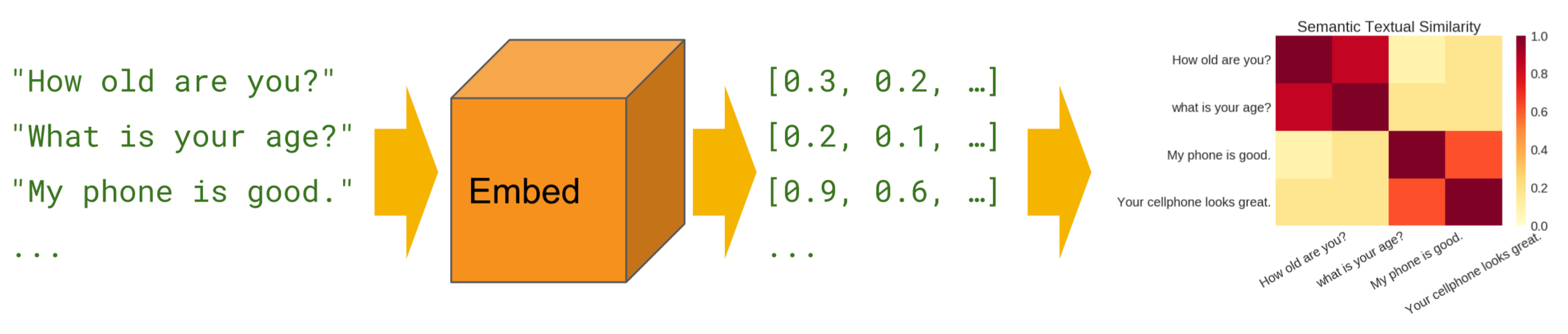
August 28, 2018 · 6 min to read

Paper Summary: Evaluation of sentence embeddings in downstream and linguistic probing tasks

Breaking down a paper that broke down some of the pros and cons of different sentence embeddings.



What's in a Sentence Embedding?

A little while ago, I wrote an article on word embeddings as an introduction into why we use word embeddings and some of the different types of word embeddings out there. We can think of a sentence embedding as just the next level of extraction: numerical representation of a sentence!

But just like we don't want to use simple numerical representation for words, we want our sentence representations to encapsulate rich meaning as well. We want them to be responsive to changes in word order, tense, and meaning.

This is a hard task! There are some ways to go about this, but recently the paper, "Evaluation of sentence embeddings in downstream and linguistic probing tasks" decided to take a stab at unravelling the question of "what is in a sentence embedding?" In it, they take a look at how difference sentence representations perform not only on downstream tasks that would benefit from sentence representation but also on tasks that are purely linguistic (tasks that show intelligent representation of the sentence with respect to linguistic features and rules).

This paper is what we dissect in this post!

Different Sentence Representations

So what representations were evaluated?

- ELMo (BoW, all layers, 5.5B): From AllenNLP, this is the pre-trained ELMo embedding. This was the English representation that was trained on the 5.5B word corpus (a combination of Wikipedia and the monolingual news crawl). Since ELMo has two layers, this representation was all the layer outputs for a total of 3072 dimensions. ELMo is just a word embedding though, so this representation was created for sentences by averaging all words together.
- ELMo (BoW, all layers, original): Another variation of ELMo, but just trained on the news crawl. Again, it has a dimensionality of 3072.
- ELMo (BoW, top layer, original): Another variation of ELMo with just the final output. This embedding only has 1024 dimensions.
- FastText (BoW, Common Crawl): From Facebook, we get FastText. My previous word embedding talks about why FastText is pretty awesome, but since it's just word embeddings, these are transformed to sentence embeddings through averaging of all words. It has a dimensionality of 300.
- GloVe (BoW, Common Crawl): GloVe, averaged together like other word embeddings. It has a dimensionality of 300.
- Word2Vec (BoW, Google News): Word2Vec, averaged together. It has a dimensionality of 300.
- p-mean (monolingual): A different way of average words together, p-mean is available through TensorFlow Hub. The dimensionality is huge though, reaching 3600.
- Skip-Thought: The first actual sentence embedding we look at. Skip-Thought uses the word2vec approach of predicting surrounding sentences based on the current sentence. It does this through an encoder-decoder architecture. This is our biggest representation, with 4800 dimensions.
- InferSent (AllNLI): Another set of embeddings trained by Facebook, InferSent is trained using the task of language inference. This is a dataset where two sentences are put together and a model needs to infer whether they are a contradiction, a neutral pairing, or an entailment. The output is an embedding of 4096 dimensions.
- USE (DAN): Google's basic Universal Sentence Encoder (USE), the Deep Averaging Network (DAN) is available through TensorFlow Hub. USE outputs vectors of 512 dimensions.
- USE (Transformer): Finally, Google's heavy duty USE, based on the Transformer network. USE outputs vectors of 512 dimensions.

The models' training and dimensionality is summarized here:

Name	Training method[1]	Embedding size
ELMo (BoW, all layers, 5.5B)	Self-supervised	3072
ELMo (BoW, all layers, original)	Self-supervised	3072
ELMo (BoW, top layer, original)	Self-supervised	1024
Word2Vec (BoW, Google news)	Self-supervised	300
p-mean (monolingual)	–	3600
FastText (BoW, Common Crawl)	Self-supervised	300
GloVe (BoW, Common Crawl)	Self-supervised	300
USE (DAN)	Supervised	512
USE (Transformer)	Supervised	512
InferSent (AllNLI)	Supervised	4096
Skip-Thought	Self-supervised	4800

Downstream Tasks

The downstream tasks from this paper are taken from the SentEval package. They feature five groups of tasks that were identified as key tasks that would be useful for a sentence embedding to help with. The five groups are: binary and multi-class classification, entailment and semantic relatedness, semantic textual similarity, paraphrase detection, and caption-image retrieval.

The categories give insights on what kind of tasks we are attempting to use these embeddings with (and if you're curious, you should check out the package), however, they incorporate all the classic tasks like sentiment analysis, question type analysis, sentence inference, and more.

The full list of classification tasks with examples can be seen here:

Table 2: Downstream classification tasks description and samples.

Dataset	Task	Example	Output
Customer Reviews (CR) [19]	Sentiment analysis of customer products' reviews	We tried it out Christmas night and it worked great .	Positive
Multi-Perspective Question and Answering (MPQA) [39]	Evaluation of opinion polarity	Don't want	Negative
Movie Reviews (MR) [31]	Sentiment analysis of movie reviews	Too slow for a younger crowd , too shallow for an older one .	Negative
Stanford Sentiment Analysis 2 (SST-2) [36]	Sentiment analysis with two classes: Negative and Positive	Audrey Tautou has a knack for picking roles that magnify her [..]	Positive
Stanford Sentiment Analysis 5 (SST-5) [36]	Sentiment analysis with 5 classes, that range from 0 (most negative) to 5 (most positive)	Nothing about this movie works	0
Subjectivity / Objectivity (SUBJ) [30]	Classify the sentence as Subjective or Objective	A movie that doesn't aim too high , but doesn't need to .	Subjective
Text REtrieval Conference (TREC) [38]	Question and answering	What are the twin cities ?	LOC:city

And semantic relatedness tasks can be seen here:

Dataset	Task	Sentence 1	Sentence 2	Output
Microsoft Common Objects in Context (COCO) [26]	Image-caption retrieval (ICR)	-	A group of people on some horses riding through the beach	Rank
Microsoft Research Paraphrase Corpus (MRPC) [12]	Classify whether a pair of sentences capture a paraphrase relationship	The procedure is generally performed in the second or third trimester.	The technique is used during the second and, occasionally, third trimester of pregnancy	Paraphase
Semantic Text Similarity (STS) [7]	To measure the semantic similarity between two sentences from 0 (not similar) to 5 (very similar)	Liquid ammonia leak kills 15 in Shanghai	Liquid ammonia leak kills at least 15 in Shanghai	4.6
Sentences Involving Compositional Knowledge Entailment (SICK-E) [27]	To measure semantics in terms of Entailment, Contradiction, or Neutral	A man is sitting on a chair and rubbing his eyes	There is no man sitting on a chair and rubbing his eyes	Contradiction
Sentences Involving Compositional Knowledge Semantic Relatedness (SICK-R) [27]	To measure the degree of semantic relatedness between sentences from 0 (not related) to 5 (related)	A man is singing a song and playing the guitar	A man is opening a package that contains headphones	1.6
Stanford Natural Language Inference (SNLI) [5]	To measure semantics in terms of Entailment, Contradiction, or Neutral	A small girl wearing a pink jacket is riding on a carousel	The carousel is moving	Entailment

To evaluate the benefits of these embeddings, simple models were used. That means simple multi-layer perceptrons with 50 neurons, logistic regression, or other very basic models. No fancy CNNs here. No RNNs. Just basic models to see how these representations fair.

Linguistic Tasks

Again taken from SentEval, there were 10 probing tasks that were conducted to evaluate different linguistic properties of sentence embeddings. These are pretty cool. They were:

- Bigram Shift: Whether or not two words were inverted
- Coordinate Inversion: Given two coordinate clauses, are they inverted?
- Object Number: Is the object singular or plural?
- Sentence Length
- Semantic Odd Man Out: Random noun/verb may be replaced. Detect if it has been.
- Subject Number: Is the subject singular or plural?
- Past Tense: Is main verb past or present tense?
- Top-Constituents: What is the class of the top syntax pattern?
- Depth of Syntactic Tree: How deep is the parsed syntax tree?
- Word Content: Which one of one thousand words is encoded in the sentence?

The idea here was that a sentence embedding shouldn't just perform well on downstream tasks, it should also encode some of these key linguistic properties as they will help yield a smart and explainable embedding!

This is all summarized in this table:

Task	Description	Example	Output
Bigram Shift (BShift)	Whether two words (tokens) in a sentence have been inverted	This is my Eve Christmas .	Inverted
Coordination Inversion (CoordInv)	Sentences comprised of two coordinate clauses. Detect whether clauses are inverted	I returned to my work , and Lisa headed for her office .	Inverted
Object Number (ObjNum)	Number of the direct object in the main clause (singular and plural)	He received the 200 points .	NNS (Plural)
Sentence Length (SentLen)	Predict the sentence length among 6 classes, which are length intervals	I can 't wait to show you and Mr. Taylor .	9 – 12 words
Semantic Odd Man Out (SOMO)	Random noun or verb replaced in the sentence by another noun or verb. Detect whether the sentence has been modified	Tomas surmised as well .	Changed
Subject Number (SubjNum)	Number of the subject in the main clause (singular and plural)	If there was ever a time to let loose , this vacation would have to be it .	Singular
Past Present (Tense)	Whether the main verb in the sentence is in the past or present tense	She smiled at him , her eyes alight with love .	Present
Top-Constituent (TopConst)	Classification task, where the classes are given by the 19 most common top-constituent sequences in the corpus	Did he buy anything from Troy ?	VBD_NP_VP_
Depth of Syntactic Tree (TreeDepth)	Predict the maximum depth of the syntactic tree of the sentence	The leaves were in various of stages of life .	10
Word Content (WC)	Predict which of the target words (among 1000) appear in the sentence	She eyed him skeptically .	eyed

And the Results???

Well... There was no clear winner!

ELMo performed the best on 5/9 tasks. USE did well on product review and question classification tasks. InferSent did well on paraphrase detection and entailment tasks.

Although p-mean didn't surpass top performers, it did surpass all baseline word embeddings like word2vec, GloVe, and FastText.

Here are the downstream results on classification:

Approach	CR	MPQA	MR	MRPC	SICK-E	SST-2	SST-5	SUBJ	TREC
Baseline									
Random Embedding	61.16	68.41	48.75	64.35	54.94	49.92	24.48	49.83	18.00
Experiments									
ELMo (BoW, all layers, 5.5B)	83.95	**91.02**	**80.91**	72.93	82.36	**86.71**	47.60	**94.69**	93.60
ELMo (BoW, all layers, original)	85.11	89.55	79.72	71.65	81.86	86.33	**48.73**	94.32	93.40
ELMo (BoW, top layer, original)	84.13	89.30	79.36	70.20	79.64	85.28	47.33	94.06	93.40
Word2Vec (BoW, google news)	79.23	88.24	77.44	73.28	79.09	80.83	44.25	90.98	83.60
p-mean (monolingual)	80.82	89.09	78.34	73.22	83.52	84.07	44.89	92.63	88.40
FastText (BoW, common crawl)	79.63	87.99	78.03	74.49	79.28	83.31	44.34	92.19	86.20
GloVe (BoW, common crawl)	78.67	87.90	77.63	73.10	79.01	81.55	45.16	91.48	84.00
USE (DAN)	80.50	83.53	74.03	71.77	80.39	80.34	42.17	91.93	89.60
USE (Transformer)	**86.04**	86.99	80.20	72.29	83.32	86.05	48.10	93.74	**93.80**
InferSent (AllNLI)	83.58	89.02	80.02	**74.55**	**86.44**	83.91	47.74	92.41	89.80
SkipThought	81.03	87.06	76.60	73.22	84.33	81.77	44.80	93.33	91.00

And on semantic relatedness:

Approach	SICK-R	STS-12	STS-13	STS-14	STS-15	STS-16	STSBenchmark
Experiments							
ELMo (BoW, all layers, 5.5B)	0.84	0.55	0.53	0.63	0.68	0.60	0.67
ELMo (BoW, all layers, original)	0.84	0.55	0.51	0.63	0.69	0.64	0.65
ELMo (BoW, top layer, original)	0.81	0.54	0.49	0.62	0.67	0.63	0.62
Word2Vec (BoW, google news)	0.80	0.52	0.58	0.66	0.68	0.65	0.64
p-mean (monolingual)	0.86	0.54	0.52	0.63	0.66	0.67	0.72
FastText (BoW, common crawl)	0.82	0.58	0.58	0.65	0.68	0.64	0.70
GloVe (BoW, common crawl)	0.80	0.52	0.50	0.55	0.56	0.51	0.65
USE (DAN)	0.84	0.59	0.59	0.68	0.72	0.70	0.76
USE (Transformer)	0.86	**0.61**	**0.64**	**0.71**	**0.74**	**0.74**	**0.78**
InferSent (AllNLI)	**0.89**	**0.61**	0.56	0.68	0.71	0.71	0.77
Skip-Thought	0.86	0.41	0.29	0.40	0.46	0.52	0.75

Information retrieval was another big test (which is scored based on how many correct results were retrieved in the top n, where n is an integer). InferSent actually performed the best, though ELMo and p-mean were other close contenders:

	Caption Retrieval				Image Retrieval			
Approach	R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
ELMo (BoW, all layers, 5.5B)	41.14	74.68	85.82	2.0	31.65	67.75	82.14	3.0
ELMo (BoW, all layers, original)	38.98	74.08	85.52	2.0	31.46	67.26	82.05	3.0
ELMo (BoW, top layer, original)	35.42	70.32	83.10	2.6	29.04	64.43	79.76	3.0
Word2Vec (BoW, google news)	33.82	66.56	80.32	2.8	27.18	61.91	77.77	3.8
p-mean (monolingual)	39.18	73.40	85.22	2.0	31.34	67.11	82.02	3.0
FastText (BoW, common crawl)	33.96	68.26	81.88	2.8	27.71	62.68	78.57	3.2
GloVe (BoW, common crawl)	33.96	66.08	79.42	2.8	26.70	61.18	77.35	3.8
USE (DAN)	29.04	62.08	76.50	3.4	23.37	57.63	74.61	4.0
USE (Transformer)	33.48	66.74	80.42	3.0	26.96	62.34	78.33	3.4
InferSent (AllNLI)	**42.14**	**75.78**	**87.08**	2.0	**33.44**	**69.50**	**83.48**	3.0
SkipThought	37.66	71.02	84.06	2.6	30.67	65.74	80.98	3.0

As for linguistic probing, ELMo again faired well though other representations were not that far behind. The results on all linguistic probing tasks can be seen here:

set.

Approach	BShift	CoordInv	ObjNum	SentLen	SOMO	SubjNum	Tense	TopConst	TreeDepth	WC
Baseline										
Random Embedding	50.16	51.38	50.82	17.07	50.44	50.79	50.02	4.71	17.57	0.12
Experiments										
ELMo (BoW, all layers, 5.5B)	**85.23**	69.92	**89.06**	**89.28**	**59.20**	**91.16**	89.73	84.50	**48.62**	88.86
ELMo (BoW, all layers, original)	84.29	69.44	88.65	89.03	58.20	90.18	90.33	**84.96**	48.32	89.90
ELMo (BoW, top layer, original)	81.18	68.47	87.61	78.20	58.64	90.16	88.78	81.54	44.97	72.78
Word2Vec (BoW, google news)	40.89	53.24	80.03	53.03	54.29	81.34	86.20	63.14	28.74	90.20
p-mean (monolingual)	50.09	50.45	83.27	86.42	53.27	81.73	88.18	61.66	38.20	**98.85**
FastText (BoW, common crawl)	50.28	53.87	80.08	66.97	55.21	80.66	87.41	67.10	36.72	91.09
GloVe (BoW, common crawl)	49.52	55.28	78.00	73.00	54.21	79.75	85.52	66.20	36.30	88.69
USE (DAN)	60.19	54.28	69.04	57.89	55.01	71.94	80.43	60.21	25.90	60.06
USE (Transformer)	60.52	58.19	74.60	79.84	58.48	77.78	86.15	68.73	30.49	54.19
InferSent (AllNLI)	56.64	68.34	80.54	84.13	55.79	84.45	86.74	78.34	41.02	95.18
SkipThought	70.19	**71.89**	83.55	86.03	54.74	86.06	**90.05**	82.77	41.22	79.64

So what does this mean? Well, as the authors state, it means we aren't at a place where we truly have a solid universal sentence encoder. There is no sentence embedding that performs best on every task and there's still a lot we can learn through linguistic probing and testing.

To me, this means that the field is still ripe for exploring! We haven't yet achieved a fantastic sentence embedding, though these are all solid models. Now it's on us to get out there and try out our own sentence embedding ideas!

If you enjoyed this article or found it helpful in any way, why not pass me along a dollar or two to help fund my machine learning education and research! Every dollar helps me get a little closer and I'm forever grateful.

Share

Author
Hunter Heidenreich
CS Undergrad at Drexel University, interested in AI and ethics

Comments