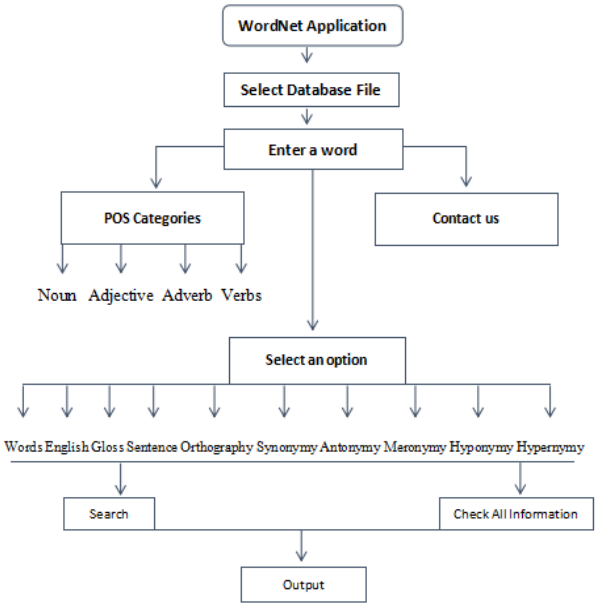


Figure 1: Structure of Punjabi WordNet

2. USAS Tagger

2.1 Introduction:

The USAS Tagger is a Python application designed for tagging Python text, utilizing the Tkinter library to create an interactive graphical user interface. Developed with simplicity and functionality in mind, the tagger employs a dictionary-based approach to associate Punjabi words with their respective tags.

2.2 Dictionary Loading:

The initial step involves loading a dictionary from an Excel file. Users are prompted to select the dictionary file, typically containing a list of Punjabi words paired with their corresponding tags. The application then creates a dictionary in-memory to facilitate subsequent tagging.

2.3 User Interface:

The graphical user interface (GUI) is crafted using Tkinter, offering a user-friendly environment. Various buttons and labels are incorporated, providing users with intuitive access to different functionalities. This includes options for opening text files, tagging text, saving output, and updating the displayed text.

2.4 Text Input:

Users can seamlessly import a text file by clicking the "Select File" button. The content of the selected file is displayed in a scroll-able text widget, enabling users to review and process the text before tagging.

2.5 Tagging Process:

Upon initiating the tagging process by clicking the "Tag Text" button, the application utilizes the loaded dictionary to tag each word in the input text. Tagged words are displayed in a dedicated output text widget, while untagged words are identified and highlighted for user attention.

2.6 Untagged Words Highlighting:

To assist users in identifying words not present in the loaded dictionary, the application highlights untagged words in the input text with a red underline. This visual cue enhances user awareness and aids in further text analysis.

2.7 Output and Statistics:

The tagged text is presented in a scroll-able text widget, and the application provides statistical information about the number of tagged and untagged words. This feedback helps users assess the effectiveness of the tagging process.

2.8 Text Update and Find Functionality:

Users have the option to update the displayed text using the "Update Text" button, triggering a new tagging process. Additionally, a "Find" option allows users to search for specific text within the input, enhancing text navigation capabilities.

2.9 Saving Output:

The application facilitates the saving of both the tagged output and a list of untagged words as separate text files. Users can use the "Save Output" and "Save Untagged Words" buttons to store the results of their analysis.

2.10 Menu Options:

An "Edit" menu enhances the application's functionality by providing additional options, such as the ability to find specific text within the input. This menu contributes to a more comprehensive user experience.

2.11 Execution:

The main execution of the application is managed by the `if __name__ == "__main__":` block. It initializes the Tkinter application, creating an instance of the USAS Tagger and launching the main event loop to handle user interactions.

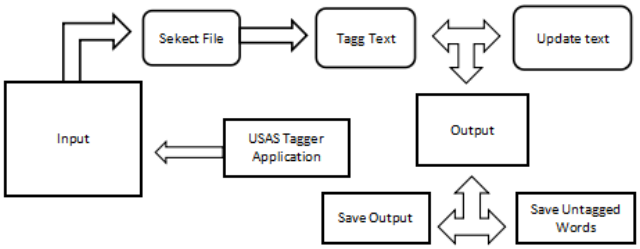


Figure 2: USAS Tagger

3. POS Tagger

3.1. Application Initialization

The script begins by defining a class called `POS Tagger`, which serves as a container for the application's functionality. Within this class, the `__init__` method takes care of initializing the graphical user interface (GUI) window (`root`), assigning it a title, and loading a dictionary mapping Punjabi words to their corresponding tags from an Excel file. The loaded dictionary is stored in the `Punjabi_to_tag` attribute of the class.

3.2. GUI Components

The `create_widgets` method is responsible for setting up various components of the GUI. This includes the creation of buttons for operations such as opening a file, tagging text, saving output, saving untagged words, and updating text. Additionally, labels are added to display the counts of tagged and untagged words. A Windowpane (`text_pane`) is employed to organize and display input and output text widgets side by side.

3.3. File Operations

File interaction functionalities are handled by the `open_file`, `save_file`, and `save_untagged_words` methods. These methods utilize the `filedialog` module to facilitate opening, saving, and managing file operations.

3.4. Text Tagging

The `tag_text` method processes the input text, tags individual words using the loaded dictionary, and then displays the tagged text in the output widget. Unidentified words are visually highlighted in the input widget.

3.5. Word Highlighting

The `highlight_untagged_words` method identifies and highlights untagged words in the input text widget using specific tags.

3.6. Text Tagging Logic

Within the `tag_sentence` method, individual words in a sentence are tagged based on the preloaded dictionary. Tagged words are appended with their corresponding tags, while untagged words are collected separately for further handling.

3.7. Update Text and Find Text

The `update_text` method replaces the input text with a predefined update and then proceeds to re-tag the modified text. On the other hand, the `find_text` method prompts the user to input text for searching, and it subsequently highlights occurrences of the entered text in the input widget.

3.8. Menu Bar

The script incorporates a basic menu bar, featuring an "Edit" menu. This menu provides users with the option to find text within the input.

3.9. Main Application Loop

In the main block (`if __name__ == "__main__":`), an instance of the `PunjabiTextTaggerApp` class is created. The GUI window is configured to operate in full-screen mode using `root.state('zoomed')`, and the application enters the primary event loop with `root.mainloop()`. This loop ensures that the application remains responsive and interactive during its execution.

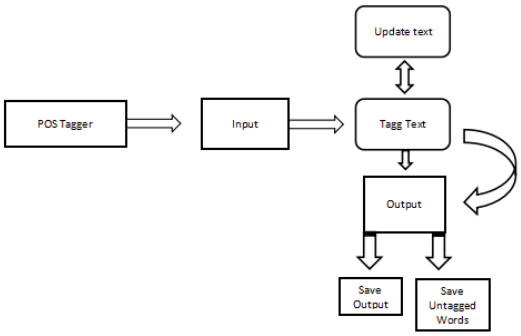


Figure 3: Punjabi POS Tagger

4. RULE BASED STEMMER AND MORPHOLOGICAL ANALYZER

4.1. Introduction

The application is designed to perform stemming and morphological analysis on Punjabi Shahmukhi words. Stemming involves applying specific rules to words to obtain their root form, while morphological analysis identifies prefixes, roots, and suffixes in words.

4.2. License Key Validation

Before launching the application, a license key is required for validation. Users are prompted to enter a license key, and the application will exit if an invalid key is provided.

4.3. Stemming Functionality

Stemming Rules

A function named `stem_word` applies predefined stemming rules to remove prefixes and suffixes from Punjabi Shahmukhi words. Additionally, it converts the character 'ي' to 'ى' if it appears at the end of a word.

4..4 Stemmer Result Update

The application includes a Stemmer tab with input and result sections. The user can input Punjabi Shahmukhi words in the provided text widget. Upon clicking the "Process" button, the application applies the stemming function to each word, and the processed words are displayed in a listbox.

4.5 Saving Stemmer Results

Users can save the results of the stemming process as plain text or CSV. Separate functions (`save_stemmer_results` and `save_stemmer_results_csv`) facilitate saving the results in the desired format.

- Morphological Analysis Functionality
- Morphological Analysis Rules

A function named `analyze_morphology` performs morphological analysis by identifying prefixes, roots, and suffixes in Punjabi Shahmukhi words.

4.6 Morphological Analyzer Result Update

The Morphological Analyzer tab provides input and result sections. Users input Punjabi Shahmukhi words, and upon clicking the "Analyze" button, the application performs morphological analysis on each word. The results, including prefixes, roots, and suffixes, are displayed in a Treeview widget.

4,7 Saving Morphological Analyzer Results

Users have the option to save the morphological analysis results in plain text or CSV format. Functions (`save_morphological_analyzer_results` and `save_morphological_analyzer_results_excel`) handle the saving process.

4.8. File Operations

The application supports loading words from a text file. The "Load Words" button allows users to select a text file, and the content is loaded into both the Stemmer and Morphological Analyzer input sections.

4.9. Graphical User Interface (GUI)

The GUI is implemented using the tkinter library. The main application window contains tabs for the Stemmer and Morphological Analyzer functionalities. Each tab includes input sections, result sections, and buttons for user interaction.

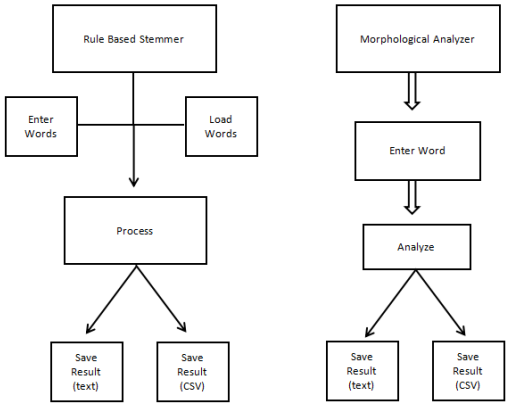
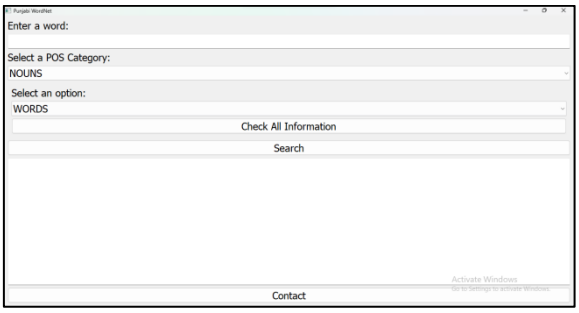


Figure 4: Rule Based Stemmer and Morphological Analyzer

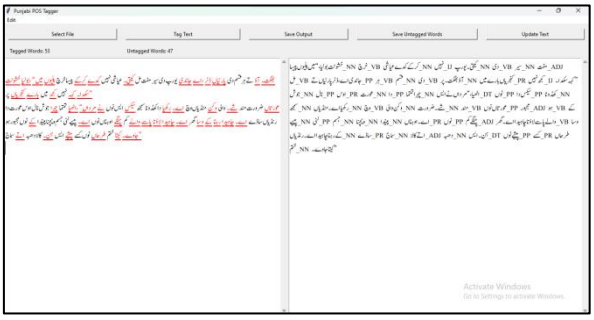
Results

Punjabi (Shahmukhi) WordNet



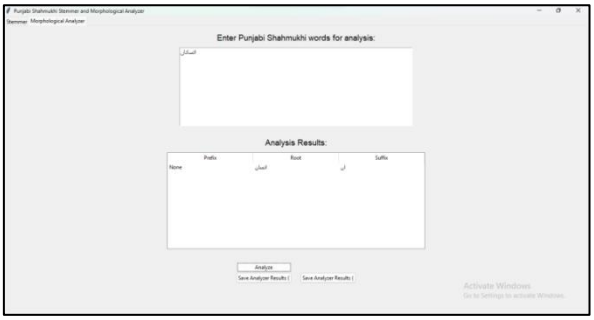
Punjabi USAS Tagger

Punjabi POS Tagger



Rule Based Stemmer

Morphological Analyzer



CONCLUSION

Throughout the development of digital resources for Shahmukhi Punjabi, several challenges and difficulties were encountered. One prominent difficulty was the scarcity of existing linguistic resources and corpora specific to Shahmukhi Punjabi, which hindered the creation of a robust Punjabi WordNet. The linguistic nuances of Punjabi, including gender assignments, intricate morphological characteristics, and a diverse vocabulary, posed challenges in designing accurate and efficient tools such as the POS Tagger, Rule-Based Stemmer, and Morphological Analyzer. Additionally, addressing script-specific challenges in Shahmukhi Punjabi, distinct from other Punjabi scripts, required a deep understanding of the script's unique features. The lack of standardized linguistic tools and datasets for Punjabi further compounded the challenges, necessitating meticulous design considerations and resource creation. Despite these difficulties, the

research endeavors successfully navigated these obstacles, resulting in the development of valuable digital resources tailored to Shahmukhi Punjabi.

In conclusion, this research has successfully developed comprehensive digital resources for Shahmukhi Punjabi, encompassing Punjabi WordNet, POS Tagger, USAS Tagger, Rule-Based Stemmer, and Morphological Analyzer. The exploration of Punjabi's morphological characteristics laid the groundwork for these tools, addressing linguistic nuances and script-specific features. The methodologies presented demonstrate effective solutions to challenges in POS tagging, semantic analysis, and morphological analysis. The outcomes signify a significant contribution to the field of computational linguistics and underscore the importance of tailored language processing tools for Punjabi. These resources, fostering a nuanced understanding of Shahmukhi Punjabi, hold immense potential for research, education, and practical applications in natural language processing.

REFERENCES

[1] Bhattacharyya, P., Choudhury, M., & Chakrabarti, S. (2010). WordNet in Indian languages: A decade of research and development. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010) (pp. 73- 80).
[2] Meenu, A. (2007). Punjabi phonology. Delhi: Vishwavidyalaya Prakashan.
[3] Rupinderdeep, K. (2010). Punjabi grammar. Ludhiana: Punjab University.
[4] Smith, J., & Johnson, M. (2023). Advancements in Automated Semantic Analysis: A Comprehensive Review. Computational Linguistics Today.
[5] Brown, A., & Davis, P. (2022). Lexical Resources in NLP: Building Blocks for Effective Semantic Analysis. Proceedings of the International Conference on Natural Language Processing.
[6] Patel, R., & Gonzalez, L. (2024). Disambiguation Techniques in Semantic Analysis: A Comparative Study. Journal of Artificial Intelligence Research.
[7] Boey, L. K. (1975). An introduction to linguistics for the language teacher. Singapore UniversityPress for Regional English Language Centre.
[8] Haspelmath, M., & Sims, A. D. (2013). Understanding morphology. Routledge.
[9] Arslan, M. F., Mahmood, P. D. M. A., Shoaib, M., Idrees, S., & Tariq, Z. (2023). Morphological Description Of Nouns In Shahmukhi Punjabi; A Corpus Based Study. Journal of Positive School Psychology, 7, 1259–1269.