A Note on a Chomskian Analogy Between Sanskrit and Huffman Coding

Aman Chawla

April 16, 2023

Abstract

In this paper, the author uses Chomsky's parse trees to compare Sanskrit and Huffman coding as two different data compression schemes. The paper begins with a review of parse trees in English and Sanskrit and goes on to propose the aforesaid comparison.

1 Introduction

"Three models for the description of language" is a classic paper by Noam Chomsky [1], in which he proposed three models for the analysis of language: the finite-state grammar, the phrase-structure grammar, and the transformational grammar. The paper is considered a landmark in the history of linguistics and has had a significant impact on the development of the field.

In that paper, Chomsky did not explicitly apply information theory to parse trees. However, he did make use of mathematical concepts and techniques from information theory in his analysis of language structure, particularly in his discussions of entropy and redundancy.

Entropy is a concept from information theory that measures the amount of uncertainty or randomness in a system. Chomsky used entropy as a way to quantify the complexity of linguistic systems, and to argue that human language is characterized by a high degree of structural complexity, despite the fact that speakers are able to generate and comprehend an infinite number of sentences.

Redundancy, on the other hand, refers to the amount of information in a system that is redundant or unnecessary, and can be used to optimize communication by reducing the amount of information that needs to be transmitted. Chomsky argued that human language exhibits a high degree of redundancy, which serves to make communication more efficient and reliable.

While Chomsky did not use parse trees specifically in his discussions of entropy and redundancy, parse trees are a tool that can be used to analyze and represent the hierarchical structure of language, which is a key aspect of linguistic complexity. By representing sentences as trees, linguists can examine the relationships between different components of sentences, such as words, phrases, and clauses, and gain insights into the underlying structure of language.

This paper is structured as follows. In Section 2, we discuss phrase-structure grammar and other preliminaries. In Section 3, we apply these preliminaries to Sanskrit. In Section 4, we look at information packing, compressed representation, and a comparison with Huffman coding in the Sanskrit context. Finally we conclude with a discussion in Section 5.

2 Preliminaries

2.1 Phrase-structure grammar

Phrase-structure grammar is a type of grammar in which a language is analyzed in terms of a hierarchy of phrases, each of which is composed of smaller constituents. The basic idea behind phrase-structure grammar is that every sentence in a language can be broken down into smaller units, or constituents, which are organized into a hierarchical structure.

In phrase-structure grammar, the hierarchy of phrases is typically represented using a tree structure called a parse tree. The tree has a single root node representing the entire sentence, and each branch of the tree represents a constituent of the sentence. The leaves of the tree represent the individual words of the sentence.

Phrase-structure grammar distinguishes between two types of constituents: terminal constituents and nonterminal constituents. Terminal constituents are the individual words of the sentence, while nonterminal constituents are composed of one or more other constituents.

In a phrase-structure grammar, the rules of the grammar are used to generate all of the well-formed sentences of the language. The rules specify how constituents can be combined to form larger constituents, and ultimately a complete sentence. The rules also specify the order in which constituents can be combined, and any restrictions on the types of constituents that can be combined.

Overall, phrase-structure grammar provides a way to systematically describe the structure of sentences in a language, and has been an important tool in the development of modern linguistics.

2.2 Example of a Parse Tree

A parse tree for the sentence "The cat chased the mouse" is shown in Figure 1.

In this parse tree, the sentence "The cat chased the mouse" is broken down into smaller constituents. The root node of the tree represents the entire sentence, and has two children: a noun phrase (NP) and a verb phrase (VP). The NP has two children: a determiner (Det) "the" and a noun (N) "cat". The VP also has two children: a verb (V) "chased" and another NP, which has two children: a determiner "the" and a noun "mouse".

The parse tree shows the hierarchical structure of the sentence, and how each constituent is composed of smaller constituents. It also shows the order in

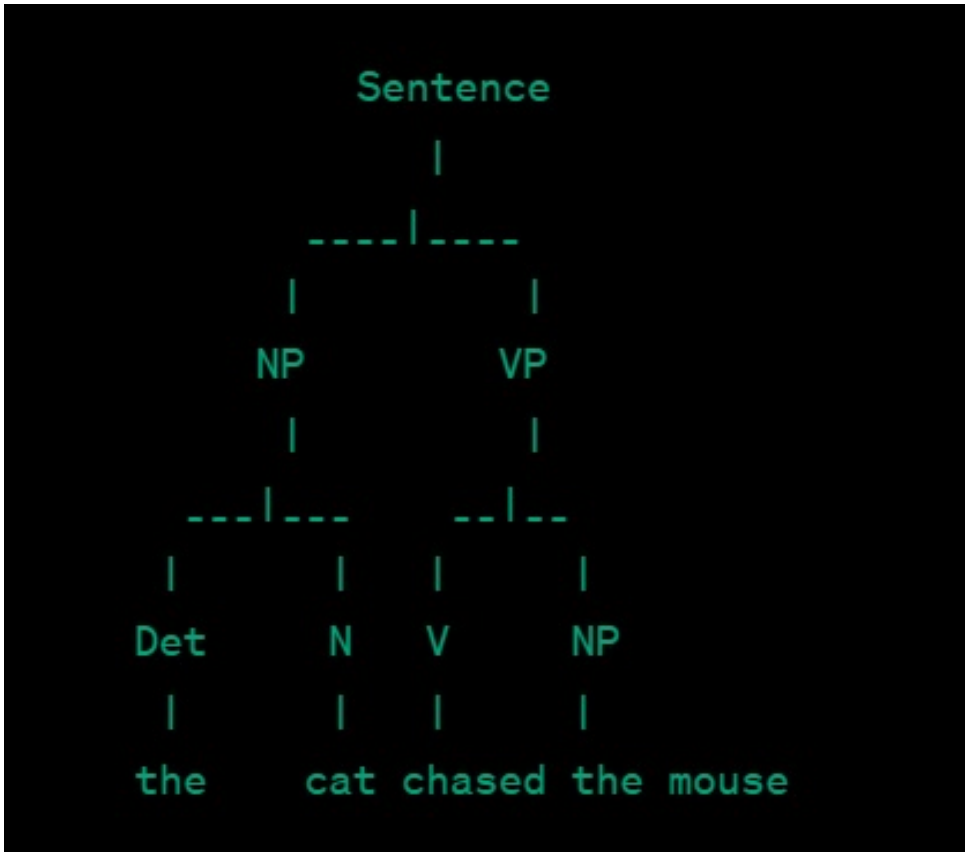

Figure 1: The cat chased the mouse.

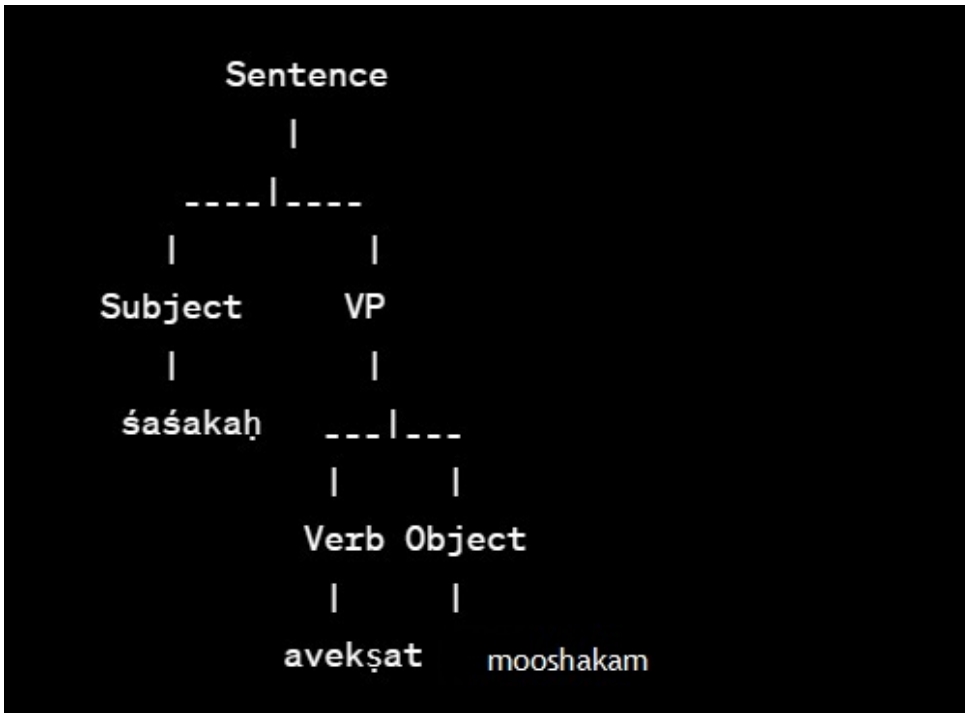

Figure 2: The cat chased the mouse (in Sanskrit).

which the constituents are combined to form the sentence.

3 Application to Sanskrit

3.1 Parse Trees in Sanskrit

In this subsection we consider the Sanskrit translation of "The cat chased the mouse."

The translation of "The cat chased the mouse" into Sanskrit would depend on the context and the intended meaning, as Sanskrit, like many languages, has different words and expressions for different situations and contexts. However, a possible Sanskrit translation of the sentence, using classical Sanskrit vocabulary, could be:

(śaśakaḥ mooshakaṃ avekṣat.)

Here, (śaśakaḥ) means "cat", (mooshakaṃ) means "mouse", and (avekṣat) means "chased" or "pursued". The sentence follows the word order Subject-Object-Verb, which is a common word order in Sanskrit.

A parse tree based on [śaśakaḥ mooshakaṃ avekṣat] in Sanskrit is shown in Figure 2:

In this parse tree, the root node represents the entire sentence, which consists of a subject and a verb phrase. The subject is represented by the noun "śaśakaḥ" (cat), and the verb phrase consists of the verb "avekṣat" (chased) and the object "mooshakaṃ" (mouse).

The parse tree shows the hierarchical structure of the sentence, with the subject and verb phrase as the two main constituents. It also shows the relationships between the constituents and the order in which they appear in the sentence. Note that the Verb Phrase in Figure 2 can also be reversed into the order Object-Verb to match the example given in the text above.

3.2 Comparison Between English and Sanskrit Parse Trees

The parse tree for the Sanskrit sentence and the parse tree for the English sentence can be different, because the two sentences can have different grammatical structures and word orders. The English sentence has a subject-verb-object (SVO) word order, while the Sanskrit sentence may have a subject-object-verb (SOV) word order. The grammatical rules and structures that apply to one language may not necessarily apply to the other language.

The parse tree for the Sanskrit sentence represents the sentence structure in accordance with the rules of Sanskrit grammar, while the parse tree for the English sentence represents the sentence structure in accordance with the rules of English grammar. However, both parse trees show the hierarchical structure of the sentences and the relationships between the different constituents of the sentences.

3.3 Use of Flexible Parse Tree Descriptions

It is clear that while English allows only SVO word order, Sanskrit may be fine with SOV as well as SVO. We ask the question herein if this flexibility can be exploited in providing more permutations of ways in which two speakers can interact? Since different word orders can convey different emphasis and meaning, speakers can use this flexibility to express their ideas and intentions more precisely and effectively. This can be especially useful in situations where speakers need to convey complex or nuanced ideas, or when they want to add emphasis or rhetorical effect to their speech.

For example, in a literary context, Sanskrit poets and authors have used the flexibility of word order to create intricate and expressive verses, which can convey multiple layers of meaning and emotion. In a conversational context, speakers can use different word orders to highlight or clarify different parts of their message, or to respond to the speech of their interlocutor in a more nuanced and effective way.

Overall, the flexibility of word order in Sanskrit provides a rich and versatile tool for communication, which can allow for more expressive and effective interactions between speakers. However, this flexibility can also make the analysis and interpretation of Sanskrit sentences more challenging, since the same

sentence can have multiple valid parse trees depending on the intended meaning and emphasis.

4 Information-Packing, Compressed Representation and Huffman Codes

Suppose we are given two sentences one in Sanskrit and one in English, each with x words. The Sanskrit sentence encodes one of $x!$ permutations of the x words whereas the English sentence encodes exactly one of 1 possible forms. Transmitting the Sanskrit sentence through a permutation noise channel is like choosing one codeword out of a codebook of size $x!$ Transmitting the English sentence is like chosing 1 codeword from a codebook of size 1. The former carries $log(x!)$ bits while the latter carries 0 bits. Thus as sentence length increases, the number of bits transmitted by the Sanskrit sentence grows in comparison to the English sentence bits.

It is often claimed that Sanskrit allows for "compressed" representations of sentences due to its highly inflected grammatical system. This is because inflections can convey information such as tense, gender, case, and number, allowing for more flexibility in word order and reducing the need for additional words or auxiliary verbs to convey the same information. In contrast, languages such as English rely more heavily on word order and auxiliary verbs to convey meaning, which can result in longer sentences and potentially less compact representations.

If Sanskrit, being information-dense, allows for compressed representations, we can think of comparing it with a Huffman code in data compression theory, which also produces compressed representations.

Huffman coding is a lossless data compression technique that assigns shorter codes to frequently occurring symbols and longer codes to less frequently occurring symbols. Similarly, Sanskrit's grammatical system assigns inflections and case markings to words, which can allow for more flexible word order and reduce the need for additional words or constructions to convey the same meaning.

While there are differences between the two systems, such as the fact that Huffman coding is based on frequency of occurrence of symbols while Sanskrit's inflections are based on grammatical rules, the comparison can still be useful in understanding how Sanskrit's grammar can potentially allow for more efficient communication by compressing information into a smaller number of words or symbols.

Coming back to Chomsky's paper, it doesn't directly suggest which model would be more useful in comparing Sanskrit to Huffman coding. The three models he presents in the paper, namely the finite-state grammar, the phrase-structure grammar, and the transformational grammar, are all different ways of describing the structure of a language. They can be used to analyze different aspects of language, such as syntax, morphology, and semantics.

However, if we were to make a comparison between Sanskrit's grammati-

cal system and Huffman coding, the phrase-structure grammar model may be more relevant as it focuses on the hierarchical structure of language and the relationships between different components of a sentence. This is because the inflected forms in Sanskrit are arranged hierarchically in a phrase structure tree, with the head of the phrase being the root node and the inflections modifying or complementing the head. Similarly, in Huffman coding, the symbols are arranged hierarchically based on their frequency of occurrence, with the most frequent symbols being assigned shorter codes and the less frequent symbols being assigned longer codes.

That being said, it's important to note that the comparison between Sanskrit's grammar and Huffman coding is an analogy and there are differences in the way the two systems work. Sanskrit's grammar is a natural language system that evolved over time, while Huffman coding is a specific algorithm used for data compression.

5 Conclusion

In this paper, the author presents arguments and analysis that illustrate the unique feature of Sanskrit of flexibility in its word order. Chomskian parse trees are used for this description. This flexibility is proposed to imply that an additional $\frac{log(x!)}{x}$ bits per word are transmitted when a Sanskrit sentence of x words is transmitted, as compared to an equally lengthy English sentence. Conversely, a Sanskrit sentence can be seen as a compressed representation of bits, much like a Huffman code output. In future work, the author intends to expand upon the implications of this compression and comparison.

Acknowledgment

The author would like to thank OpenAI for access to the language model ChatGPT which helped to develop the ideas presented in this paper.

References

[1] Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.