Semantically Cohesive Word-Grouping in Indian Languages

Anonymous ACL submission

Abstract

Indian languages are inflectional and agglutinative and typically follow clause-free word order. The structure of sentences across most major Indian languages are similar when their dependency parse trees are considered. While some differences in the parsing structure occur due to peculiarities of a language or its preferred natural way of conveying meaning, several apparent differences are simply due to the granularity of representation of the smallest semantic unit of processing in a sentence. The semantic unit is typically a word, typographically separated by whitespaces. A single whitespace-separated word in one language may correspond to a group of words in another. Hence, grouping of words based on semantics helps unify the parsing structure of parallel sentences across languages and, in the process, morphology. In this work, we propose *word-grouping* as a major preprocessing step for any computational or linguistic processing of sentences for Indian languages. Among Indian languages, since Hindi is one of the least agglutinative, we expect it to benefit the most from word-grouping. Hence, in this paper, we focus on Hindi to study the effects of grouping. We perform quantitative assessment of our proposal with an intrinsic method that perturbs sentences by shuffling words as well as an extrinsic evaluation that verifies the importance of word-grouping for the task of Machine Translation (MT) using decomposed prompting. We also qualitatively analyze certain aspects of the syntactic structure of sentences. Our experiments and analyses show that the proposed grouping technique brings uniformity in the syntactic structures, as well as aids underlying NLP tasks.

1 Introduction

The process of extracting meaningful phrases from sentences, known as chunking, is an important task in NLP. From a more granular level, the ability to identify semantic units of a sentence can be advantageous for a variety of NLP applications. In this paper, we discuss the importance of word-grouping in a sentence, which together form a single, independent meaningful unit of the sentence.

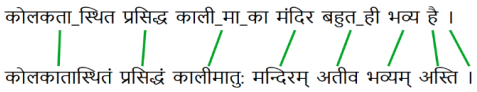


Figure 1: Alignment of parallel sentences in Hindi and Sanskrit, after word-grouping.

Majority of Indian languages follow similar grammatical structure. The key changes in a syntactic structure like a dependency parse tree of a sentence, emerge mostly from differences in the number of whitespace-separated words[1] that represent a particular semantic concept. This variation occurs since we consider the words of a sentence as the basic units of processing. When we consider parallel sentences in various Indian languages, generally it is possible to obtain a non-overlapping word/phrase-level alignment. The major reason for not having a one-to-one mapping is the variation in the word count as discussed above. We find that grouping of words helps in a better alignment of Indian languages. Figure 1 displays a pair of parallel sentences in Hindi and Sanskrit. It shows how multiple words in one language correspond to a single word in another language.

Dangarikar et al. (2024) show that Hindi language exhibits a significant deviation from other major Indian languages in terms of the number of words used to represent a concept. Data statistics provided by Gerz et al. (2018) using Polyglot Wikipedia also show a similar trend. The reason for such a deviation is that, among the Indian languages considered, Hindi is the least agglutinative in nature (Pimpale et al., 2014) and, at times, follow isolating features. Owing to such a deviation, we expect the word-grouping effort to be more crucial and effective for Hindi and, hence, in this paper, we

[1] In the paper, usage of 'word' is for the whitespace-separated texts in any sentence.

focus on word-grouping for Hindi. For example, the words जा रहा है (jā rahā hai[2]) in Hindi, corresponds to a single word hōgutiddāne in Kannada and yācchē in Bangla. Similarly, while Hindi tends to use "case-markers" such as kī, kē, etc. as separate words, highly agglutinated languages like Kannada and Malayalam use inflectional suffixes fused with the word. However, we emphasize that such differences are only at the typographic surface-level, and the *underlying semantic structure* of the languages is similar. Thus, (mōhana sē) in Hindi, (mōhanēna) in Sanskrit, and (mōhananinda) in Kannada, all have the same semantic structure: a nominal root, followed by a case-marker.

Following are our contributions:

- We propose Indian-language-specific word-grouping criteria to make the tokens semantically coherent.
- We propose a rule-based method to perform the word-grouping task, by generating rules using a combination of data statistics and linguistically educated decisions.
- We assess the importance of word-grouping qualitatively as well as quantitatively through intrinsic and extrinsic evaluation methods respectively.

2 Related Works

2.1 Text Chunking vs word-grouping

Chunking is an important preprocessing step in several NLP tasks, and is considered especially useful as a precursor for dependency parsing task (Abney, 2022). Other downstream tasks for which chunking plays an important role include Named Entity Recognition (Zhou and Su, 2002), information extraction (Dong et al., 2023), etc. Works on machine learning-based text chunking have been around for several decades (Church, 1988; Ramshaw and Marcus, 1999). Most of these works are based on English or related languages, and the most widely adopted granularity for chunking a sentence is phrases (noun and verb phrases). Bharati et al. (1991) showed that the concept of phrasal chunks is not natural for Indian languages and the necessity for *Local word-grouping (LWG)* instead. We extend the concept of LWG by defining a word-group to be the smallest indivisible, semantically complete and meaningful unit of a sentence expressing a single linguistic function (known in Indian linguistic tradition as "ēkārthībhāva" and "sāmarthya"). The concept of Multi-Word Expressions (Otani et al., 2020) aligns with the word-grouping approach discussed in Section 3. This concept is even more pronounced in the Paninian Grammatical concept of a 'pada' as emphasised by Bharati et al. (2015) and Dangarikar et al. (2024).

2.2 Unfairness in Tokenization

Tokenization is a standard preprocessing step in NLP tasks, where a given input is broken into the smallest units for a system to process. Prior works (Petrov et al., 2024; Ahia et al., 2023) have shown the unfairness that arises in Language Models due to large variations in number of tokens associated with the same semantic content for different languages. In addition to this, we also see a language-dependent imbalance caused by the variation in the number of words used to convey the same concept in different languages (Dangarikar et al., 2024), which is not addressed in these works.

Generally, we consider the space-separated sequence of characters as a word. Considering the diversity of languages, the semantic information present in each word differs significantly. An isolating language like Chinese involves close to just one morpheme per word whereas, an agglutinated language like Malayalam has words that include multiple morphemes added sequentially, with various morphological information, such as gender, tense, person, etc., fused with the word in the form of affixes. Such a variation exist among multiple Indian languages too[3].

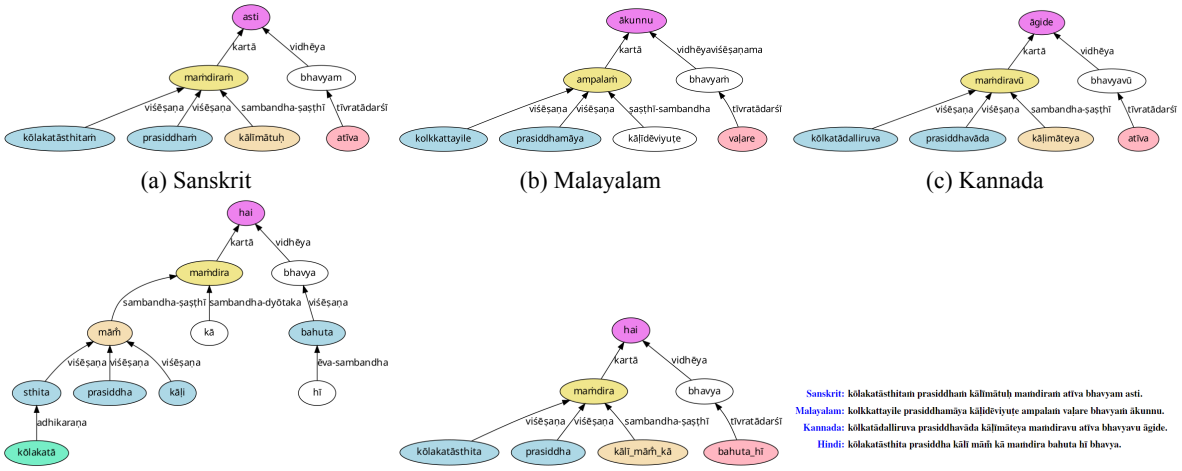
3 Methodology

Following are the basic rules followed in our proposal for grouping of words in accordance with Dangarikar et al. (2024). The method is loosely based on the principle of samāsa in Sanskrit grammar.

- **Inflectional unity**: grouping nouns followed by post-positions, which are essentially the inflectional morphemes.
  Example groups: राम ने (rāma nē), हाथ से (hātha sē), बच्चों को (baccōm kō)
- **Derivational unity**: grouping verb and auxiliary verbs of a sentence, resulting in a single and complete action.
  Example groups: जा रहा है (jā rahā hai), कर दिया गया (kara diyā gayā)

[2]ISO15919 Indic Transliteration scheme

[3]We add statistics for the major languages in Appendix A



(a) Sanskrit (b) Malayalam (c) Kannada

(d) Hindi (without grouping) (e) Hindi (word grouped) (f) Parallel sentences

Figure 2: Dependency Parse Trees

- **Named entities**: A named entity (NE) with multiple words form a single group.
  Examples: ए पी जे अब्दुल कलाम (ē pī jē abdula kalāma), अरुणाचल प्रदेश (aruṇācala pradēśa)

3.1 Similar Syntactic Structures

A dependency parse tree is a syntactic structural representation of a sentence, where the words or phrases form the nodes, and the edges show the dependencies between the nodes and their syntactic roles in the sentence.

Figure 2 shows dependency parse trees[4] of the sentences in Figure 2f, following (Dangarikar et al., 2024). Each word in the sentence forms a node in the tree. Figure 2e is the dependency parse tree for the Hindi sentence obtained after word-grouping. Notice the similarity in the parse trees structure after grouping, while the structure of Hindi example was very different from others before word-grouping, as seen in Figure 2d.

Rule-Based word-grouping

In this section, we present the process followed to automatically generate word groups for Hindi data. The method used, though atypical in nature, generates good quality grouped data for Hindi.

We used (Kosaraju et al., 2012) data with kāraka-based dependency tags (Tandon et al., 2016), a widely used treebank dataset for Hindi, to statistically generate rules for word-grouping.[5] The rules are generated from the dataset by finding the most frequent dependency relations between consecutive words in the sentence, along with the respective frequent POS tags of the tokens. We finalized the rules after verification by language-experts.

[4]Drawn using anvaya chitranam

[5]Generated rules are added in the Appendix A

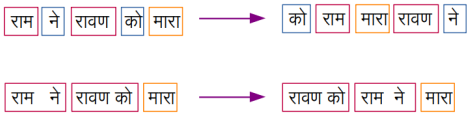


Figure 3: (Top) words are randomly jumbled (Bottom) jumbled sentence with word-groups preserved

4 Experiments and Results

4.1 Sentence Perturbation

To justify the requirement for word-grouping, we design an intrinsic evaluation method using sentence perturbation by jumbling words, a commonly adopted method to evaluate representational co-relatedness in sentences (Alleman et al., 2021; Sai et al., 2021). Our hypothesis is that word-grouping allows sentences to preserve semantic roles/identities of its components, even on random shuffling. The hypothesis also aligns with the relatively free-word-order nature of Indian languages.

In the experiment setting (i) We perform random shuffling of the space-separated words vs the word-groups across the complete sentence. A simple method of randomly jumbling words has a drawback when the sentences under consideration are long and complex, containing multiple clauses. Each clause may contain its own set of subject, object, verb, etc., and meanings may not be preserved despite the word groups being preserved when the words or groups are jumbled across multiple clauses. To address this issue, we consider setting (ii) local shuffling of word units within the extracted clauses, to ensure that the context is preserved. We follow Sharma et al. (2013) to extract

Setting	Grouped / Not	Languages							
		Hindi	Kannada	Malayalam	Sanskrit	Tamil	Telugu	Bangla	Marathi
(i)	Ungrouped	0.867	0.705	0.716	0.681	0.695	0.693	0.718	0.703
	Randomly Grouped	0.898	0.716	0.729	0.680	0.706	0.706	0.731	0.713
	Word-Grouped	**0.899**	**0.719**	**0.731**	**0.685**	**0.711**	**0.710**	**0.732**	**0.719**
(ii)	Ungrouped	0.886	0.713	0.723	**0.683**	0.701	0.701	0.726	0.712
	Randomly Grouped	0.876	0.703	0.714	0.660	0.691	0.691	0.716	0.694
	Word-Grouped	**0.912**	**0.718**	**0.732**	0.677	**0.708**	**0.710**	**0.733**	**0.719**

Table 1: Cosine similarity between shuffled Hindi sentences with the parallel sentences in other Indian languages. In each setting, shuffling is done with and without the preserved word-groups. (i) Shuffling entire sentence (ii) Local Shuffling within clauses. Experiments are also performed with random word-grouping (termed Randomly Grouped) without considering the proposed grouping criteria.

Languages	DecoMT w/o grouping		DecoMT with grouping	
Source → Target	spBLEU	chrF++	spBLEU	chrF++
Hindi → Malayalam	18.9	36.87	**19.4**	**37.29**
Hindi → Kannada	19.2	37.55	**19.5**	**38.21**
Hindi → Sanskrit	4.3	19.34	**4.7**	**20.77**
Hindi → Bengali	19.3	36.35	**19.6**	**36.69**
Hindi → Marathi	14.0	35.34	**14.6**	**35.96**

Table 2: Performance comparison of DecoMT (Few-Shot) preserving word groups, against the baseline that use fixed-length chunks.

the clauses within the sentences. The total number of identified clauses from the data is 2146.

It is possible that randomly shuffling a sentence disrupt all word relationships, significantly reducing its semantic coherence compared to shuffling the sentence while preserving some fixed adjacent words, which may still retain some local relationships and semantic similarity to the original sentence. To verify the significance of the proposed word-group, we also perform the jumbling of sentences by randomly grouping adjacent words in the sentence and measure the corresponding similarities. We generate sentence embeddings for both sets of sentences using Sentence-BERT(Reimers, 2019) and cosine similarity to measure the similarity of the jumbled sentences with both the original unshuffled sentence and parallel sentences in 7 languages (Table 1). The table shows that in general, jumbled sentences with our proposed word-groups preserved, has the most similarity with the original sentences compared to randomly jumbled sentences.

4.2 Decomposed Few-Shot Prompting

Puduppully et al. (2023) have shown improvements in MT task through few-shot prompting between related languages using decomposed prompts (DecoMT). They use mT5-XL(Xue et al., 2021) with 3.7B parameters as the base model for their experiments. DecoMT performs a combination of chunk-based translation and an iterative contextual translation and learns a combined loss. Input sentences are segmented by splitting them as fixed size word chunks. We observe that this method causes a meaningful semantic unit to be split across segments, which may ultimately reduce the translation quality, especially at the segment level. Considering word groups as a single unit prevents this split into different decompositions within a prompt. With this hypothesis, we perform experiments with DecoMT using FLORES-200 data (Costa-jussà et al., 2022), with the chunks as in (Puduppully et al., 2023), versus the chunks with word groups preserved. We perform the experiments to translate sentences from Hindi to 5 other languages. Table 2 shows consistent improvement on preserving word groups.

5 Conclusions

We propose the grouping of whitespace-separated words based on semantics, as a major preprocessing step for any computational and linguistic processing of sentences. Given the least agglutinative nature of Hindi compared to other Indian languages, we focus our experiments on Hindi, expecting it to benefit the most from grouping. We perform an intrinsic experiment, and an extrinsic experiment on MT. From both sets of experiment results, it is evident that a word group as a single semantic unit of a sentence provides a consistent improvement across experiments.

Limitations

The process used for automatic word-grouping is not straightforward. Though it generated a good quality word grouped data, it involves a dependence on another deep learning model. There is a need

to have a more explicit method to generate grouping rules for automatic word-grouping. For highly agglutinated languages, more than grouping, there may be a requirement to split a space-separated word into constituents, which we do not do in this work.

We intend to further simplify and facilitate the process of automatic word-grouping in Hindi and also extend the grouping process (also splitting where necessary) to other languages.

References

Steven P Abney. 2022. Principle-based parsing. In *12th Annual Conference. CSS Pod*, pages 1021–1021. Psychology Press.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

Matteo Alleman, Jonathan Mamou, Miguel A Del Rio, Hanlin Tang, Yoon Kim, and SueYeon Chung. 2021. Syntactic perturbations reveal representational correlates of hierarchical phrase structure in pretrained language models. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 263–276, Online. Association for Computational Linguistics.

Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. 1991. Local word grouping and its relevance to indian languages. *Frontiers in Knowledge Based Computing (KBCS90), VP Bhatkar and KM Rege (eds.), Narosa Publishing House, New Delhi*, pages 277–296.

Akshar Bharati, Sukhada, Prajna Jha, Soma Paul, and Dipti M Sharma. 2015. Applying Sanskrit concepts for reordering in MT. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 357–366, Trivandrum, India. NLP Association of India.

Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, USA. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Chaitali Dangarikar, Arnab Bhattacharya, N J Karthika, Hrishikesh Terdalkar, Pramit Bhattacharyya, Annarao Kulkarni, Chaitanya S Lakkundi, Ganesh Ramakrishnan, and Shivani V. 2024. *Samanvaya: An Interlingua for the Unity of Indian Languages*. Central Sanskrit University.

Kuicai Dong, Aixin Sun, Jung-jae Kim, and Xiaoli Li. 2023. Open information extraction via chunks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15390–15404, Singapore. Association for Computational Linguistics.

Daniela Gerz, Ivan Vulić, Edoardo Ponti, Jason Naradowsky, Roi Reichart, and Anna Korhonen. 2018. Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction. *Transactions of the Association for Computational Linguistics*, 6:451–465.

Prudhvi Kosaraju, Bharat Ram Ambati, Samar Husain, Dipti Misra Sharma, and Rajeev Sangal. 2012. Intra-chunk dependency annotation : Expanding Hindi inter-chunk annotated treebank. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 49–56, Jeju, Republic of Korea. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A lightweight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Naoki Otani, Satoru Ozaki, Xingyuan Zhao, Yucen Li, Micaelah St Johns, and Lori Levin. 2020. Pre-tokenization of multi-word expressions in cross-lingual word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4451–4464, Online. Association for Computational Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2024. Language model tokenizers introduce unfairness between languages. *Advances in Neural Information Processing Systems*, 36.

Prakash B. Pimpale, Raj Nath Patel, and Sasikumar M. 2014. SMT from agglutinative languages: Use of suffix separation and word splitting. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 2–10, Goa, India. NLP Association of India.

Ratish Puduppully, Anoop Kunchukuttan, Raj Dabre, Ai Ti Aw, and Nancy Chen. 2023. DecoMT: Decomposed prompting for machine translation between related languages using large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4586–4602, Singapore. Association for Computational Linguistics.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ananya B. Sai, Tanay Dixit, Dev Yashpal Sheth, Sreyas Mohan, and Mitesh M. Khapra. 2021. Perturbation CheckLists for evaluating NLG evaluation metrics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7219–7234, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rahul Sharma, Soma Paul, Riyaz Ahmad Bhat, and Sambhav Jain. 2013. Automatic clause boundary annotation in the hindi treebank. In *Proceedings of the 27th Pacific Asia Conference on Language, Information and Computation*, pages 499–504. Waseda University.

Juhi Tandon, Himani Chaudhry, Riyaz Ahmad Bhat, and Dipti Sharma. 2016. Conversion from Paninian karakas to Universal Dependencies for Hindi dependency treebank. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 141–150, Berlin, Germany. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an HMM-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

A Appendix

A.1 Word count imbalances across languages

Generally, we consider the space-separated sequence of characters as a word. Considering the diversity of languages, the semantic information present in each word differs significantly. An isolating language like Chinese involves close to just one morpheme per word. In contrast, an agglutinated language like Malayalam has words that include multiple morphemes added sequentially, with various morphological information, such as gender, tense, person, etc., fused with the word in the form of inflectional affixes[6].

Figure 4 shows the total number of words in different Indian languages for the parallel sentences, representing the same content in FLORES-200 devtest data (Costa-jussà et al., 2022). The graph also contains some non-Indian languages to show the extent to which the number of words can vary across languages to convey the same information. Note that, the number of words in Jingpho is over 3.6× the number of words in the corresponding parallel data in 'Shan'. In cross-lingual tasks involving languages exhibiting such variations, the associated models are responsible for implicitly learning the semantic unit-level mapping in addition to the underlying task, adding to the model complexity.

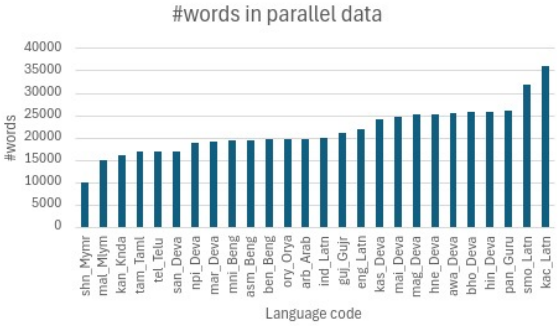


Figure 4: Number of space-separated words present in the parallel sentences of a subset of languages in FLORES-200 devtest data

Table 3 shows the word count of parallel sentences from a commonly used benchmark data for MT evaluation, *viz.,* FLORES-200 (Costa-jussà et al., 2022) devtest data. From the table, it is evident that, out of the six languages considered(Malayalam, Kannada, Sanskrit, Marathi, Bengali, and Hindi), Hindi exhibits a significant deviation from the rest. Data statistics provided by Gerz et al. (2018) using Polyglot Wikipedia, also show a similar trend. The reason for such a deviation is that among the Indian languages considered, Hindi is the least agglutinative in nature(Pimpale et al., 2014), and at times follow isolating features. In this paper, we particularly focus on grouping of words in Hindi, due to this distinctive feature that makes it to deviate the most from other languages. In light of such a deviation, we expect the grouping effort to be more crucial and effective for Hindi.

[6]The words case-marker, vibhakti-marker, inflectional affixes and post-positions are used interchangeably in the paper

Language	Total #words
Malayalam	15001
Kannada	16577
Sanskrit	16992
Marathi	19046
Bengali	19585
Hindi (without grouping)	25643
Hindi (grouped)	18980

Table 3: Total number of whitespace separated words (or semantic units) in FLORES-200 devtest data. Hindi, when grouped, has a count similar to other Indian languages.

A.2 Rules for Automatic word-grouping

Table 4 shows the rules used to perform automatic word-grouping for sentences. The rules are generated by a combination of treebank data statistics and feedback from linguists. The sentences are first input to trankit (Nguyen et al., 2021) tool to generate the corresponding values as shown in the table fields. This step is followed by the application of rules for grouping.

A.3 Scores of Translation across different Sentence-Lengths

The DecoMT approach translates source sentences in sequential chunks, and we hypothesize that integrating word-grouping will enhance translation adequacy, as it ensures that meaningful semantic units remain intact within chunks rather than being split across them. To investigate this, following a method similar to Puduppully et al. (2023), we categorize source sentences into buckets based on length, where each bucket's width corresponds to the standard deviation of the source sentence lengths. Buckets with fewer than 20 instances are merged with neighbouring ones. Figure 7 illustrates the relationship between source sentence length and chrF++ scores for translations from Hindi to Malayalam, Kannada, Sanskrit, Bengali, and Marathi. For all language pairs, we observe that DecomMT with word-grouping consistently outperforms DecoMT in terms of chrF++ scores.

A.4 Dataset and Evaluation Metrics

For both sets of experiments (Sentence Perturbation and DecoMT translation evaluation), we utilise a commonly used Benchmark dataset for Machine Translation Evaluation, FLORES-200 (Costa-jussà et al., 2022), specifically the dev-test data, which has 1012 sentences, parallelly available in 200 languages.

In the experiments discussed in Section 4.2, we perform the experiments to translate sentences from Hindi to five other languages *viz.,* Malayalam, Kannada, Sanskrit, Bengali and Marathi. We used BLEU and chrF++ scores to present the results. The specific signatures used for the metrics are ***BLEU Signature:*** nrefs:1| case:mixed| eff:no| tok:flores200| smooth:exp| version:2.4.2 and ***chrF++ Signature:*** nrefs:1| case:mixed| eff:yes| nc:6| nw:2| space:no| version:2.3.1.

Note: For MT experiments, we chose the three target languages, which are agglutinative in nature, with the intuition that the splitting at a sub-group level in Hindi sentences may cause decline in quality because of non-splittable words in these languages.

Dependent token		Head		Dependency relation	Examples
Category	**POS tag**	**Category**	**POS tag**	-	-
pn/n/v	NN/NNP/VM/PRP/PSP	psp	PSP	lwg_psp	हॉल(h)-में(d) ; जाने-के-लिए ; लगने-वाला
v	VAUX/VM	v	VAUX	lwg_vaux_cont	जाता-था ; रहता-है ; हुआ-है ; गए-है
v	VM	v	VAUX	lwg_vaux	बसाया-गया ; कहती-है ; बनवाया-था; बनी-हुई देती-है ; लिए-हुए
v	VM	avy	RP	lwg_rp	देखते-ही ; बड़ा-सा; स्थान-में-भी; भीड़ लगी-ही रहती है; कितने-ही; एक-ही
n	NN	v	JJ	pof	बंद-रहता; आनंद-उठा ; स्मृति-दिलाता ; अलग-होना ; कैद-होना
n/adj	NNPC/NNC	any	any	pof_cn	म.प्र.(d)-पर्यटन(h)-बोट(h)-क्लब(h); विमान-सेवा ; 17वीं-शताब्दी; रुद्र-प्रताप
n/adj	NN/NNP	any	any	pof	क्या नहीं-किया ; प्रवेश नही-मिलता

Table 4: Rules used for Automatic word-grouping

Translate from Hindi to Bengali:
Hindi: सोमवार को, स्टैनफ़ोर्ड यूनिवर्सिटी स्कूल
Bengali: সোমবারে স্ট্যানফোর্ড ইউনিভার্সিটি স্কুল
Hindi: ऑफ़ मेडिसिन के वैज्ञानिकों ने
Bengali: অফ মেডিসিনের বিজ্ঞানীরা
Hindi: एक नए डायग्नोस्टिक उपकरण के
Bengali: একটি নতুন ডায়াগনস্টিক উপকরণের
Hindi: आविष्कार की घोषणा की
Bengali: আবিষ্কারের ঘোষণা করেছেন
Hindi: कोशिकाओं को उनके प्रकार के
Bengali: কোষকে তাদের প্রকারের
Hindi: आधार पर छाँट सकने वाले,
Bengali: উপর ভিত্তি করে বাছাইযোগ্য,
Hindi: एक छोटी प्रिंट करने योग्य
Bengali: একটি ছোটো মুদ্রণযোগ্য
Hindi: चिप जिसे स्टैण्डर्ड इंकजेट प्रिंटर
Bengali: চিপ যেটা স্ট্যান্ডার্ড ইঙ্কজেট প্রিন্টারের
Hindi: का उपयोग करके लगभग
Bengali: ব্যবহার করে প্রায়
Hindi: एक अमेरिकी सेंट के लिए
Bengali: এক মার্কিন সেন্টের মধ্যে
Hindi: निर्मित किया जा सकता है।
Bengali: নির্মাণ করা যেতে পারে।

...3 more examples here

Translate from Hindi to Bengali:
Hindi: पायलट की पहचान स्क्वाड्रन लीडर
Bengali: পাইলটের পরিচয় স্কোয়াড্রন লিডার
Hindi: दिलोकृत पटावी के रूप में
Bengali: দিলোকৃত পাতাভির রূপে
Hindi: की गई।
Bengali: করা হয়েছে।

Translate from Hindi to Bengali:
Hindi: जीवित रहने की दर
Bengali: <mask>

Figure 5: DecoMT Prompt Template for Independent Translation with a Test Example: The template includes five sentences in the source (Hindi) and target (Bengali) languages divided into word chunks. The model receives a test example source chunk and a target language prompt with a <mask> placeholder, aiming to predict the corresponding target chunk

Translate from Hindi to Bengali:
Hindi: सोमवार को, स्टैनफ़ोर्ड यूनिवर्सिटी स्कूल ऑफ़ मेडिसिन के
Bengali: সোমবারে স্ট্যানফোর্ড ইউনিভার্সিটি স্কুল অফ মেডিসিনের
Hindi: वैज्ञानिकों ने एक नए डायग्नोस्टिक उपकरण के
Bengali: বিজ্ঞানীরা একটি নতুন ডায়াগনস্টিক উপকরণের
Hindi: आविष्कार की घोषणा की
Bengali: আবিষ্কারের ঘোষণা করেছেন
Hindi: कोशिकाओं को उनके प्रकार के आधार पर
Bengali: কোষকে তাদের প্রকারের উপর ভিত্তি করে
Hindi: छाँट सकने वाले, एक छोटी प्रिंट करने योग्य
Bengali: বাছাইযোগ্য, একটি ছোটো মুদ্রণযোগ্য
Hindi: चिप जिसे स्टैण्डर्ड इंकजेट प्रिंटर का उपयोग करके
Bengali: চিপ যেটা স্ট্যান্ডার্ড ইঙ্কজেট প্রিন্টারের ব্যবহার করে
Hindi: लगभग एक अमेरिकी सेंट के लिए
Bengali: প্রায় এক মার্কিন সেন্টের মধ্যে
Hindi: निर्मित किया जा सकता है।
Bengali: নির্মাণ করা যেতে পারে।

...3 more examples here

Translate from Hindi to Bengali:
Hindi: पायलट की पहचान स्क्वाड्रन लीडर
Bengali: পাইলটের পরিচয় স্কোয়াড্রন লিডার
Hindi: दिलोकृत पटावी के रूप में की गई।
Bengali: দিলোকৃত পাতাভির রূপে করা হয়েছে।

Translate from Hindi to Bengali:
Hindi: जीवित रहने की दर आधी हो सकती है.
Bengali: <mask>

Figure 6: Proposed Prompt Template for Independent Translation with a Test Example: The template includes five sentences in the source (Hindi) and target (Bengali) languages divided into word chunks according to the word-groupings. The model receives a test example source chunk and a target language prompt with a <mask> placeholder, aiming to predict the corresponding target chunk

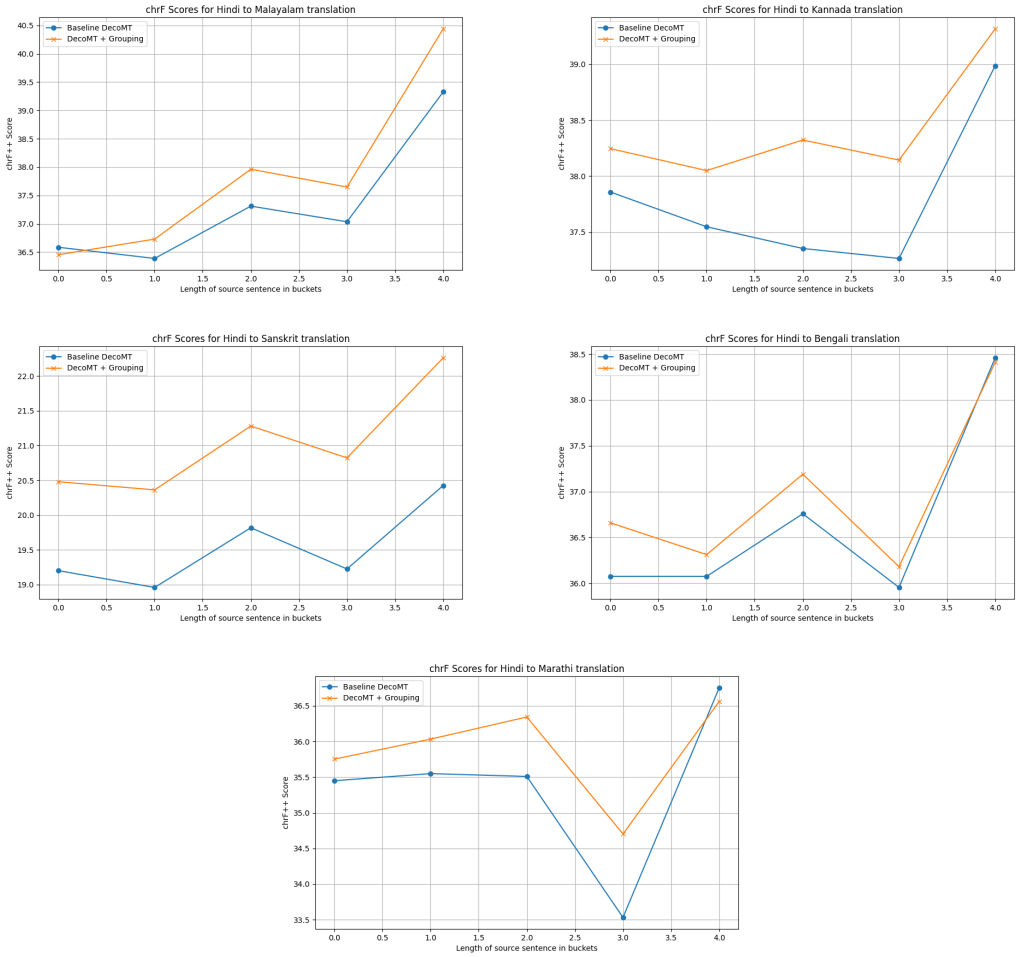


Figure 7: The plots show the relationship between source sentence length and chrF++ scores for translation from Hindi to Malayalam, Kannada, Sanskrit, Bengali, and Marathi. Lengths are bucketed, each equal to the source sentence lengths' standard deviation, with any bucket with less than 20 sentences merged with its neighbor. The data implies the chrF++ scores of DecoMT combined with our grouping outperform baseline DecoMT's performance.