# The Mangalam Dictionary of Buddhist Sanskrit: automating lexicographic data with generative LLMs

**Ligeia Lugli**[1]

[1]Mangalam Research Center, 2018 Allston Way, Berkeley (CA) USA.
E-mail: ligeia.lugli@london.ac.uk

**Abstract**

This paper reports on recent advancements in the development of the *Mangalam Dictionary of Buddhist Sanskrit*, the first corpus-driven dictionary dedicated to Buddhist Sanskrit. This is a low-resource, historical, and domain-specific language variety instantiated in South Asian Buddhist literature dating from approximately the first millennium CE. The paper focusses on advances in the automation of this dictionary's data with generative Large Language Models (LLMs), with a view to share our solutions with scholars working with other low-resource historical languages. Specific doomed to fail ally, the paper addresses the effectiveness and viability of leveraging latest generation LLMs to automate three tasks that are central to our lexicographic work: semantic annotation of corpus sentences, identification of a headword's semantic prosody in different contexts, and comparison of a headword's synonyms. The paper first evaluates the relative performance of different commercially available models (including GPT 4.1, Sonnet4 and Gemini 2.5) on a semantic tagging task and then details different approaches we experimented with for enriching our corpus with word-sense and semantic prosody tags using LLMs. It concludes with a brief discussion of commercial LLMs' ability to compare Sanskrit synonyms on the basis of corpus sentences.

**Keywords:** Buddhist Sanskrit; generative LLMs; semantic tagging; historical corpora

## 1. Semantically annotated corpora for lexicography

This paper reports on recent experiments on the use of Large Language Models (LLMs) for semantically tagging a corpus of Buddhist Sanskrit literature dating approximately from the II century BC to the XII century CE (bit.ly/MangalamCorpusOfBuddhist-Sanskrit). This corpus was created specifically for lexicographic purposes, with a view to enable the development of the *Mangalam Dictionary of Buddhist Sanskrit,* the first corpus-driven dictionary for this language variety (bit.ly/VisualDictionary-BuddhistSanskrit and mangalamresearch.shinyapps.io/MangalamDictionaryOfBuddhistSanskrit).

'Buddhist Sanskrit' is intended here as the domain-specific type of Sanskrit attested in historical Buddhist sources. This differs from classical Sanskrit mainly in vocabulary and semantics, but often also in syntax and morphology. Since Buddhist Sanskrit constitutes an extremely low-resource language, we have designed our lexicographic workflow to maximize the re-usability output for linguistic and natural language

processing purposes. To this end, we encode highly curated linguistic information directly in our corpus, and we then automatically derive our dictionary entries from those annotations (Lugli, 2021a). Our corpus annotations include, among other features, word-senses (annotated with English sense labels), and semantic prosody, intended here as contextually instantiated connotation (Stewart, 2010; Guo et al., 2011).

Annotating this information manually is extremely time consuming, as it entails reading substantial portions of text, which are typically difficult to interpret due to the highly specialized and often philosophically intricate nature of the material. This has very practical implications for our work, as it severely limits the amount of data we can manually annotate within the timeframe of our project, and this, in turn, limits the scope and usefulness of our lexicographic output. Typically, we can manually annotate a maximum of two hundred sentences per headword. Even though we carefully sample these sentences to make our curated data as representative as possible, we cannot be sure that our sample captures all the senses of a headword. Moreover, while we can analyze the distribution of a headword and its collocates across the corpus, we cannot confidently portray the distribution of its senses, as only a minority of word-senses are closely associated with collocational patterns. Similarly, we cannot offer much insight on how the lexicalization of concepts changes over time, or across genres, as the scale of polysemy in Sanskrit makes it virtually impossible to determine when the words included in the lexical field of a concept actually refer to it. The word *kalpa* for example predominantly lexicalizes the concept of aeon, but it also frequently refers to the idea of conceptualization, or mental construct, and can also denote a rule, an argument or line of reasoning, a mythical tree, likeness and propriety. Of these, only the sense of mythical tree is consistently tied to collocations. Any attempt to trace the evolution of the concept of aeon by analyzing the distribution of the word *kalpa* is therefore doomed to fail.

What is necessary to pursue such line of enquiry, is a fully semantically annotated corpus. This has been a great desideratum for Sanskrit research and lexicography for a long time. Our project in particular has been very keen to develop such a corpus, since automating semantic annotations would significantly speed up our work (see section 4). Yet, until recently creating such a corpus seemed an unattainable goal, as semantically tagging corpora of ancient languages is a notoriously difficult task, even for languages much better resourced than Buddhist Sanskrit (McGallivray et al., 2022; Vatri and McGallivray, 2018).

Within the purview of our lexicographic project, we had made some progress in this direction in the past, using BERT word embeddings (Lugli et al., 2022). But our results, while promising, were not deemed sufficiently accurate for semantical tagging (Martinc et al., 2023). At the time of our BERT experiments, we also tried using ChatGPT (version 3.5) for the task, as this tool was then beginning to gain traction. This, too, did not yield usable results, as the model appeared extremely skewed towards

the most frequent, non-Buddhist, meanings of Sanskrit words and failed to capture the semantics of Buddhist vocabulary (Lugli et al., 2023). Fortunately, in the intervening years the situation has drastically changed and commercial generative Large Language Models have now become much better at processing Buddhist Sanskrit. So, while achieving the degree of accuracy and delicacy required for semantically tagging our corpus remains difficult, this endeavor appears now achievable. We share here some encouraging results we have obtained in our first experiments with commercial AI for corpus tagging, with a view to help scholars working with other historical low-resource languages shape their strategies in the use of generative LLMs for semantic tagging and lexicography. Specifically, we first report on our experiments in tagging word-senses, then we discuss our attempts to tag semantic prosody, and finally we briefly discuss our preliminary work on adopting AI for synonyms comparison.

## 2. Using LLMs for semantic tagging

### 2.1 Word-sense discrimination vs word-sense induction

Semantically tagging a corpus basically consists in assigning a sense label to the words in the corpus. Before we could set out to use LLMs for this task, we had to decide how to obtain the sense labels for each word. More generally, we had first to decide whether to opt for word-sense disambiguation and give the LLM a predefined set of senses to choose from, or opt for word-sense induction and ask the model to come up with its own sense labels based on how each word is used in the corpus.

Word-sense induction has the great advantage of leaving open the possibility of discovering new senses. The flipside is that the results are often hard to interpret and evaluate, as this approach may lead to a proliferation of senses that do not map well onto gold-standard datasets of manually tagged sentences. An unwieldy proliferation of senses was our experience when we tried word-sense induction in the context of our BERT experiments (Martinc et al., 2023), so this time we focussed on word-sense disambiguation instead.

Still, we did tentatively experiment with word-sense induction and our preliminary results held some promise. We tried this approach using Anthropic's Sonnet 4 on two polysemic words that are very frequent in our corpus, *dharma* and *dhātu*. In both cases we gave the model a set of about one hundred Sanskrit sentences instantiating the headword in various senses, as well as their translation in English (more on the use of translations later). We then prompted the model to tag the sentences in this way: "First, look at all the sentences and determine the range of meanings that the headword [...] expresses in these sentences. Second, come up with clear English sense-labels for those meanings, try to keep meanings general so that the number of sense-labels are informative and not overly fine-grained. Third, tag each sentence with the

corresponding English sense-label."

For the word *dharma* this led to extremely good results. The model crafted very clear sense-labels that were easy to interpret and also mapped well onto the senses recorded in our manually curated dictionary data, which we used as gold-standard to evaluate the quality of the LLM's output. Moreover, the model assigned the sense tags in a way that matched almost perfectly our gold-standard dataset, achieving very high accuracy. *Dharma*, however, is hardly a representative example, as it is an extraordinarily well-documented word. When we tried the same approach on the less researched word *dhātu*, the results were not quite as neat. In this case, Sonnet 4 crafted a set of sense labels that differed significantly from our gold-standard data, missing some senses and conflating meanings that would be lexicographically useful to keep distinct. For example, it subsumed under the tag 'constituent: psychophysical constituents/components of experience' cases where *dhātu* refers to a person's innate nature, or character, as well as cases where it refers to the object of one's experience, including sense-objects. Similarly, it did not distinguish between the sense 'element', which is used abstractly and metaphysically in the corpus, and the very concrete sense 'metal ore'. Overall, the model's semantic categorisation of *dhātu*, was valid, but missed nuances that are rather crucial for lexicographic and research purposes. So, while there is surely scope for improving the performance of LLMs on word-sense induction tasks, we set this avenue of experimentation aside and concentrated on the simpler problem of word-sense disambiguation.

To this end, we derived a set of predefined senses, or sense inventory, for each of the words that we wanted to annotate. We used English sense labels, as this is aligned with our semantic annotation goal. In the case of words not yet included in our dictionary we used Sonnet 4 to streamline into clear and concise sense labels the rather redundant lists of English equivalents provided in popular Sanskrit dictionaries compiled by Monier-Williams and Edgerton (Monier-Williams, 1899; Edgerton, 1953). For words already included in our dictionary, we derived sense inventories from our dictionary data. This entailed choosing the appropriate level of granularity for the semantic tags, since our dictionary carves a headword's semantic spectrum into broad 'senses' as well as fine-grained 'subsenses'. Senses typically distinguish between very different meanings of a headword. This means that they tend to be easier to identify automatically, but often prove too broad for meaningful application in research. the reverse holds true for subsenses. Choosing between senses and subsenses and finding the right balance between semantic delicacy and tagging accuracy required some trial and error (see sections 2.3 & 4). At first, we started with the simplest setting and used the broader sense labels to test various commercially available LLMs on a semantic tagging task.

### 2.2 Comparison of commercially available models on semantic tagging

In order to estimate which model would be most suitable for our purposes, we compared the performance of some of the leading commercial AI models on a streamlined version of our semantic tagging task. First, we selected twenty polysemic words for which we had already manually annotated many sentences in the course of our dictionary work (*ākhyāta, artha, dharma, dhātu, ghoṣa, jalpa, kalpa, nāman, nimitta, prajñapti, ruta, saṃjñā, saṃketa, smṛti, upacāra, vastu, vikalpa, vyañjana, vyavahāra, śabda*). We then derived from our dictionary basic sense inventories for each of these words. These sense inventories only list English sense labels for the main senses of the words, and do not record the more granular semantic differences that are included in the subsenses our dictionary. For example, the sense inventory for the word *dharma* would only feature four main senses ("cosmic law", "things/phenomena", "quality" and "rightness"), while our dictionary also includes a more nuanced division of *dharma*'s semantic spectrum into eight subsenses ("constituents of reality", "reality/truth", "phenomena", "natural order", "Buddhist doctrine (Dharma)", "quality", "quality/property (in logic)" and "rightness"). Restricting the sense inventories to the main senses of the words was meant to simplify the word-sense disambiguation task. Once we created all the sense inventories, for each word we asked various models to tag a set of Sanskrit sentences that feature that word with the senses taken from our sense inventory for that word. We did not provide the models with any examples, nor with any translation of the sentences. In all cases we set the model's temperature to zero and we opted for the non-thinking version of the model (all the prompts we used for this paper are available on Zenodo, 10.5281/zenodo.17375658). We performed this task with Anthropic's Sonnet 4, Google's Gemini 2.5, OpenAI's GPT 4.1, Perplexity's Sonar Large (which is built on Meta's Llama 3.1 70B) and DeepSeek's R1. R1 systematically timed out on the task and we could not obtain any result from it. Sonnet 4 and Gemini 2.5 performed the best, with Sonnet 4 yielding a better median Cohen's kappa (0.35 versus Gemini 2.5's 0.28) and slightly better percentage of correctly tagged sentences, when averaged across all twenty words (69% versus Gemini 2.5's 67%). GPT4.1 lagged behind, with median kappa of 0.18 and an average of 62% sentences tagged correctly. Sonar performs the worst, as it is to be expected given that this model is optimised for retrieval and summarization of information found on the web and is therefore the least suitable for a tagging task. Based on these results, we adopted Sonnet4 for our further semantic tagging experiments.

| model | correct_avg | correct_range | k_median |
|---|---|---|---|
| **Sonnet4** | 67.9% | 0.111:0.958 | 0.355 |
| **Gemini2.5** | 65.8% | 0.111:0.947 | 0.2824 |
| **GPT4.1** | 62.6% | 0:0.929 | 0.187 |

| | | | |
|---|---|---|---|
| **Sonar** | 49.9% | 0:0.933 | 0.0648 |

Table 1: results of our models' comparison. Given the small size of some of our manually annotated datasets, we have evaluated the AIs' outputs using Leave-One-Out cross validation, instead of the standard five-fold one.

### 2.3 Experiments with semantic tagging

While promising, the simple setting we used for comparing different LLMs is not suitable for application in real world lexicography. First, the accuracy is too low. Second, more fine-grained sense labels are often essential to distinguish important uses of the word. In the case of *dharma*, for example, differentiating between cases where it denotes to the cosmic law that governs reality and cases where such cosmic law refers specifically the content of the Buddhas' teaching helps capture the difference between the use of this word in Buddhist and non-Buddhist material and should thus not be neglected. Hence, a better approach is needed to improve the accuracy and delicacy of automated semantic tagging.

What follows is a report on the avenues that we have tried so far in our search for such a better approach. In a nutshell, we have devised a multi-step tagging workflow that involves two or more iterations of semantic tagging and evaluation, depending on the results. The first step consists of deciding the desirable level of granularity for the semantic tags of a headword, taking into account the lexicographic usefulness of the possible semantic distinctions and the likelihood that such distinctions will be captured well by the LLM. The second step involves prompting the model to semantically tag a set of sentences for that headword on the basis of their English translation, which we provide. This typically achieves much better results than the simple scenario we used to compare the models' performance. Still, this approach only works for portions of the corpus that have been translated and aligned, which are a minority. For the rest of the corpus, we need to introduce a further step. If the results of translation-based tagging are deemed sufficiently good, we use the sentences that the model has tagged so far as examples for the next round of tagging. Here, we provide the model with many examples of semantically tagged Sanskrit sentences, and prompt it to first learn from these examples the contextual patterns that best predict each sense and then tag the remaining corpus sentences accordingly. If the results of the translation-based tagging are not good, we try to improve the quality of the tagging by pairing translations with examples of semantically annotated sentences from our dictionary data. This however has the drawback of requiring a fairly large preexistent dataset of annotated sentences, which we only have for a few words.

The results of each round of tagging vary greatly depending on the word. A more detailed look at our tagging workflow will help make sense of this variation and clarify the strategies we have devised to mitigate it.

In the first tagging round we only use a small portion of our corpus that we have aligned with published English translation at sentence level. We extract from this subcorpus all the sentences instantiating a given headword and provide the model with both the Sanskrit and the English translation of each sentence that it is asked to tag. Specifically, we prompt it to first identify in the translation the English equivalent of the Sanskrit headword to be tagged, and then to choose a semantic tag from our sense inventory on basis of both the English translation equivalent and the Sanskrit sentence. We also ask the model to indicate its confidence in each tag. We have tried this approach on a set of 22 words and on average 77% of the sentences were correctly tagged (median kappa 0.62). This means that the tagged datasets were usable for lexicography in most cases. Still, about a third of the sets were not tagged sufficiently well for our purposes (average accuracy below 70% and median kappa below 0.4).

Poor results depended mostly on three factors: the quality of the translations, the way our sense labels are worded and how different are the senses that have to be tagged. We use good quality published translations for this task, but even those are not without problems. Frequently, english translations of Buddhist Sanskrit texts mask the polysemy of a word with overly consistent renditions (Lugli, 2021b). For example, *saṃjñā* is overly rendered with 'conception', regardless of whether it refers to a metaphysical entity (the *saṃjñā-skandha*), a cognitive process or a concept. Similarly, *dhātu*'s English equivalent 'element' is used to translate sentences that are annotated with five different senses in our dictionary data ('foundation of perception', 'element', 'character', 'metal ore' and 'world).

When using translations yielded poor results, we attempted an alternative approach. We exposed the model to examples taken from our manually curated dictionary data. The amount of curated data available for each word varies, on average we used 35 examples per headwords. Each example includes a Sanskrit sentence, its translation and a sense tag. We prompted the model to use the Sanskrit examples to identify the lexical or syntactic patterns that best predict each sense and then to tag the sentences accordingly. When no clear pattern could be identified for a sense, we instructed the model to rely the translation for tagging.

In several cases this approach led to substantial improvements. For example, the tagging accuracy of *dhātu*, whose translation is problematic, increased from 49% to 89%, with Cohen's kappa raising from 0.27 to 0.83. Improvements are registered also with words whose senses are closely associated with specific contextual patterns. For example, the tagging accuracy for the word *artha*, whose most frequent use is linked to distinctive syntactic construction, raised from 62% to 82% (kappa from 0.51 to 0.74).

Yet, in some cases this strategy alone did not drastically improve results. In these cases, we revised our sense inventories. We made three types of revisions. First, we rendered some sense labels more descriptive, while still keeping them concise. For

example, for the word *saṃjñā* we reworded the sense 'regard as' into 'regard as/interpret something as something else' and the sense 'consciousness' into 'consciousness/being conscious/being aware of'. Second, for senses strongly associated with specific constructions, we added details or examples of the main constructions they participate in. For the word *nāman*, for instance, we changed the sense label 'namely' into 'namely (with *tad yathā api*)' and for the word *pada* we added to the sense label 'site/abode' an example of the most frequent compound that instantiate this sense, changing the label into 'site/abode (e.g. *jana-pada*)'. Finally, in one case, we changed our sense categorization altogether and merged two similar senses into one. This happened for the word *prajñapti*, which in our dictionary has two main senses that are too broad for meaningful semantic disambiguation, and six subsenses that are too fine-grained for accurate tagging. We therefore merged into a single label two subsenses, "designation", which we use for cases where the word denotes the name of something, and "verbal expression/notion", which we use for cases where the *prajñapti* expresses the more general meaning of something that is expressed or thought of verbally (more on this in section 4).

After revising our sense inventories and adding tagged examples from our dictionary to our prompts when necessary, we obtained usable results for almost all words, with an average accuracy of over 80%.

| **lemma** | **approach** | **Correct** | **k** | **p_value** | **observations** |
|---|---|---|---|---|---|
| **ākhyāta** | translation | 100% | 1.000 | 8.34E-10 | 25 |
| **vyañjana** | translation | 98.2% | 0.940 | 0E+00 | 56 |
| **upacāra** | translation | 96.3% | 0.927 | 2.29E-12 | 54 |
| **kalpa** | translation | 96.1% | 0.914 | 0E+00 | 153 |
| **vikalpa** | translation | 94.3% | 0.660 | 0E+00 | 194 |
| **jalpa** | translation | 92.3% | 0.469 | 2.09E-03 | 26 |
| **dhātu** | translation + examples | 92% | 0.876 | 0E+00 | 50 |
| **prajñapti** | translation | 90.4% | 0.820 | 0E+00 | 156 |
| **smṛti** | translation | 89.4% | 0.732 | 6.66E-16 | 113 |
| **akṣara** | translation | 86.7% | 0.595 | 1.18E-02 | 15 |
| **nimitta** | translation + examples | 86.7% | 0.748 | 2.05E-07 | 30 |
| **nāman** | translation | 85.1% | 0.729 | 0E+00 | 74 |
| **vacana** | translation | 83.3% | 0.747 | 8.04E-09 | 24 |
| **śabda** | translation | 82.5% | 0.679 | 4.61E-07 | 40 |

| | | | | | |
|---|---|---|---|---|---|
| **dharma** | translation | 78.9% | 0.690 | 0E+00 | 114 |
| **artha** | translation + examples | 78.4% | 0.668 | 1.93E-14 | 51 |
| **saṃjñā** | translation + examples | 76.5% | 0.701 | 0E+00 | 34 |
| **saṃketa** | translation | 76% | 0.148 | 3.69E-01 | 25 |
| **vyavahāra** | translation | 74.1% | 0.447 | 0E+00 | 259 |
| **ghoṣa** | translation | 73.3% | 0.459 | 2.28E-06 | 86 |
| **pada** | translation + examples | 70.6% | 0.640 | 4.42E-09 | 17 |
| **ruta** | translation | 63% | 0.400 | 2.06E-04 | 27 |
| **vastu** | translation | 58.5% | 0.343 | 5.4E-13 | 118 |

Table 2: best results of translation and translation cum examples approaches

We then used the automatically tagged sentences thus obtained to tag the sentences for which we do not have aligned translations. So far, we have only tagged 13 words in this way, and the results are generally good, with average accuracy of 80%.

| **lemma** | **Correct** | **k** | **p__value** | **observations** |
|---|---|---|---|---|
| **ākhyāta** | 100% | 1.000 | 4.31E-05 | 11 |
| **upacāra** | 100% | 1.000 | 2.83E-08 | 24 |
| **vyañjana** | 97.1% | 0.910 | 2.77E-11 | 34 |
| **nāman** | 91.7% | 0.850 | 7.95E-09 | 24 |
| **pada** | 87.1% | 0.827 | 0E+00 | 31 |
| **prajñapti** | 86.7% | 0.729 | 0E+00 | 128 |
| **ghoṣa** | 80% | 0.643 | 1.35E-03 | 15 |
| **akṣara** | 78.6% | 0.276 | 2.87E-01 | 14 |
| **śabda** | 77.8% | 0.607 | 1.62E-03 | 18 |
| **vacana** | 70% | 0.524 | 2.97E-05 | 20 |
| **saṃjñā** | 64.8% | 0.534 | 0E+00 | 108 |
| **ruta** | 60.6% | 0.376 | 2.9E-04 | 33 |
| **vyavahāra** | 57.6% | 0.179 | 8.75E-09 | 170 |

Table 3: results using AI-tagged examples

In most cases, however, the quality of the tagging is lower than in the previous round (it is important to note that we did not conduct our evaluation on the same set of sentences, as we only tagged without using translations sentences for which an aligned translation is not available; hence the datasets tagged with and without translations have no overlap). This is perhaps due to the noise from the wrongly labelled sentences that are inevitable included in the automatically tagged examples we use at this stage.

Finding a viable way to weed out wrongly labelled sentences from the examples fed to the model in the last round of tagging is therefore desirable. So far, we have experimented with using the model stated confidence in its own tags to this end. When tagging sentences, we ask the model to assign a value between 0 and 1 describing how confident it is that the tag is correct. We then sample the examples to be used for tagging untranslated sentences from the sentences with high confidence scores. However, all the models we tried proved overconfident, making this an unreliable solution. While on average the mean confidence of correctly tagged sentences is higher than the mean confidence of wrongly tagged sentences (0.86 vs 0.82), the difference is small and many wrong sentences are assigned a confidence of 1. A more promising approach is to tag the same set of sentences with different models and take the sentences that are tagged in the same way by both. We tried this on a set of sentences that were very poorly tagged due to low translation quality and, while the accuracy of the tagging by each model individually (Sonnet 4 and Gemini 2.5) was about 40%, taking only the sentences where both models agreed raised it to 78%. This approach, however, is not financially viable for our project, as processing the same data multiple times with commercial LLMs is costly. Therefore, we have not pursued this avenue further. A potentially more viable approach is to only use as examples for further tagging rounds sentences whose semantic tags have been manually proofread. We have not tried this yet, due to imbalances in our manually curated dataset. For many words, we only have few manually annotated examples for rare senses (often just a single one), which means that we have insufficient data for a model to identify which patterns predict rare senses. This situation is hopefully going to change soon, as the integration of automated semantic tagging into our lexicographic workflow is enabling us to find more examples for rare senses, which we can then proofread for our dictionary and feed to the model as examples.

| **step** | **workflow for each lemma** |
|---|---|
| 1 | create sense inventory based on existing dictionaries |
| 2 | AI-tag all aligned sentences on the basis of translation |
| 3 | stratified random sampling of ~50 sentences by tag and genre |
| 4 | evaluate quality of AI tag against manually tagged sentences |

| | |
|---|---|
| 4a | if quality of tagging is *not* good either modify sense inventory and repeat steps 2-4, or feed the model manually tagged examples for each sense, if available. |
| 5 | feed AI the set tagged on basis of translation as examples (hundreds) |
| 6 | ask model to identify contextual features that best predict each task |
| 7 | ask model to tag all remaining sentences (thousands) on basis of identified predictors |
| 8 | evaluate a sample of AI tags against manually tagged sentences |

Table 4: summary of AI-semantic tagging workflow

## 3. Tagging sematic prosody

Beside semantic tagging, we also plan to automate the tagging of semantic prosody. We tried four different ways to automate this task, on a set of 15 words. The results so far have been less satisfactory than with semantic tagging, with accuracy not exceeding 85% even on the best performing words. At first, we prompted Sonnet 4 to tag semantic prosody following the same steps that we would take in our manual lexicographic work. The first step consists in identifying the context items that are closely connected to the headword either syntactically or conceptually, and annotate their relationship to the headword using a predefined dependencies tagset. The second step requires more subjective interpretation, as it involves deciding which one of the context item thus identified bears the most on the headword's semantic prosody, and whether the resulting semantic prosody is positive, negative, neutral or neutral-negative, the last one being reserved for cases where the headword is negated (e.g. *nir-vikalpa* or *an-akaṣara*). The model proved very good at the first step, but encountered difficulties with the second one, where it vastly over tagged sentences as positive. On average across all words, the model agreed with human lexicographers only in 56% of cases (median weighted quadratic kappa of 0.42; we use weighted quadratic kappa here to account for the distance between the polar positive and negative tags).

In an attempt to simplify the task, we gave the model English translations of the sentences as well. This helped, but the average accuracy of the tagging only increased to 65%, with a median weighted quadratic kappa of 0.55. We then tried to provide the model with many examples of each tag from our curated dictionary data. Since typically the distribution of the semantic prosody tags is very skewed, with some words having very few positive tags and other having very few negative ones, we first provided many examples of each tag regardless of the headword they referred to. We assumed that the model would be able to spot which contextual patterns best predict each tag, regardless of the word to be tagged (we also provided general rules on how to assign each tag in our prompt). This actually decreased the accuracy of the tags, bringing it down to 61% (median weighted quadratic kappa of 0.38). Finally, we only gave the model examples of tagged sentences for the headword it had to tag. This increased the correctly tagged sentences to 69% and brought quadratic kappa up to 0.54, making

this the best performing setting for sematic prosody tagging. Going forward, we intend to use the translation-based approach to tag a first set of sentences, then manually proofread them in our lexicographic work and finally use a combination of such proofread examples and translations to tag the rest of the corpus.

| approach | correct_avg | correct_range | k2_median |
|---|---|---|---|
| **mono-lemma examples** | 69.1% | 0.33:0.86 | 0.541 |
| **translation** | 65.2% | 0.43:0.85 | 0.549 |
| **mixed-lemmas examples** | 61.5% | 0.3:0.75 | 0.382 |
| **lexical relations only** | 56.5% | 0.25:0.92 | 0.428 |

Table 5: results of our approaches to semantic-prosody tagging

## 4. Impact on our lexicographic practice

Using LLMs for semantic tagging has vastly helped our dictionary work in both foreseen and unforeseen ways. The most beneficial contribution to our lexicographic practice lies in improving our data sampling. Before we automated semantic tagging, we used stratified random sampling by genre and collocate to diversify the sentences to be annotated and analyzed for our dictionary entries. This proved often impractical. The distribution of word-senses is typically very skewed in our corpus, and most senses are not closely associated with specific collocates or text-types. So, we often had to annotate vast amounts of sentences in order to find instances of the rarest meanings. Besides being time consuming, this led to unsatisfactory entries, as in many cases the examples we found for less frequent senses were too few to yield truly meaningful analysis. Now, thanks to AI-semantic tagging, we can conduct stratified sampling based on word-senses, and, to a lesser extent, semantic prosody.

Now, when starting to work on a new headword, we first get all the sentences for that headword for which we have an aligned translation and ask Sonnet 4 to tag them based on the translation. We then sample from the such tagged sentences about 30 to 50 examples featuring all the available sense-tags for that headword and we manually proofread these. Once we have the proofread the dataset we use it to evaluate the quality of the automated tags and decide whether the AI-tagged sentences are sufficiently good to be used as examples for the next round of tagging, where we tag the corpus sentences for which we have no aligned translation. Often, during this process we notice that the sense-labels need some adjustments and restart the process with a revised sense-inventory. But even in these cases, the time devoted to obtaining a representative set of semantically annotated sentences for a headword is drastically reduced, often from weeks to hours. Moreover, since our dictionary entries are automatically generated from the semantically annotated sentences, this workflow

directly reduces the time needed to produce the dictionary entries, while simultaneously increasing the amount of annotated data on which the entries are based. We used to base our entries on a couple hundred sentences at most. Now, for each headword we can couple a small number of manually proofread annotations (typically around 50) with thousands of automatically tagged sentences. Users are alerted of which portion of the information we provide is based on non-proofread automated tags, and of the estimated accuracy of the automated tags for each headword. So, they can exercise discretion in how to use our lexicographic data.

AI semantic tagging is also enabling the addition of important new features to our dictionary (scheduled for release in December 2025). For many years now, we have been offering a thesaurus view of our resource. Here users can view, for example, which words express a certain concept and estimate how its lexicalization changes across the corpus on the basis of our manually annotated sentences. Now we can automatically annotate all sentences in the corpus for all the words expressing a given concept, offering a much better representation of how its lexical field changes over time or in different types of literature. This also facilitates the assessment of the prototypicality of a headword for given concept, a feature especially helpful for translators, who make up an important segment of our target audience. Sanskrit displays an overabundance of synonyms, and it is not trivial to establish which ones constitute the more typical lexicalization of a concept. For example, there are several words that express the idea of 'word' (e.g. *nāman, vacana, śabda, pada, ruta, saṃjñā, prajñapti*, etc), and all of them are polysemic. In the absence of a semantically tagged corpus, one would have to rely on intuition to estimate which among these near-synonyms would be best rendered with the English equivalent 'word', and which ones require a more specialized or nuanced translation. Thanks to a semantically tagged corpus, we can see that while *nāman* is overall the most widespread of these synonyms, *śabda* is the one that expresses the concept of 'word' most frequently, but only in commentarial literature, where it denotes a word that is being glossed on, whereas *pada* appears to be the word that lexicalizes the idea of 'word' most consistently throughout the whole corpus, thus proving the best equivalent for the English 'word'.

In the area of translation, AI tagging has also allowed us to introduce summaries of how a headword is typically rendered in published English translations of Buddhist literature. This is a byproduct of our translation-based semantic tagging workflow, where the model is asked to identify a headword's English equivalent in each sentence it tags, which *de-facto* produces lists of translation equivalents for the headword.

While these features are undoubtedly a valuable addition to our dictionary, the most impactful aspect of our experiments with AI-semantic tagging lies in the scope of the revisions that it prompted us to undertake in our curated dictionary data. The most substantial revisions fell in two areas, sense division and semantic prosody annotations.

The former has so far been extremely rare; but still significant, since it has directly affected our dictionary. It involved the sense division for the word *prajñapti*, for which automated semantic tags were initially very poor. Upon inspection of the automated tags, we realized that the model was unable to find any strong contextual predictors for two of the subsenses that we had identified in our entry, 'designation' and 'verbal expression/notion'. We took this as a sign that such very fine-grained semantic distinction was not warranted and we revised our original entry accordingly.

In the case of semantic prosody, we noticed that for some headwords the disagreement between the model's tags and the annotations in our data was suspiciously high. This led us to conduct a systematic review of our annotations, which we would not otherwise have undertaken. To our consternation, we discovered that the model was often correct and our data wrong. this offered us a rare opportunity to spot and revise mislabeled sentences. More generally, the widespread low accuracy of the automated semantic prosody tags made us more aware of the degree of subjectivity intrinsic in this task. While double checking our annotations against the AI tagged datasets, we found many cases where multiple tags can be considered correct due to contextual ambiguities. This has prompted us to revise our annotation guidelines, furnishing annotators with a more robust framework for annotating ambiguous cases consistently.

## 5. Other applications of LLMs in our dictionary

Although this paper focusses on corpus tagging, a report on our experimentations with LLMs for lexicography would not be complete without mentioning our work on synonym comparison. As detailed earlier, the thesaurus view of our dictionary offers the possibility to view how a given concept is lexicalized by different synonyms. The dictionary also allows users to compare pairs of synonyms in various ways, such as by viewing side by side examples of both words in the same sense, or by contrasting their semantic spectra and distribution over the corpus. These features allow our users to explore our annotated data and form an opinion on the similarities and differences of the synonyms they choose to compare. Now we are also adding narrative paragraphs that provide an overview of how the synonyms in a given semantic domain compare to each other. To obtain such summaries, we start by identifying a group of known synonyms in a given semantic domain. We then we use the BERT word embeddings we trained in the past to retrieve further synonyms of these words (Lugli et al., 2022). When we reach a reasonably exhaustive list of synonyms for the given domain, we extract from the semantically tagged corpus all the sentences in which these synonyms lexicalize the domain we are interested in. We then feed these sentences to generative LLMs and prompt them to summarize the similarities and differences of the synonyms based on the sentences provided.

So far, we have tried this with Sonnet 4 and Sonar, on words related to the semantic domain of language. We have compared *śabda*, *ruta* and *ghoṣa*, which lexicalise

primarily the audible aspect of language, and *vyavahāra*, *saṃketa*, *prajñapti* and *upacāra*, which express the conventional and approximative aspect of language. Both models provided very accurate summaries for both sets of synonyms. Sonnet 4 offered an extremely well written and punctual comparison, which was virtually undistinguishable from the output of a human lexicographer. Sonar proved stronger in the identification of sentences and lexical patterns that exemplify key differences between the synonyms. However, even if prompted to base its summary on the dataset we provided, Sonar relied excessively on online material, often including rather speculative arguments based on non-academic websites. Going forward, we will discard Sonar and focus on prompting Sonnet 4 to exemplify the main points of its summary with corpus sentences.

## 6. Conclusions

Our first experiments using LLMs for producing dictionary data have been promising. Thanks to latest generation AI, corpus semantic tagging, which used to be very difficult for ancient languages, seems now amenable to automation even for an extremely low-resource language like Buddhist Sanskrit. We have tested four commercial Large Language Models on a simplified version of the semantic tagging task we intend to carry out as part of our lexicographic work. Both Sonnet 4 and Gemini 2.5 have yielded good results, with Anthropic's model achieving slightly better scores. We have then prompted Anthropic's Sonnet 4 to semantically tag all corpus instances of 22 Sanskrit headwords, using an iterative approach that combines English translations and examples sentences, as well as well-crafted sense inventories. We have obtained usable results for the majority of the 22 headwords, with an average accuracy of 80%. By contrast, tagging semantic prosody, intended here as context-based connotation, has proven harder, with average accuracy remaining below 70%. This is partly due to the interpretive nature of task, a consideration that has led us to revise our annotation guidelines to ensure more consistent tagging, on the part of our lexicographers as well as of the LLM. Finally, we have briefly experimented with using LLMs to produce narrative comparisons of near-synonyms. Sonnet 4 has generated lexicographer-grade outputs, persuading us to integrate these summaries in our dictionary. Overall, LLMs appear to offer a workable solution to accelerate our lexicographic work on Buddhist Sanskrit while at the same time increasing the data on which we can base our entries, effectively allowing us to develop a much better resource than we would have been able to produce with traditional tools and techniques.

## 7. Acknowledgements

The LLMs performance has been evaluated using a gold-standard dataset prepared in collaboration with Luis Quiñones. This research was funded by the Mangalam Research Center for Buddhist Languages.

## 8. References


Edgerton, F. (1953). *Buddhist Hybrid Sanskrit Grammar and Dictionary* (2 vols.). New Haven: Yale University Press

Guo, X., Ma, F., Dienes, Z., & Graham, S. (2011). "Acquisition of conscious and unconscious knowledge of semantic prosody." *Consciousness and Cognition*, 20(3), 481-492

Lugli, L. (2021a). Dictionaries as collections of data stories: an alternative post-editing model for historical corpus lexicography. In Itzok Kosem, et al. (eds.). *Post-Editing Lexicography: eLex 2021*, 216–231.

Lugli, L. (2021b). Words or terms? Models of terminology and the translation of Buddhist Sanskrit vocabulary. In A. Collett (ed.) *Buddhism and Translation: Historical and Contextual Perspectives*, New York: SUNY, 149–172.

Lugli, L. (2022). Embeddings models for Buddhist Sanskrit. *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)*, 3861–3871.

Lugli, L., Martinc, M., Pollak, S., Pelicon, A. (2023). Computing the Dharma: NEH White Paper. figshare. https://doi.org/10.6084/m9.figshare.24065868.v1

Martinc, M, Pelicon, A, Pollak, S, Lugli, L. (2023). Word Sense Induction on corpus of Buddhist Sanskrit literature. In Medved M. et al. (eds), *Proceedings of the eLex 2023 Conference: Silent Lexicography*, pp: 191–205.

McGillivray, B., Kondakova, D., Burman, A., Dell'Oro, F., Bermúdez Sabel, H., Marongiu, P., Márquez Cruz, M. (2022). A new corpus annotation framework for Latin diachronic lexical semantics, *Journal of Latin Linguistics*, vol. 21(1): 47–105.

Monier-Williams, M. (1899). *A Sanskrit-English Dictionary: Etymologically and Philologically Arranged with Special Reference to Cognate Indo-European Languages*. Oxford: The Clarendon Press

Stewart, Dominic. (2010). Semantic Prosody: A Critical Evaluation. Routledge.

Vatri, A., McGillivray, B. 2018. The Diorisis ancient Greek corpus: Linguistics and literature. *Research Data Journal for the Humanities and Social Sciences*, 3(1): 55–65.