Embeddings models for Buddhist Sanskrit

Ligeia Lugli, Matej Martinc, Andraž Pelicon, Senja Pollak

LREC 2022

This study was conducted at the *Mangalam Research Center* thanks to funding from the *NEH* (HAA-277246-21)

Our Focus: Buddhist Sanskrit

domain-specific variety of Sanskrit used in Indic Buddhist literature, esp. Mahāyāna.

Is characterized by

- Specialized vocabulary
- Vernacular influences
- Spelling variation

In this study we use the label Buddhist Sanskrit to refer to the language of Buddhist Sanskrit literature, regardless of the level of vernacular influence instantiated in individual texts

this includes but is *not limited to Buddhist Hybrid Sanskrit*

Our goal: comparing embeddings models

compare the performance of different static and contextual embeddings models trained on a Buddhist Sanskrit corpus, and more generally on *small historical corpora*

Our Corpora

A corpus of **Buddhist** Sanskrit

- tokenized & lemmatized
- 311 texts
- 6.7 million words
- Most texts dating II BCE-XII CE

A corpus of **general** Sanskrit

- tokenized
- 342 texts
- 13.3 million words
- Most texts dating VI CE-XII CE

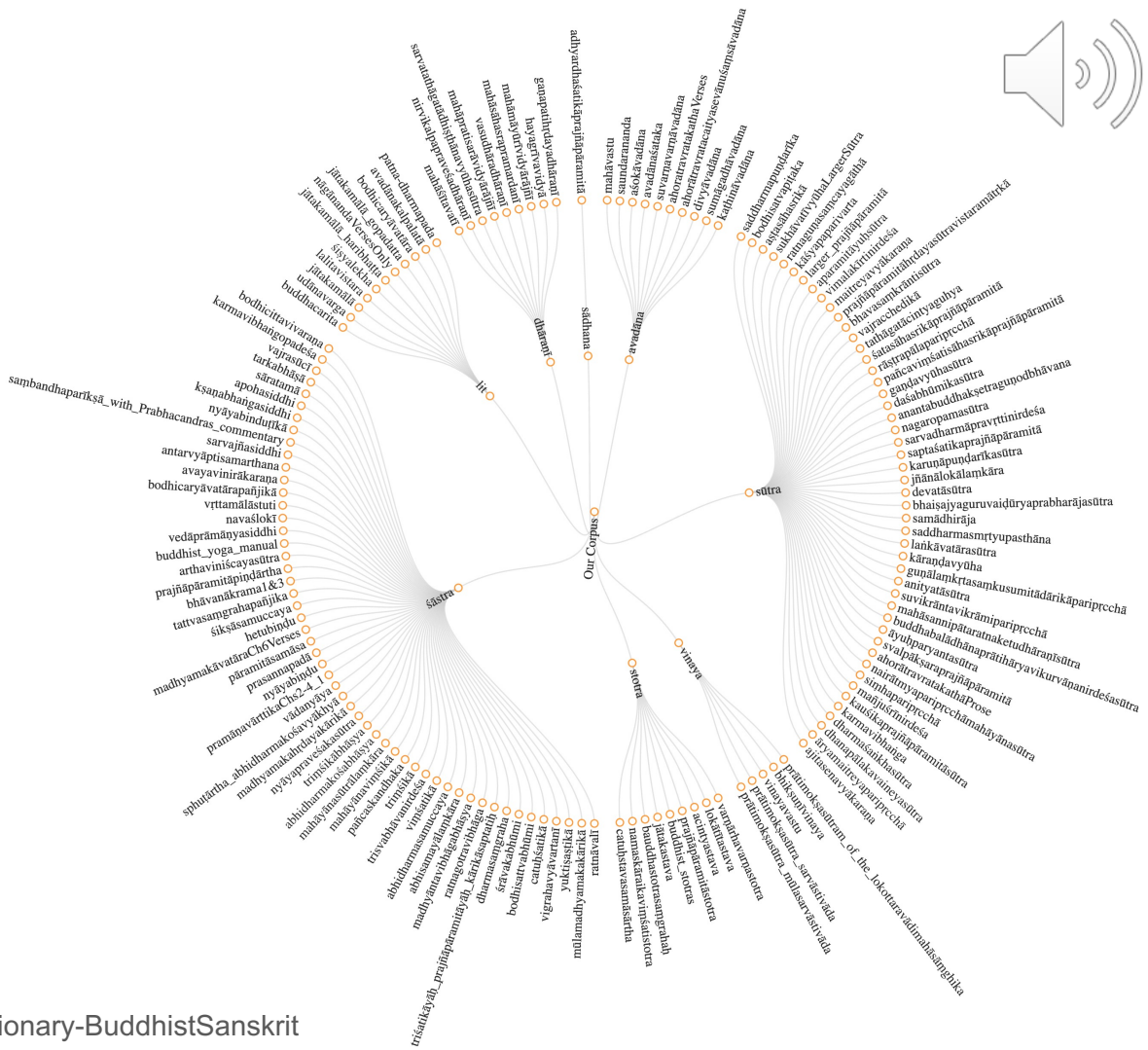


fig: Buddhist corpus by genre. source: bit.ly/VisualDictionary-BuddhistSanskrit

Contextual embedding models - training

- Two models, BERT and GPT-2:
  a. We expect difference in performance due to difference in models' size and language model objective
- Two training regimes:
  a. training just on Buddhist Sanskrit corpus
     - Byte pair tokenizer training - vocabulary size of 30.000 tokens
     - masked (BERT) or autoregressive (GPT-2) language model objective
  b. pretraining on the general Sanskrit corpus
     - Byte pair tokenizer training on the concatenation of general and Buddhist Sanskrit corpus - vocabulary size of 30.000 tokens
     - Models are pretrained on the general corpus before training on the Buddhist corpus.

Contextual embedding models - embedding extraction

- We test **three distinct embedding generation regimes**, following Vulić et al. (2020):
  a. averaging first six encoder layers
  b. averaging last four encoder layers
  c. averaging all encoder layers
- Generated contextual embeddings are **averaged across the corpus on the level of word's lemma**
- Final representation is a **single word-type level embedding for each word's lemma.**

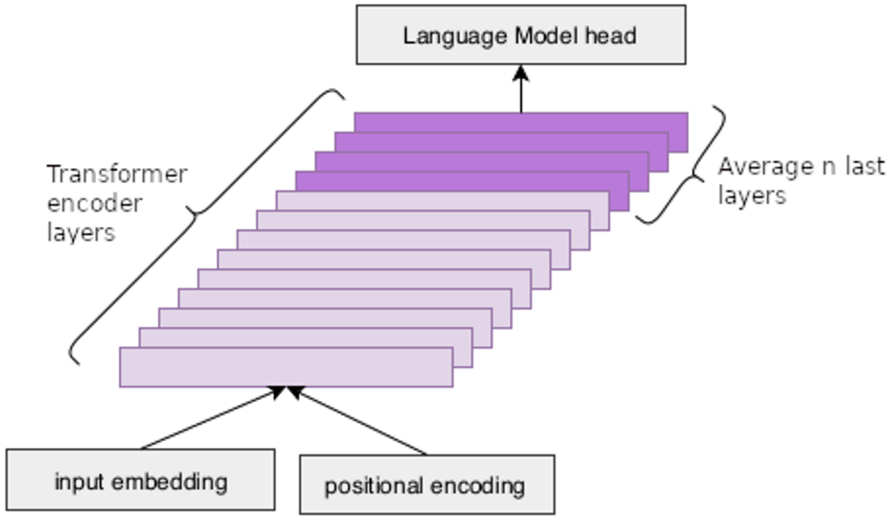
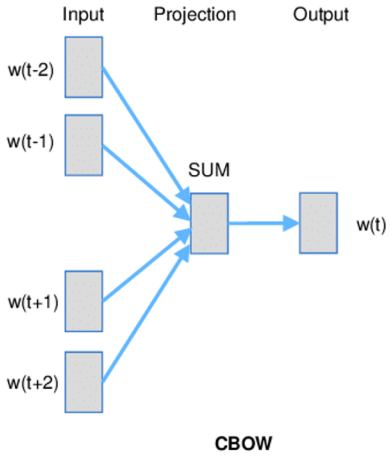


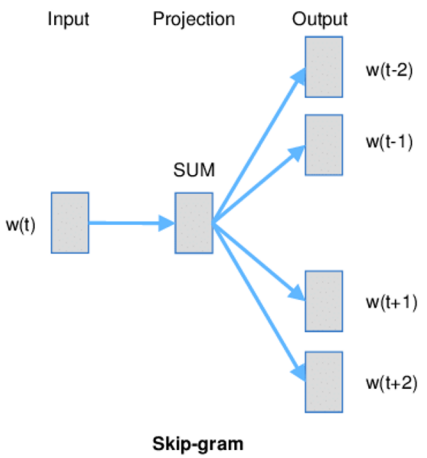
fig: the embedding extraction procedure, where n transformer encoder layers are averaged to obtain a contextual embedding

Static Embeddings Models

- Tested two static embeddings algorithms - Word2vec and fastText
  a. We assume fastText algorithm will produce better embeddings as Buddhist Sanskrit is a highly inflectional language

- Performed hyperparameter optimization for both algorithms
  a. fastText and word2vec: model type (Skipgram or CBOW), embeddings dimensions, window size, number of training epochs
  b. fastText only: minimum subword length, maximum subword length
  c. Optimization performed over 100 runs; hyperparameters were tested on a small subset of the analogy task
  d. Compared the optimized model with the model trained using hyperparameters from related work





Evaluation data

- Analogy task
  - 24 sets of 5 morphologically related lemmata derived from a single root:
    a verb, a past participle, a noun, an action noun in *-ana* and an agentive in *-in*
    e.g. root = √kḷp, set = *kalp kalpita kalpa kalpana kalpin*
  - Few roots appear in the corpus in all 5 forms and some forms are rare
    so, this dataset is small and includes low-frequency lemmata

- Simlex task
  - 98 noun pairs, most very frequent in the Buddhist corpus
  - Scored for semantic similarity on a 0-6 scale
  - 4 annotators, one discarded due to low inter-annotator agreement

Semantic similarity in ancient languages: a caveat

It is extremely difficult to gauge semantic similarity in ancient languages

Especially with highly polysemic vocabulary, as in Buddhist Sanskrit

To facilitate the task annotators were asked to

- take into account contextual and paradigmatic relations
- focus on the sense a word typically expresses in Buddhist literature

Still some ambiguity remained, e.g. for the pair *mārga-gati*

Evaluation setting

- Analogy task
  - Given a triplet of words where the first pair defines a relationship, the model has to retrieve the word which is in the same relationship with the third word.

kalpa kalp smṛti ?

= smar

  - Defined three relationships: verb - past participle; noun - verb; noun - past participle.
  - For each relationship, all possible triplets from the analogy dataset were constructed for a total of 552 triplets per task.

Evaluation setting

- Simlex task
  - Given two words, the task is to give a score of their semantic similarity.

(vitarka, vikalpa) = similarity score

  - Similarity is measured by cosine similarity ranging from 0 (no similarity in meaning) to 1 (same meaning).
  - Model scores are compared to the annotator scores from the simlex test dataset using correlation analysis.

Results

- Analogy task

	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

Results

- Simlex task

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	/

Static Embeddings - Impact of Hyperparameters

- We performed a correlation analysis between hyperparameters and model performance
- Used data from the 100 runs of hyperparameter optimization
- **fastText:**
  a. embeddings dimension has the greatest effect on model performance ($\rho = 0.5255$, p-value = 1.98e-08)
  b. CBOW models tend to outperform Skipgram models ($\rho = -0.2276$, p-value = 0.0227)
  c. minimum length of subwords ($\rho = -0.2947$, p-value = 0.003); possibly enables the model to cover larger proportion of out-of-vocabulary words
- **word2vec:**
  a. choice of model has the greatest effect on model performance (CBOW; $\rho = -0.6364$, p-value = 1.11e-12)
  b. larger training epochs correlate with better models ($\rho = 0.5596$, p-value = 1.43e-09); may indicate word2vec is less prone to overfitting

Conclusions

- For semantic similarity fastText embeddings yield the best results, for word analogy tasks, BERT embeddings work the best.
- The optimal layer combination for contextual embedding construction is task dependent.
- **Pretraining** the contextual embeddings models on a general reference corpus of Sanskrit is **beneficial**.
- In our setting, **hyperparameter optimization** does **not** produce significantly better static embeddings models when compared with default hyperparameters.

Thanks

special thanks to

Bruno Galasek-Hul
Luis G. Quiñones
Jai Paranjape

For their work on the evaluation datasets used in this study