Embeddings models for Buddhist Sanskrit

Ligeia Lugli[1], **Matej Martinc**[2], **Andraž Pelicon**[2], **Senja Pollak**[2]
[1]Mangalam Research Center for Buddhist Languages, Berkeley, CA, USA
[2]Jožef Stefan Institute, Ljubljana, Slovenia
ligeia.lugli@kcl.ac.uk; {matej.martinc,andraz.pelicon,senja.pollak}@ijs.si

Abstract

The paper presents novel resources and experiments for Buddhist Sanskrit, broadly defined here as including all the varieties of Sanskrit in which Buddhist texts have been transmitted. We release a novel corpus of Buddhist texts, a novel corpus of general Sanskrit and word similarity and word analogy datasets for intrinsic evaluation of Buddhist Sanskrit embeddings models. We compare the performance of word2vec and fastText static embeddings models, with default and optimized parameter settings, as well as contextual models BERT and GPT-2, with different training regimes (including a transfer learning approach using the general Sanskrit corpus) and different embeddings construction regimes (given the encoder layers). The results show that for semantic similarity the fastText embeddings yield the best results, while for word analogy tasks BERT embeddings work the best. We also show that for contextual models the optimal layer combination for embedding construction is task dependent, and that pretraining the contextual embeddings models on a reference corpus of general Sanskrit is beneficial, which is a promising finding for future development of embeddings for less-resourced languages and domains.

Keywords: Buddhist Sanskrit, embeddings, intrinsic evaluation

1. Introduction

This study presents the evaluation of the performance of static and contextual embeddings models trained on a small historical corpus of Buddhist Sanskrit literature. This literature constitutes the textual foundation of Mahāyāna, one of the main branches of Buddhism, which flourished in India from around the first couple of centuries BCE to the XII century CE. Despite extensive scholarly endeavors, much uncertainty still surrounds this body of literature, especially regarding matters of chronology, authorship, and compositional history. Moreover, the lexicographic documentation of its vocabulary remains largely outdated (Lugli, 2018).

While many world literatures and languages have benefited from the adoption of computational methods, the study of Buddhist Sanskrit sources still relies almost exclusively on traditional methods, such as close reading and philology. This is likely due to the scarcity of language resources available for Sanskrit in general and especially for the variety of Sanskrit used in Buddhist texts, which is characterized by domain-specific vocabulary, morphological patterns heavily influenced by local vernaculars and abundant spelling variation.[1] We aim to improve on this situation by introducing novel corpora and language models to extend the resources available for both general and Buddhist Sanskrit. We also seek to contribute to the Digital Humanities debate over the feasibility and relative advantages of using static and contextual word embeddings with small domain-specific historical corpora (Wevers and Koolen, 2020), and more generally to the current developments in natural language processing for low-resourced languages (Wang et al., 2020; Agić and Vulić, 2019).

We conduct a comparison study on how different static and contextual embeddings models perform in a low-resource scenario with limited training data and also explore several options for models' performance improvement. More specifically, this paper introduces novel resources for Sanskrit, including:

- novel corpora of Buddhist and general Sanskrit (see Section 3),
- novel static (fastText and word2vec) and contextual pretrained (BERT and GPT-2) embeddings models for Buddhist Sanskrit[2],
- novel word similarity and analogy evaluation datasets for Buddhist Sanskrit,
- extensive experimental evaluation of various embeddings models, including evaluation of different layers selection in contextual models, and assessing transfer learning capacity from general to Buddhist Sanskrit models.

The paper is structured as follows. After related work 2, we describe the corpora.Section 4 covers the training

[1]Note that in this study, we refer to the Sanskrit used in Buddhist literature as 'Buddhist Sanskrit', regardless of the level of vernacular influence instantiated in each text.

[2]The code for experiments is publicly available under the MIT license at `https://gitlab.com/matej.martinc/buddhist_sanskrit_embeddings` and the best performing contextual embedding model has been uploaded to the Huggingface library (`https://huggingface.co/Matej/bert-base-buddhist-sanskrit`).

of static and contextual embeddings, while Section 5 provides details on the evaluation datasets, settings and results. The paper concludes with a sketch of our plans for future work in Section 6.

2. Background and Related work

In recent years, approaches using embeddings representations have shown impressive performance in various downstream tasks and have become a crucial resource for natural language processing. In our work, we use static and contextual models. The basic **static word embeddings** model is word2vec (Mikolov et al., 2013). The aim of the algorithm is to map each word appearing in the training corpus to a unique vector representation in a shared vector space where semantically similar words are situated closer together than semantically dissimilar words. A drawback of the word2vec algorithm is that words which do not appear in the training corpus do not have their own unique representation in the vector space. The fastText (Bojanowski et al., 2017) algorithm is an update to the word2vec algorithm which deals with the the out-of-vocabulary words. To this end, fastText calculates vector representations also for subword n-grams. If any out-of-vocabulary word can be reconstructed from the n-grams, the algorithm sums the representations of the subwords into a final vector representation for the out-of-vocabulary word. The use of fastText algorithm is very appropriate for conducting experiments on a language with a rich morphology. It was shown in the original study that fastText embeddings outperform baseline word2vec static embeddings on semantic tasks, for example word analogy derivation, on morphologically rich languages.

Static word embeddings have been produced for several low-resourced languages including Khmer (Buoy et al., 2021) and Sinhala (Lakmal et al., 2020). Michel et al. (2020) developed static word embeddings based on word2vec and fastText algorithms for Hiligaynon. In their work, they produced monolingual version of the embeddings which they then projected into the common space with the English-language embeddings to produce bilingual embeddings. The authors note that the main obstacle for producing quality mono- and bilingual embeddings was lack of training data. Several works have also developed static word embeddings for general Sanskrit (e.g. Sandhan et al. (2021)). Kanojia et al. (2019) further utilized the embeddings, trained using the fastText algorithm, to produce phylogenetic trees for Sanskrit texts, and (Kumar et al., 2020) developed pretrained embeddings for several Indian languages. In Kumar et al. (2020) various embeddings models for 14 Indian languages were trained and static embeddings evaluated on part-of-speech and named entity recognition tasks. To complement the previous studies, our work focuses on developing static and contextualized embeddings for Buddhist Sanskrit.

Static embeddings models are recently being replaced by **contextual models** that can handle polysemy, such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019). BERT and GPT-2 are both based on the transformer architecture (Vaswani et al., 2017) but employ a different pretraining regime, masked language modelling and autoregressive language, respectively. Due to their popularity, they are being released for several languages, including general Sanskrit (Sandhan et al., 2021) and a variety of Indian languages (Kumar et al., 2020). The studies that evaluate these models are abundant (Rogers et al., 2020). For example, a systematic study of several models has been conducted by Vulić et al. (2020b),who tested whether pretrained language models encode type-level knowledge, and explored different ways of distilling this knowledge from the contextual representation. Several models and knowledge extraction strategies have been compared on a set of semantic tasks, namely lexical semantic similarity, word analogy derivation and lexical relation prediction. They conclude that in most cases and on most tasks contextual embedding models outperform static embedding models and are capable of modelling type-level lexical knowledge. Among other, their results suggested that using an averaged subword embeddings from multiple contexts works better than employing embeddings from a single context. They also show that transformers carry type-level lexical knowledge that is distributed across multiple layers but is nevertheless more condensed in lower encoder layers, in contrast to the semantic knowledge, which is encoded in upper layers.

Research that deals with training, evaluation and employment of contextual embeddings models on low-resource languages is scarcer. This is most likely due to the fact, that these models are believed to require massive textual resources for optimal training (Devlin et al., 2019), which are harder to obtain for these languages. Nevertheless, recently these models have been trained on smaller corpora and the results indicate that they still offer competitive performance when compared to static embedding models. In the study by Sandhan et al. (2021), they trained a transformer based model ALBERT (Lan et al., 2019) (a lite BERT model with less parameters) and the ELMo contextual embeddings model (Peters et al., 2018) based on deep bi-directional LSTM architecture on a Sanskrit corpus containing just around 6 M tokens. The models were tested in an intrinsic setting, on a set of four tasks, namely similarity, analogy, relatedness, and categorization prediction. Surprisingly, ELMo model outperformed static embeddings models, including fastText, word2vec and GloVe (Pennington et al., 2014) on most tasks and ALBERT proved competitive on some.

Another strategy for representation learning on low-resource languages is the employment of multilingual

models. The merit of this approach is analysed in the study by Vulić et al. (2020a). They compare multilingual BERT (Devlin et al., 2019) and XLM (Conneau et al., 2020) to several static embedding models on a dataset covering 12 typologically diverse languages, among them major languages such as Mandarin Chinese and Spanish, as well as less-resourced ones such as Welsh and Kiswahili. For major languages for which monolingual pretrained transformer based models exist, they report a big gap in performance between multilingual and monolingual pretrained encoders in favor of the latter, confirming the overall uncompetitiveness of multilingual models. Multilingual models performed especially badly on low-resource languages, since the data for these languages was scarce in the multilingual training sets.

We have produced evaluation datasets, which are standardly used for embeddings evaluation (Wang et al., 2019; Bakarov, 2018). There exists a range of **word similarity datasets**, including WS353 (Finkelstein et al., 2002) and SimLex999 (Hill et al., 2015) datasets for English, the derivative resources for other languages, e.g. Turkish (Ercan and Yıldız, 2018), Finnish (Venekoski and Vankka, 2017), Polish (Mykowiecka et al., 2018), and multilingual resources (Mrkšić et al., 2017; Leviant and Reichart, 2015; Camacho-Collados et al., 2017; Barzegar et al., 2018). The largest and most consistent available multilingual similarity judgement dataset is Multi-SimLex (Vulić et al., 2020a), which is a large-scale lexical resource and evaluation benchmark covering datasets for twelve typologically diverse languages. In terms of **analogy datasets**, possibly the most widely adopted analogy dataset is the Google analogy dataset for English language introduced by (Mikolov et al., 2013). As part of training static word embeddings for several languages, (Grave et al., 2018) introduced three original analogy datasets for Polish, French and Hindi. In terms of general Sanskrit, (Sandhan et al., 2021) developed an original similarity and analogy datasets for evaluating their models.

3. Corpus

A corpus of general Sanskrit of about double the size of the Buddhist Sanskrit corpus has been prepared to pretrain BERT and GPT-2 models for this study (Lugli et al., 2022). This corpus comprises 267 non-Buddhist Sanskrit texts for a total of 13.3 million tokens (excluding punctuation). The texts have been taken from GRETIL, SARIT and CTS e-texts and tokenised with the compound splitter proposed in Hellwig and Nehrdich (2018). This corpus comprises non-Buddhist religious and secular literature from the 6th century BCE to modern times, with most texts dating from the 6th to 12th century (see Table 2). The language of this corpus is mostly classical Sanskrit and presents considerably less spelling variation than the Buddhist corpus.

The main corpus used in this study is a collection of Sanskrit Buddhist texts that has been developed at the Mangalam Research Center for Buddhist Languages in California and is newly released in this study (Lugli et al., 2022). It includes all the Buddhist texts published in major repositories of unpreprocessed digitized Sanskrit material[3] that are not reconstructions from other languages, as well as a few newly digitized Buddhist Sanskrit works. All texts have been lemmatised with the tools developed at the Mangalam Research Center and enriched with metadata (Lugli, 2019). The version of the corpus used for this study comprises 311 texts for a total of 6.7 million tokens (excluding punctuation). It spans over two millennia of Buddhist literature, with works ranging from the 1st century BCE up to contemporary times, but its bulk consists of Mahāyāna scriptures (emphsūtra) dating from the first five centuries CE and treatises (emphśāstra) from around the 6th-12th century (see Table 1). As with much Sanskrit literature, the texts in the corpus are difficult to date with any certainty and several remain impossible to categorise chronologically. The language of the corpus lies on a cline between classical Sanskrit and the so-called 'Buddhist Hybrid Sanskrit', a variety of Sanskrit heavily influenced by local vernaculars (prakrits) (Edgerton, 1953). Most texts display some level of morphological and orthographic deviation from the classical language and virtually all employ domain-specific vocabulary, either by featuring words not attested outside of Buddhist literature, or, more frequently, by deploying general Sanskrit words with specialised Buddhist meanings. About one third of the texts in this corpus contains corrupted words and passages (emphlacunae) due to illegible portions in the manuscripts on which the editions are based. For this study, corrupted words have been excluded from the corpus (about 66 thousand tokens).

4. Model training

4.1. Static embedding models

Considering a relatively small unlabeled training corpus compared to pretraining regimes in related work (Devlin et al., 2019; Radford et al., 2019), we first opted to develop and benchmark static embedding models, where every token is represented as one vector in the shared representation space regardless of the context in which the word appears in. We expect the static embedding models to be competitive with the contextual embedding models on some evaluation settings or to at least serve as a strong lower bound for the performance of the embedding models trained on this particular corpus.

In this work we experiment with two algorithms for static embeddings generation, namely the word2vec

[3]GRETIL, SARIT, Thesaurus Literaturae Buddhicae, Digital Sanskrit Buddhist Canon and CTS e-texts.

Genre	I BCE-V CE	VI-XII CE	Later	Indeterminate	Total
scriptures (*sūtra*)	1,694,429	185,507	0	94,950	1,974,886
treatises (*śāstra*)	395,266	2,274,198	54,971	19,991	2,744,426
religious stories (*avadāna*)	348,843	29,365	31,324	323,455	732,987
monastic rules (*vinaya*)	376,248	54,642	0	11,937	442,827
literature and hymns (*kāvya, stotra*)	269,301	307,534	3,933	11,554	592,322
tantric texts and formulas (*tantra,sādhana,dhāraṇī*)	3,867	256,536	0	31,736	292,139
Total	3,087,954	3,107,782	90,228	493,623	6,779,587

Table 1: Composition of the Buddhist Sanskrit corpus

Genre	I BCE-V CE	VI-XII CE	Later	Indeterminate	Total
scriptures (*upaniṣad, āgama, tantra*)	12,052	519,259	0	17,419	548,730
treatises (*śāstra*)	1,171,326	6,679,420	206,736	4,712	8,062,194
religious stories (*purāṇa*)	600,270	1,227,969	0	217,425	2,045,664
literature, hymns (*kāvya, stotra*)	309,918	1,521,087	0	811,113	2,642,118
Total	2,093,566	9,947,735	206,736	1,050,669	13,298,706

Table 2: Composition of the general Sanskrit corpus

(Mikolov et al., 2013) and fastText (Bojanowski et al., 2017) algorithms. Both algorithms require to set a number of hyperparameters in advance that can influence the optimization process and impact the quality of the final models. We first set the hyperparameters for both algorithms to values which produced quality embeddings as reported in related work. For fastText algorithm, the embeddings were calculated according to Vulić et al. (2020a) using the CBOW model type with embedding dimension set to 300, character n-grams of length 5, window size of 5 and 10 training epochs. For word2vec algorithms, we have set the embedding dimensions to 300, window size to 11 and the number of epochs to 80 as it was reported in Sandhan et al. (2021). Since these hyperparameters are set experimentally, we decided to additionally perform a random search over hyperparameter space for both algorithms. For each algorithm, we repeated the training process 100 times, each time randomly varying the hyperparameters. During the random search, a slightly different set of hyperparameters was optimized depending on the algorithm used. The following set of hyperparameters was optimized for both word2vec and fastText algorithms:

- **model** - which model, either Skip-gram or CBOW, should be used to train the static word embeddings.
- **embedding dimensions**: dimensions of the final embedding space. The random search process was optimising between the following set of values for the embedding space dimensionality: 50, 100, 150, 200, 250, 300.
- **context window size** - the number of words in the neighbourhood of the target word which are used to calculate the representation of the target word. The random search optimizes for values in the range from 5 to 11.
- **number of epochs** - the number of times the algorithm runs through the whole training dataset. The random search optimizes for values in the range from 3 to 16.

In addition to the above set of hyperparameters, the following two hyperparameters were optimized for the fastText algorithm:

- **minimum subword length** - the minimum length of the subword ngrams for which a separate representation will be calculated.
- **maximum subword length** - the maximum length of the subword ngrams for which a separate representation will be calculated.

During hyperparameter optimization, each trained model was evaluated on the word analogy task over a small subset of 24 verb-noun triplets (for detailed explanation of evaluation tasks see Section 5). The model with the highest score on the subset analogy task was then chosen for further evaluation. Given the evaluation dataset, our static embedding models were trained on a lemmatized version of the training corpus, by which we also mitigate the data sparsity problem caused by the rich morphology and spelling variation of Buddhist Sanskrit.

4.2. Contextual embedding models

We experiment with two distinct contextual embedding methods, BERT (Devlin et al., 2019) and GPT-2 (Vulić et al., 2020b). While these two models are both based on the transformer architecture (Vaswani et al.,

2017), there are differences between the models in terms of size and pretraining objectives. The bert-base-uncased model[4], which we employ, contains around 110 million parameters[5] and employs masked language modelling pretraining objective. With around 1.5 billion parameters, GPT-2[6] is more than 10 times larger and employs autoregressive language modelling objective during the pretraining. The initial hypothesis is that there will be significant differences between the performance of the models. We expect that GPT-2 will be more sensitive to overfitting due to its large size and due to a relatively small training corpus size. We also expect that BERT might have a performance advantage due to its leveraging of the right side context.

We test two training regimes. The first strategy involves training a Byte-Pair Encoding (BPE) tokenizer (Sennrich et al., 2016) with a vocabulary of 30,000 tokens. The BPE tokenization is conducted after a white space pre-tokenization that splits the training data into words and ensures that byte-pairs do not range across words. The byte-pair tokens produced by the tokenizer are fed to both models in sequences of 512 tokens and the models are trained on the Buddhist Sanskrit corpus for up to 300 epochs. To tackle the possible overfitting issue, we employ the following early stopping strategy: 10% of the corpus is set aside as a development set and the loss for both models is calculated on this development set every 10,000 training steps. If the loss at a specific checkpoint is larger than at previous checkpoint, the training is stopped and the model at the previous checkpoint is used for the embedding construction.

In the second training regime, we pretrain both contextual models on the general Sanskrit corpus described in Section 3. Since there is considerable overlap in the vocabulary and grammar of general and Buddhist Sanskrit, we hypothesise that models might be able to leverage this additional textual resource to compensate for the relatively small size of the Buddhist corpus and learn some useful lexical, semantic, and grammatical information. We preprocess the corpus with the compound splitter proposed in Hellwig and Nehrdich (2018) to obtain word tokens. For the training of the BPE tokenizer, this tokenized corpus is merged with the tokenized Buddhist corpus in order to obtain a byte-pair vocabulary that covers all the training corpora. Same as in the previous training regime, the vocabulary size is set to 30,000 and the white space pre-tokenization is employed. We label the models trained according to this training regime as 'pretrained' in Tables 4 and 5.

Both models are pretrained on the general Sanskrit corpus using the same hyperparameters as for training on the Buddhist corpus, i.e. the input sequence length is set to 512, the models are trained for up to 300 epochs and the same early stopping mechanism is applied. After the pretraining, both pretrained models are trained on the Buddhist corpus, again for up to 300 epochs, with the input sequence length of 512 and the early stopping mechanism.

The resulting four contextual models (BERT, GPT-2 with and without pretraining) are used for embedding construction. We experiment with three distinct embedding generation regimes, similar to Vulić et al. (2020b). More specifically, the final embedding representation is created by averaging (1) first six encoder layers , (2) last four encoder layers, or (3) all encoder layers. The study by Vulić et al. (2020b), which was conducted for BERT models trained on several languages, has shown that the usage of first 6 encoder layers results in the best representation since lexical knowledge is mostly encoded in the lower encoder layers. Since we also employ the GPT-2 model with an autoregressive language model objective and since models in our experiments are trained on much smaller corpora than the models used in the experiments by Vulić et al. (2020b), we opted to also test two other embedding construction strategies besides the recommended first 6 encoder layers, since these differences in the experimental setup might result in a different distribution of information of specific type (e.g. lexical, semantic, etc.) in the models' encoders. If a word is split into more than one subword token, we take an embedding for each subword constituting a word and average these embeddings in order to derive a contextual representation for each word occurence. Finally, the resulting contextual embeddings are averaged across the corpus on the level of word's lemma in order to obtain a single word-type level embedding for each word's lemma. This is once again motivated by the study of Vulić et al. (2020b), where they indicated that averaging subword embeddings from multiple contexts improves the quality of the representation.

5. Evaluation

5.1. Evaluation datasets

Two evaluation datasets have been prepared for this study, a set of 98 pairs of nouns manually scored for semantic similarity (**Semantic similarity dataset**) and a set of 120 morphologically related words (**Analogy dataset**).

In creating the **semantic similarity dataset**, we tried to conform to Vulić et al. (2020a) and annotate a representative sample of vocabulary using a 7-point scale (0 to 6) that focuses on purely semantic similarity (as opposed to contextual and paradigmatic

[4] https://huggingface.co/bert-base-uncased

[5] Note that we also tested a lighter version of the BERT architecture, with 6 hidden layers instead of 12, which performed worse than the bert-base-uncased architecture. We omit the description of these experiments due to space constraints.

[6] https://huggingface.co/docs/transformers/model_doc/gpt2

relatedness). However, this approach is impractical for historical languages with no remaining native speakers. Some adaptations proved necessary. First, given historical nature and cultural specificity of Buddhist Sanskrit vocabulary, we opted for creating our own word pairs, rather than translating the resource by (Vulić et al., 2020a). Second, it is extremely difficult for non-native speakers to gauge semantic similarity with the delicacy required by a 7-point scale. To facilitate the task, annotators were invited to consider some contextual and paradigmatic relations, thus departing from purely semantic judgment. Specifically, scores 3 and 4 were recommended for words belonging to the same conceptual domain and differing levels of semantic similarity (*vitarka-vicāra*, 4; *vitarka-manas*, 3), whereas scores 1 and 2 were recommended for words that present strong contextual similarity and differing levels of conceptual relatedness (*vikalpa-prapañca*, 2; *smṛti-āyātana*, 1). Next, the need to annotate a representative sample of Buddhist Sanskrit vocabulary had to be balanced with the annotators' lexical competence. To avoid over-reliance on dictionaries, whose semantic descriptions do not always reflect the language of our corpus, and ensure a scoring based on the annotators' semantic intuition, the word pairs to be scored were selected from among vocabulary the annotators were most familiar with. This led to limiting the word pairs to 196 nouns that are either very frequent (e.g. *artha*) or well researched in Buddhist studies (e.g. *abhijñā*). Finally, given the paucity of scholars well versed in Buddhist Sanskrit literature, only 4 annotators worked on the dataset, and one had to be discarded due to low inter-annotator agreement. Part of the difficulty of achieving high inter-annotator agreement stems from the extreme polysemy of much of the Sanskrit vocabulary. While a pair of words may be similar in respect to one shared sense, their semantic spectra may differ considerably overall. Annotators were asked to focus on the sense a word typically expresses in Buddhist literature, but this does not solve all polysemy. For example, both words in the pair *gati-mārga* are used to lexicalize the concept of a road, but *mārga* often refers the path to liberation from the cycle of rebirths, while *gati* often means the type of existence into which one is reborn, thus being almost antithetical to *mārga*. Two annotators emphasised the similarity between these two words and assigned a 5 score, two highlighted the difference and rated the pair's similarity 2. To assess the agreement between the four annotators, each annotator's scores were compared with the average scores of the other annotators using Spearman rank correlation coefficient. The scores are presented in Table 3.

The **analogy dataset** consists of 24 sets of 5 morphologically related words derived from a single root: a verb, a past participle, a noun, an action noun in *-ana* and an agentive in *-in* (e.g. *kalp kalpita kalpa kalpana kalpin*). This dataset is very small because very few roots appear in our corpus in all 5 forms. Moreover we strove to craft sets that display a degree of semantic regularity between the different word forms and include at least one word typical of Buddhist Sanskrit literature. These constraints led us to include in the datasets some low-frequency items. Both datasets feature words in the lemma form, as typically given in Sanskrit dictionaries produced in Europe. This is to reduce data sparsity due to morphological and spelling variation (note that there is no unanimous lexicographic consensus as to how to lemmatize verbs; we use the stem of present active third person singular).

5.2. Evaluation setting

We evaluated the models on two intrinsic tasks introduced above. The first task is an analogy task where the model is given a triplet of words. In a standard analogy task (Mikolov et al., 2013), the first pair of words establishes a relationship and the model has to retrieve the word which is in the same relationship with the third word of the triplet (a:a* :: b:x). In our case, the first pair of words represents two word forms that stem from the same root and the model has to retrieve the word which stems from the same root as the third word in the triplet while respecting the word classes of the given words. For example, given the pair kalpita and kalpa, which are the past participle and noun forms stemming from the root kḷp, and given a third word smṛta, a past participle, the model has to return the word smṛti, a noun form stemming from the same root as smṛta, smṛ.

We construct three versions of this task using our **Analogy dataset** (see Section 5.1). In the first version, the model has to retrieve the noun word form given the verb word form; in the second the model is expected to retrieve the past participle given the verb; and in the third the model has to retrieve the noun given the past participle. For each of these versions, each word set from the analogy dataset is compared with every other word set, giving us in total 552 unique triplets for each version of the analogy task. To assess the model performance on this task we use the accuracy at one (Acc@1) and accuracy at ten measures (Acc@10).

In the second task the model is queried with two

	Correlation	p-value
Annotator A	0.9138	1.1707e-30
Annotator B	0.8660	2.4137e-39
Annotator C	0.8669	8.8789e-31
Annotator D	0.7396	3.3703e-18
Krippendorf α (Annotators A, B, C interval scale)		0.8583

Table 3: Spearman rank correlation between each annotator's score and the average score of other annotators. Note that the data of Annotator D, who had considerably lower correlation scores, was excluded from this study. In additinon, Krippendorf α is calculated.

words and it has to return a score of their semantic similarity. The semantic similarity is measured using cosine similarity, a continuous score which ranges from 0, denoting no similarity in meaning, to 1, denoting that the words have identical meaning. The task is performed using the **Semantic similarity dataset**. The performance of the model is measured by Spearman rank correlation between similarity scores output by the model and the gold standard scores.

5.3. Evaluation results

The results for the analogy task are presented in Table 4. Generally speaking, the best performance is observed for the BERT models pretrained on the general Sanskrit corpus, followed by BERT models trained only on the Buddhist Sanskrit corpus, static embedding models, pretrained GPT-2 models, and finally GPT-2 models trained only on the Buddhist Sanskrit corpus. When it comes to contextual embeddings models, there is a large gap in performance between GPT-2 and BERT models, with the best pretrained BERT achieving scores about twice as good (or even better) as the best pretrained GPT-2 model according to all criteria and across all analogy tasks. The gap is in line with our initial hypothesis, which assumed the overfitting of the GPT-2 model due to its large size and a small training corpus.

Pretraining the contextual embedding models on a general reference corpus is clearly beneficial. A pairwise comparison between pretrained models and models trained only on the Buddhist Sanskrit corpus employing the same embedding construction mechanism reveals that the pretrained model always outperforms its counterpart trained only on the Buddhist Sanskrit corpus. The improvements are in most cases substantial.

When it comes to comparison of three distinct strategies for construction of embeddings, the results are less clear. Using all encoder layers for embedding construction seems to work the best in terms of accuracy@10. Using last four encoder layers on the other hand in many cases improves the accuracy@1, at least when pretrained BERT model is used. The accuracy@1 obtained by these embeddings are substantially better on the analogy prediction between past partiples and nouns. Interestingly, the embeddings construction strategy recommended by Vulić et al. (2020b), in a majority of cases performs the worst of all three embedding construction possibilities for the analogy task. Further research would be required to confirm or deny the hypothesis that this is somehow connected to the small size of the training corpus, which might prevent the model to obtain encoder layers specialized for different types of information during training.

The results for the Simlex task are presented in Table 5. For this task, static embeddings models outperform contextual embeddings models. The only contextual model that offers a somewhat comparable performance is BERT pretrained on the general Sanskrit corpus, especially when embeddings are constructed from the first 6 layers, as recommended by Vulić et al. (2020b). On the other hand, constructing embeddings from the last 4 layers results in a substantial loss in performance. The GPT-2 model once again performs much worse than BERT, confirming the overfitting hypothesis, and pretraining the contextual models on the general Sanskrit corpus first, improves results also for the Simlex task.

When comparing static embedding models, the results show that models trained with hyperparameters from related work are comparable in performance with the best models obtained through hyperparameter optimization. This indicates that for our setting hyperparameter optimization might not be worth pursuing given additional computational cost it incurs. Additionaly, we observe that the best values for hyperparameters obtained through optimization are relatively similar with best parameters suggested in the related work. Final hyperparameters for best static embedding models obtained through hyperparameter optimization are presented in Appendix A. Additionally, we have conducted an analysis of the impact of chosen hyperparameters on the model's performance. To this end, we have calculated correlation coefficients between each optimized hyperparameter and the results on the evaluation subset using the data from the 100 hyperparameter optimization runs. We have estimated the correlation between the model type, which is a categorical variable with two possible values, either CBOW or skipgram, and the evaluation results using point biserial correlation coeffficient. For performing the calculations we assigned the value 0 to the CBOW model type and value 1 to the skipgram model type. For other hyperparameters, we have calculated the Spearman correlation coefficient. The critical value for the significance of the results was set to $\alpha = 0.05$ prior to conducting the analysis.

Our analysis shows that the hyperparameter that most affects the performance of a fastText model is embedding dimension (ρ=0.5255, p-value=1.98e-08) which is in line with the findings from the related work. Other hyperparameters that seem to affect the model quality to a lesser degree are model type (ρ=-0.2276, p-value=0.0227), where using the CBOW model seems to correlate with better results on the evaluation subset, and minimum length of subwords (ρ=-0.2947, p-value=0.003). This result indicates that allowing for shorter subwords improves the final model performance, possibly due to the ability of the model to cover higher proportion of out-of-vocabulary words.

The most impactful hyperparameter for the word2vec models proved to be the chosen model type, where again the CBOW model seems to correlate with better results (ρ=-0.6364, p-value=1.11e-12). This result

	verb-noun		verb-ppp		ppp-noun	
Model	Acc@1	Acc@10	Acc@1	Acc@10	Acc@1	Acc@10
fastText (default)	0.0127	0.1685	0.0072	0.2409	0.0000	0.0743
fastText	0.0562	0.2301	0.0000	0.1993	0.00	0.0888
word2vec (default)	0.0779	0.1993	0.0616	0.2156	0.0562	0.1775
word2vec	0.0707	0.2210	0.0616	0.2047	0.0489	0.1558
BERT pretrained all layers	0.1214	**0.4275**	**0.2464**	**0.5725**	0.1105	0.4149
BERT pretrained first 6 layers	0.1051	0.3841	0.2065	0.5489	0.0507	0.3859
BERT pretrained last 4 layers	**0.1286**	0.4058	0.2301	0.5072	**0.1975**	**0.4239**
BERT all layers	0.1105	0.3533	0.1649	0.3714	0.0562	0.3134
BERT first 6 layers	0.1014	0.3460	0.1576	0.3750	0.0417	0.2953
BERT last 4 layers	0.0978	0.2880	0.1123	0.3043	0.1014	0.2772
GPT-2 pretrained all layers	0.0707	0.1993	0.0562	0.2482	0.0217	0.1830
GPT-2 pretrained first 6 layers	0.0616	0.1812	0.0652	0.2591	0.0163	0.1594
GPT-2 pretrained last 4 layers	0.0688	0.1902	0.0634	0.2228	0.0199	0.1775
GPT-2 all layers	0.0236	0.0652	0.0072	0.0399	0.0145	0.0670
GPT-2 first 6 layers	0.0308	0.0761	0.0072	0.0453	0.0163	0.0707
GPT-2 last 4 layers	0.0199	0.0670	0.0054	0.0344	0.0145	0.0580

Table 4: Results for the analogy task. (For static embeddings, we compare optimized parameters to default ones. For contextual models, *pretrained* denotes pretraining on the general Sanskrit corpus and distinct embedding construction strategies using different layers are compared.

may indicate that CBOW model is the preferred model type for training static embeddings on low-resourced and morphologically rich languages. The other hyperparameter which significantly impacts the model quality is the number of training epochs which shows medium strong correlation with the evaluation results (ρ=0.5596, p-value=1.43e-09). This result could indicate that word2vec algorithm is less prone to overfitting on the smaller training set which makes it relatively robust for training on low-resourced languages. Surprisingly, the embedding dimensions do not have statistically significant correlation with the quality of the final word2vec model (p-value=0.0875).

Model	Correlation	P-value
fastText (default)	**0.6824**	0.0000000
fastText	0.6821	0.0000000
word2vec (default)	0.6672	0.0000000
word2vec	0.6647	0.0000000
BERT pretrained all layers	0.6492	0.0000000
BERT pretrained first 6 layers	0.6644	0.0000000
BERT pretrained last 4 layers	0.5554	0.0000000
BERT all layers	0.5753	0.0000000
BERT first 6 layers	0.6313	0.0000000
BERT last 4 layers	0.4660	0.0000013
GPT-2 all layers	0.3401	0.0006114
GPT-2 first 6 layers	0.3674	0.0001979
GPT-2 last 4 layers	0.3225	0.0012023
GPT-2 pretrained all layers	0.5689	0.0000000
GPT-2 pretrained first 6 layers	0.5681	0.0000000
GPT-2 pretrained last 4 layers	0.5459	0.0000000
Average annotator correlation	0.8822	/

Table 5: Results for the Simlex task.

For full results of correlation analysis between static embeddings hyperparameters and performance of the trained models, see Appendix B.

6. Conclusion and future work

The results show that for semantic similarity the fastText embeddings yield the best results, while for word analogy tasks, BERT embeddings work the best. We also show that for contextual models the optimal layer combination for embedding construction is task dependant, and that pretraining the contextual embeddings models on a general reference corpus of Sanskrit is beneficial, which is an interesting finding for future development of embeddings for less-resourced languages and domains.

There might be several reasons for better performance of static embeddings on the SimLex task and better performance of the contextual ones on the analogy tasks. First, in our datasets the SimLex task consist of more frequent words, while in the analogy dataset several words with very low frequency appear. BERT-based embeddings might be capable of building better representations for rare words due to BPE encoding and larger contextual window. Our findings are also aligned with Sandhan et al. (2021) on general Sanskrit, where contextual embedding performed better on the syntactic tasks.

In future work, we will use the developed models for synonym detection and word sense disambiguation. We will also further investigate how different layer combinations work for different tasks and generalise to other languages.

Availability

The corpora are available on Zenodo[7], the code for experiments and the evaluation datasets on GitLab[8] and Zenodo[9], respectively, and the best performing BERT model on Hugging Face[10].

Acknowledgements

This work was funded by a NEH Digital Advancement Grant level 2 (HAA-277246-21) and the creation of the Buddhist Sanskrit Corpus was partly funded by the British Academy (NF161436). We also acknowledge the Slovenian Research Agency core programme P2-0103. We also would like thank Bruno Galasek-Hul, Luis Quiñones and Jai Paranjape for their contribution to the creation of the evaluation datasets.

Appendix A: Final hyperparameters for best static embedding models

Table 6 presents hyperparameters of best static embedding models obtained through hyperparameter optimization.

	fastText	word2vec
model	CBOW	CBOW
embedding dimension	200	300
context window size	5	6
number of epochs	12	15
minimum subword length	3	/
maximum subword length	8	/

Table 6: Hyperparameters for best models using fastText and word2vec algorithms as found through the process of hyperparameter optimization.

Appendix B: Correlation between static embeddings hyperparameters and performance of the trained models

In Table 7 we present the full results of correlation analysis between static embedding models hyperparameters and the performance of the models, trained during hyperparameter optimization. For discussion of the results, refer to Section 5.3

	fastText	
Hyperparameter	Correlation	p-value
model (0 - CBOW; 1 - skipgram)	-0.2276	0.0227
embedding dimension	0.5255	1.98E-08
context window size	-0.1345	0.1821
number of epochs	0.1130	0.2630
minimum subword length	-0.2947	0.0029
maximum subword length	-0.0125	0.9016
	word2vec	
Hyperparameter	Correlation	p-value
model (0 - CBOW; 1 - skipgram)	-0.6364	1.11E-12
embedding dimension	0.1717	0.0875
context window size	0.1298	0.1982
number of epochs	0.5596	1.43E-09

Table 7: Full results of the correlation analysis between hyperparameters of static embedding models and performance of the trained models.

7. Bibliographical References

Agić, Ž. and Vulić, I. (2019). JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy, July. Association for Computational Linguistics.

Bakarov, A. (2018). A survey of word embeddings evaluation methods. *ArXiv*, abs/1801.09536.

Barzegar, S., Davis, B., Zarrouk, M., Handschuh, S., and Freitas, A. (2018). SemR-11: A multi-lingual gold-standard for semantic similarity and relatedness for eleven languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Buoy, R., Taing, N., and Chenda, S. (2021). Khmer text classification using word embedding and neural networks. *arXiv preprint arXiv:2112.06748*.

Camacho-Collados, J., Pilehvar, M. T., Collier, N., and Navigli, R. (2017). SemEval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26, Vancouver, Canada, August. Association for Computational Linguistics.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. pages 8440–8451, July.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ercan, G. and Yıldız, O. T. (2018). AnlamVer: Semantic model evaluation dataset for Turkish -

[7] https://doi.org/10.5281/zenodo.5847100

[8] https://gitlab.com/matej.martinc/buddhist_sanskrit_embeddings

[9] 10.5281/zenodo.6523884

[10] https://huggingface.co/Matej/bert-base-buddhist-sanskrit

word similarity and relatedness. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3819–3836, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131, January.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Hellwig, O. and Nehrdich, S. (2018). Sanskrit word segmentation using character-level recurrent and convolutional neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2754–2763.

Hill, F., Reichart, R., and Korhonen, A. (2015). SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, December.

Kanojia, D., Dubey, A., Kulkarni, M., Bhattacharyya, P., and Haffari, G. (2019). Utilizing word embeddings based features for phylogenetic tree generation of sanskrit texts. In *Proceedings of the 6th International Sanskrit Computational Linguistics Symposium*, pages 152–165.

Kumar, S., Kumar, S., Kanojia, D., and Bhattacharyya, P. (2020). "a passage to India": Pre-trained word embeddings for Indian languages. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 352–357, Marseille, France, May. ELRA.

Lakmal, D., Ranathunga, S., Peramuna, S., and Herath, I. (2020). Word embedding evaluation for sinhala. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1874–1881.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Leviant, I. and Reichart, R. (2015). Judgment language matters: Multilingual vector space models for judgment language aware lexical semantics. *CoRR*, abs/1508.00106.

Lugli, L. (2018). Drifting in timeless polysemy: Problems of chronology in sanskrit lexicography. *Dictionaries: Journal of the Dictionary Society of North America*, 39(1):105–129.

Lugli, L. (2019). Buddhist sanskrit segmenter, available at: https://doi.org/10.5281/zenodo.3526469.

Michel, L., Hangya, V., and Fraser, A. (2020). Exploring bilingual word embeddings for hiligaynon, a low-resource language. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2573–2580.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mrkšić, N., Vulić, I., Ó Séaghdha, D., Leviant, I., Reichart, R., Gašić, M., Korhonen, A., and Young, S. (2017). Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

Mykowiecka, A., Marciniak, M., and Rychlik, P. (2018). SimLex-999 for Polish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Sandhan, J., Adideva, O., Komal, D., Behera, L., and Goyal, P. (2021). Evaluating neural word embeddings for sanskrit. *arXiv preprint arXiv:2104.00270*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Venekoski, V. and Vankka, J. (2017). Finnish resources for evaluating language model semantics.

In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 231–236, Gothenburg, Sweden, May. Association for Computational Linguistics.

Vulić, I., Baker, S., Ponti, E. M., Petti, U., Leviant, I., Wing, K., Majewska, O., Bar, E., Malone, M., Poibeau, T., et al. (2020a). Multi-simlex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity. *Computational Linguistics*, 46(4):847–897.

Vulić, I., Ponti, E. M., Litschko, R., Glavaš, G., and Korhonen, A. (2020b). Probing pretrained language models for lexical semantics. In Bonnie Webber, et al., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7222–7240. Association for Computational Linguistics.

Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C.-C. J. (2019). Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, 8:e19.

Wang, Z., K, K., Mayhew, S., and Roth, D. (2020). Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online, November. Association for Computational Linguistics.

Wevers, M. and Koolen, M. (2020). Digital begriffsgeschichte: Tracing semantic change using word embeddings. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 53(4):226–243.

8. Language Resource References

Edgerton, Franklin. (1953). *Buddhist hybrid Sanskrit grammar and dictionary (2 Vols.)*. Motilal Banarsidass.

Ligeia Lugli and Bruno Galasek-Hul and Luis Quiñones. (2022). *Segmented Corpus of Buddhist Sanskrit (proof of concept (v1.7)*. Zenodo: http://doi.org/10.5281/zenodo.5847100.