Literature Review XCS224U February 2021, by Kees Varekamp

Universal Sentence Embeddings

Recent models and their properties

1.1 Introduction

Sentence embeddings are vectors that represent sentences. Often such sentence embeddings arise as a by-product of some task – for example the hidden state in a seq2seq translation model. These embeddings will be useful for the task at hand, but they may not perform as well on other tasks.

Ever since the success of general word embeddings such as Word2Vec and GloVE, the race has been on to find similar universally useful embeddings for entire sentences. The advantage of such embeddings is that they can be used off the shelf for new tasks. You may perhaps want to fine-tune them (and you can!), but you don't want to have to train them from scratch. This can quickly become expensive, both in terms of data and compute.

1.2 SentEval

Whether a sentence embedding is any good is generally measured by the SentEval benchmarking toolkit (Conneau & Kiela, 2018). This testing suite consists of two parts: The "downstream" part, which consists of a set of tasks that test whether the resulting vectors are any good as inputs on a set of fairly standard NLP tasks, and the "probing" part, which "probes" the embedding space to "to evaluate what linguistic properties are encoded in your sentence embeddings". So you could say that the "downstream" part is concerned with the vector properties of of the embedding ("Are these vectors useful for computing with?"), whereas the "probing" part is concerned with the sentence part of the embedding ("To what extend does this vector still represent the original input?")

This summary compares 4 of the most recent attempts at producing universal sentence embeddings. Some of the questions this article tries to answer are: What are the underlying ideas of their strategies? What do they have in common? Where do they differ? It also describes a 5th paper that takes the "probing" task more literally than SentEval – it literally tries to reconstruct the input from the output.

2 The Models

2.1 Predecessors (2015-2018)

A lot of work has been done in this space, and there is not enough room here to go over all of it. So this summary only includes four of the most recent models in some detail. It is omitting important ground work done by the following 3 studies and their resulting models in particular:

- **SkipThought** (Kiros et al, 2015): Basically "Thou shalt know a word by the company it keeps", but extended to sentences. An unsupervised method based on the idea that sentences that are close to each other are more similar than sentences that are far away from each other, much like word2vec and glove.
- **InferSent** (Conneau et al, 2017): max pooling on the hidden states of a biLSTM trained on the Stanford Natural Language Inference dataset
- **QuickThoughts** (Logeswaran & Lee, 2018): A non-generative version of SkipThought – rather than trying to reconstruct neighbouring sentences, it tries to train a classifier for predicting neighbouring/not neighbouring.

2.2 Universal Sentence Encoder (2018)

Universal Sentence Encoder (USE) (Cer et al, 2018) is a model that comes out of Google. The accompanying paper is unfortunately rather light on details:

- The goal of USE is to provide easy-to-use sentence embeddings with good transfer performance.
- There are actually 2 implementations of the encoder:
    - A Transformer based one using (self) attention that mean-pools the word embeddings to produce sentence embeddings
    - A Deep Averaging Network (DAN) that simply averages word embeddings and bi-gram embeddings and feeds the resulting representation into a straightforward deep neural network. The DAN trades accuracy for performance, with respect to the transformer, which scales with the square of the sentence length.
- The models are trained on a set of self-supervised tasks and on SNLI
- The unsupervised tasks are trained on Wikipedia, web news, web question-answer pages and discussion forums
- The model performs well on SentEval, better than InferSent
- One interesting note in the paper mentions that for STS tasks they use the angular distance (the arccos of the cosine similarity) as a distance measure, because “arccos converts cosine similarity into an angular distance that obeys the triangle inequality. We find that angular distance performs better on STS than cosine similarity.”

2.3 Sentence-BERT (2019)

Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) is a BERT based model that uses a Siamese network approach to fine-tune BERT on a set of tasks in order to produce useful sentence embeddings.

It starts with pointing out that BERT by itself already produces token level context aware embeddings that can be (mean) pooled into sentence embeddings but that these embeddings are not very effective. Furthermore, BERT-like models can be fine-tuned into high performing classifiers for STS in a straightforward manner but the problem with that is that in the resulting model the two sentences to be compared need to be fed into the input simultaneously. This may be fine for benchmark STS tasks but it poses a problem for compute intensive tasks such as clustering or nearest neighbours search: Say you want to find the two most similar sentences in a set of 10000, you will need to feed every combination (n*(n-1)/2 = 49995000 combinations) into the model.

For SBERT then, the goal is to produce embeddings that are primarily useful for computationally and/or combinatorially expensive tasks like clustering and nearest neighbour search – producing vectors that can be used in optimized computation libraries such as numpy or sklearn.

- SBERT promises semantically meaningful sentence embeddings (similar sentences are close in vector space).
- It tries to accomplish this by pooling the token embeddings while training on STS tasks but using fixed weights between 2 BERTs, so that the resulting embeddings for the two input sentences are created using identical weights.
- It claims to be better on STS or SentEval than InferSent or USE.

- SBERT trains using 3 objectives:
    - Classification, by pooling the embeddings of u and v as [u,v and u-v] as input to a softmax, using cross entropy loss
    - Regression (presumably on STS), by using cosim(u,v) using MSE as the loss
    - Triplet loss: Given 3 sentences; a (anchor), p (positive example) and n (negative example), train the model such that |a-p| < |a-n|, or in other words: minimize |a-p| - |a-n|.
- Trained on SNLI and MultiNLI
- Does well on STS tasks (as expected)
- Does well on SentEval tasks related to sentiment too (unexpected)

2.4 LASER (2019)

LASER (Artetxe & Schwenk, 2019) is a bit different from the other models in this summary in that it is primarily interested in universal *language agnostic* sentence embeddings. It is trained on a large dataset consisting of 93 languages. It uses a relative simple biLSTM to encode the input text – BPE tokenized text of any language is fed into the same encoder. This creates a fixed size language agnostic embedding that is then used in the decoder to translate the output into a specified language. The paper claims that fixed length representations are more versatile and compatible than variable length reps: “For instance, there is not always a one-to-one correspondence among words in different languages (e.g. a single word of a morphologically complex language might correspond to several words of a morphologically simple language), so having a separate vector for each word might not transfer as well across languages.”

- The embeddings are results of max pooling the hidden states of layers of the biLSTM of dimensionality 512, concatenating forward and backward hidden state into sentence representations of dimensionality 1024
- Does well on
    - XNLI (entailment)
    - MLDoc (document classification)
    - BUCC (finding the same sentence in another language)
    - Tatoeba: a new cross language similarity search benchmark
- The paper mentions an interesting problem arising from K nearest neighbour search tasks: It says that when you use cosine similarity between two sentences x and y there may be a scaling inconsistency (Guo et al 2018). It refers to an alternative approach (Artetxe and Schwenk, 2018) that consists of taking some margin function of the cosine similarity between x and y and the sum of the average cosine similarity of x and y with their respective K nearest neighbours. They suggest the quotient as a suitable margin function.
- In the future, the authors would like to improve the result by using a self attention encoder, pre trained word embeddings, and back translation.

2.5 DeCLUTR (2020)

DeCLUTR (Giorgi et al, 2020) aims to produce useful universal sentence embeddings by unsupervised learning only. It points out that the best results so far have been obtained by methods that used at least some supervised learning, but that it is important to close the gap between supervised and unsupervised methods for languages and domains for which no supervised data exists.

- Wants to be good at wide variety of tasks
- Most universal sentence embedders train supervised on SNLI or multiNLI (entailment, contradiction, neutral). Examples are InferSent, USE, and SBERT.
- The authors describe Skip-thought as an unsupervised generative model that uses sentence embeddings to predict words in neighbouring sentences. They mention that the generative nature of the model makes it expensive and surface focused.

QuickThoughts tries to improve on this by classifying context sentences from non-context sentences rather than generating them.

- The authors describe DeCLUTR's approach as similar to SBERT, but self-supervised, and the objective as similar to QuickThoughts, but using segments rather than whole sentences.
- Like SBERT's triplet approach, it uses contrastive loss: It tries try to minimize |a-p| - |a-n|.
- Like QuickThoughts, it classifies sentences (or rather sentence segments) as near by or far away.

3 Vec2Sent

The last paper in this summary is not a model, but rather a proposed benchmark that may provide a better measure of preserved meaning than the probing task in SentEval.

3.1 A better probing task?

Vec2Sent (S2V) (Kerscher & Eger, 2019) proposes that in order to probe the opaque space of a sentence embedding model, you could simply "Unveil the language encoded in sentence embeddings by conditionally generating from them".

- They do this by building a "Mixture of Softmax" RNN model and applying it to various embeddings
- They suggest a couple of metrics (x=input, y=reconstructed output): Id(x, y), the fraction of sentences where x=y, Perm(x, y), the fraction of sentences where the words of x are a permutation of the words of y, Id/Perm, the ratio of those two, BLEU(x, y), the n-gram overlap, and Mover(x, y), the MoverScore.
- They suggest that Id/Perm has the best correlation with the SentEval downstream tasks. They explain this as follows: "This means that an encoder performs better on downstream tasks if it satisfies two conditions: it can correctly identify all words in x and place them in correct word order". This doesn't make a whole lot of sense though, as Perm(x, y) is in the denominator and Id(x, y) is in the numerator

The paper then makes a couple of interesting observations. The first one is that you can try to use these sentence embeddings to do analogies just like the famous king-man+woman=queen example from word embeddings. This was already possible of course, but without being able to decode a result of such an analogy task you have to pick the nearest neighbour of the result. This works alright for word analogy tasks, where it is feasible to precompute the embeddings of "all the words", but you can not precompute the embeddings of "all the sentences". So having a decoder makes it possible to investigate sentence analogy tasks at scale.

The second interesting observation is that different embedding models produce different results in the decoder while being roughly equivalent on SentEval. The main example is SBERT, which does well on SentEval but scores badly on S2V. The authors suggest that some sentence embeddings seem to focus more on "surface level information", resulting in bad S2V performance, whereas others focus more on downstream performance.

Finally, the paper is light on examples (1 example for input reconstruction, 2 examples for analogy tasks), and it has not published their code (although it seems they intend to).

4 Comparison

These are the main similarities and differences of four models presented above:

4.1 Objective

USE, SBERT, and DeCLUTR all try to produce universally useful sentence embeddings, with SBERT perhaps putting slightly more emphasis on STS tasks than the others. LASER is the odd one out here, since its main focus is on cross language embeddings.

4.2 Testing

USE, SBERT, and DeCLUTR use SentEval to measure their performance on a variety of tasks. LASER is again the exception; it uses cross language specific benchmarks. It nevertheless produces decent results on SentEval, so it is worth including it in this summary.

Another interesting phenomenon is that all the models tend to produce only a subset of SentEval benchmarks – probably the ones they perform best in. It would be interesting to run the full suite on all of the tests on all of the models and publish a proper sentence embedding shootout dashboard.

4.3 Supervised or Not?

The four approaches use various mixes of supervised and unsupervised training (Unsupervised here includes self-supervised and transfer learning models)

Model	Supervised	Unsupervised
USE	✓	✓
SBERT	✓	✓
LASER	✓	
DeCLUTR		✓

In the world of word embeddings it seems that the unsupervised methods have won out over the supervised ones – mostly because of the sheer amount of training data that becomes available to them by virtue of being unsupervised. This may imply that the future of sentence embeddings is also unsupervised, but we are not there yet.

One interesting approach that is missing from all methods so far is to utilize the dependency parse tree of a sentence. It is not clear whether this has already been tried and rejected or whether there is another reason that it does not show up in the literature.

4.4 Methods

- Almost all the supervised models (including InferSent, excluding LASER) use SNLI for training.
- Some of the unsupervised methods use generative modelling (SkipThought, LASER) and some discriminative modelling (QuickThought, DeCLUTR). USE is unclear on which unsupervised methods were used.
- Both SBERT and DeCLUTR use what they call triplet loss or contrastive loss, respectively: Given 3 sentences; a (anchor), p (positive example) and n (negative example), train the model such that |a-p| < |a-n|, or in other words: minimize |a-p| - |a-n|.

5 Future Work

Universal sentence embeddings seem a popular research topic, and if anything this summary demonstrates that there are many approaches to the problem. There does not seem to be real consensus yet on the best approach in general.

5.1 Better testing

There does seem to be consensus on the benchmark – all of the models that explicitly aim for universal sentence embeddings use SentEval to measure their performance. Unfortunately they all seem to restrict themselves to only subsets of SentEval. It would be interesting to:

- Build an impartial shootout of sentence embeddings according to SentEval, and potentially Sent2Vec and other metrics.

The Sent2Vec paper suggests that using a decoder for reconstructing the input from the output is a good and objective way of “probing the embedding space”. But the paper misses out on USE and DeCLUTR. It also does not list complete results for S2V correlation between the SentEval “Probing” tasks and the S2V task, and it is sparse with examples on the reconstruction task and the sentence analogy task.

It may be interesting to extend on the work of S2V by:

- Building a better decoder, perhaps using transformer architecture with self attention and beam search.
- Extend the results to include USE and DeCLUTR
- Extend the analogy task by building a dataset of sentence analogies
- Compare the results of S2V to the results of SentEval Probing tasks.

Another interesting idea in the testing field would be to spend some more effort on evaluating the quality of the resulting vectors from an analytic point of view: Two of the papers mention problems with cosine similarity for example, indicating that the space in which these embeddings live might not be as smooth and flat as we would like it to be. It would be interesting to:

- Experiment with various distance measures, find which ones perform best for STS, and whether this applies across all models or if this is model dependent.

An interesting application of sentence embeddings is to cluster them – it is in fact one of the main motivations of SBERT to make this possible. It would be interesting to:

- Extend SentEval by testing how well the embeddings of a given model are suited to clustering. This could be done by using the labeled classification datasets in SentEval and interpret the labels as clusters. Then you could use cluster quality metrics (such as Silhouette Score) to measure the embeddings “fitness for clustering”

Finally: most of the benchmarks in NLP are not that interested in the cost of inference. However, this is a massively important aspect of technologies such as these. If the cost of inference is prohibitive for using the model in production, the model does not have a great chance of success out in the wild. It would be interesting to:

- Extend benchmarks such as SentEval with an analysis of the inference cost

5.2 Better Results

Some interesting ways of trying to produce better results are:

- Making a “Frankenbedding” by concatenating all of the models’ outputs and see how the result performs. This would never produce any breakthroughs into how to create good sentence embeddings, but it may shed some light on how the number of dimensions affects the performance.
- The above mentioned “Frankenbedding”, or a dimensionally reduced version of it, could also be a good starting point for a “student-master” setup to try to recreate this embedding.

Finally, since most of the well performing models use some form of supervised training, it may be interesting to look into forms of weak supervision. Technologies like Snorkel (Ratner et al, 2017) have shown that weak supervision is useful in NLP, because transformer architectures such as BERT are capable of generalizing the results of models trained on weakly labeled data quite well. One interesting approach could be to:

- Augment training data by artificially creating sentences that are more or less similar to each other. This could be achieved by using WordNet together with word embeddings to replace certain words in a sentence, and then taking the geometric mean of the cosine similarities of the replaced words with their replacements to calculate a score for sentence similarity. This would only work for sentence pairs with identical structures and just different words, but models like SBERT might be able to generalize this to sentence pairs with different structures.

6 References

- Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. CoRR, abs/1506.06726.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 670–680, Copenhagen, Denmark, September. Association for Computational Linguistics.
- Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. CoRR, abs/1803.02893.
- Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. CoRR, abs/1803.05449.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. 2018. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert- networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 11 2019.
- Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross- lingual transfer and beyond. CoRR, abs/1812.10464.
- John M. Giorgi, Osvald Nitski, Gary D. Bader, Bo Wang, 2020. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. arXiv:2006.03659v2
- Martin Kerscher, Steffen Eger. 2020. Vec2Sent: Probing Sentence Embeddings with Natural Language Generation. Proceedings of the 28th International Conference on Computational Linguistics
- Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, Christopher Ré. 2017. Snorkel: Rapid Training Data Creation with Weak Supervision. Proceedings of the VLDB Endowment, 11(3), 269-282, 2017