Skip-Thought Vectors

Introduction

- Several approaches have been developed for learning composition operators that map word vectors to sentence vectors (RNN, CNN, RCNN)
- All of these methods produce sentence representations that are passed to a supervised task and depend on a class label in order to bp through the composition weights
- These methods learn high-quality sentence representations but are tuned only for their respective task

Introduction

- Encode a sentence to predict the sentences around it

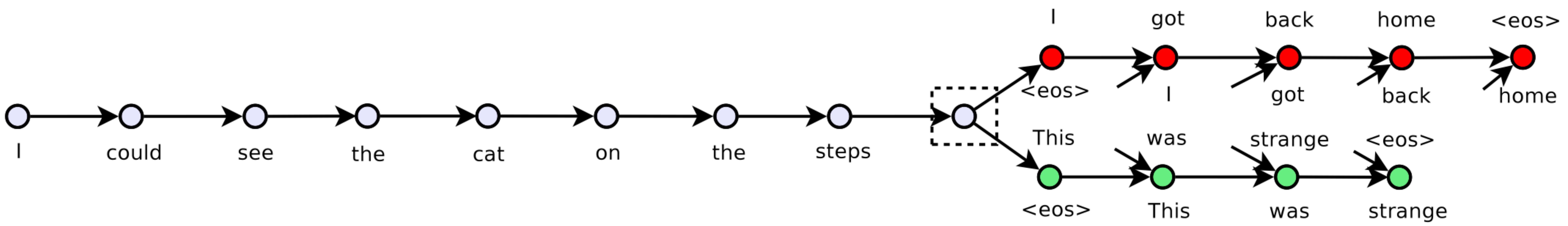
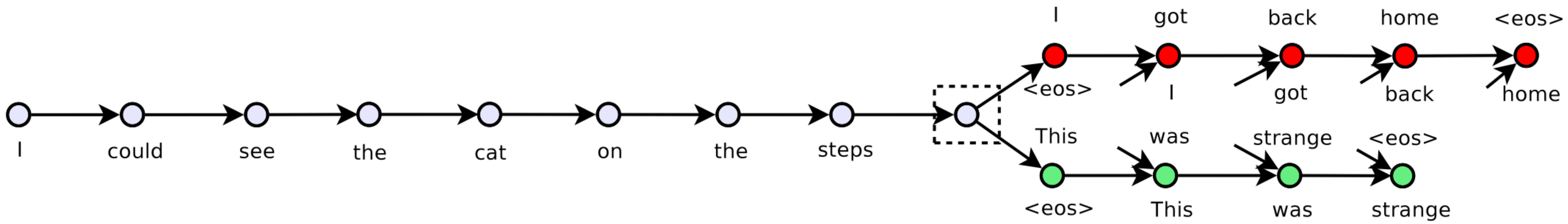


Figure 1: The skip-thoughts model. Given a tuple (s_{i-1}, s_i, s_{i+1}) of contiguous sentences, with s_i the i-th sentence of a book, the sentence s_i is encoded and tries to reconstruct the previous sentence s_{i-1} and next sentence s_{i+1}. In this example, the input is the sentence triplet *I got back home. I could see the cat on the steps. This was strange.* Unattached arrows are connected to the encoder output. Colors indicate which components share parameters. $\langle$eos$\rangle$ is the end of sentence token.

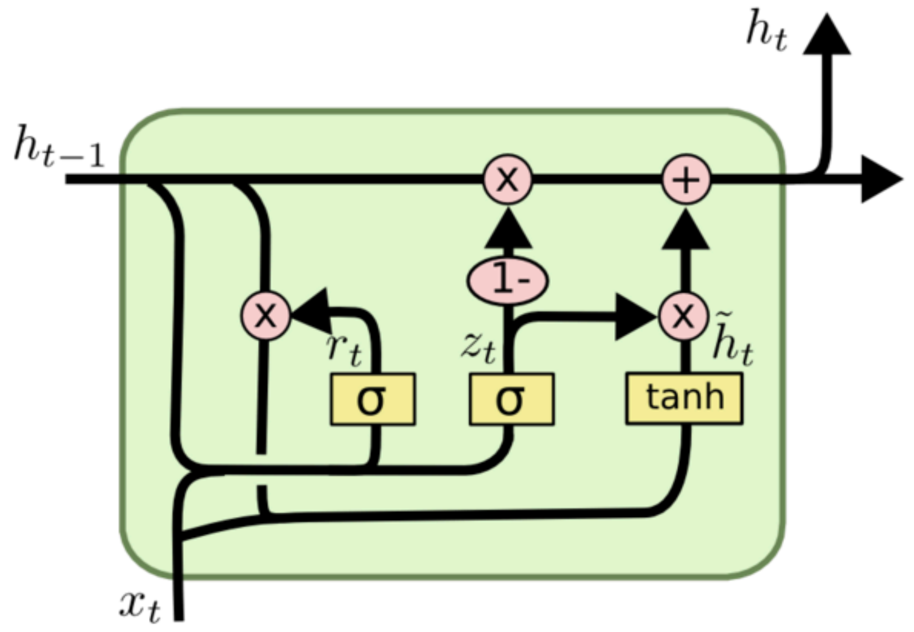
Inducing skip-thought vectors

- An encoder maps words to a sentence vector and a decoder is used to generate the surrounding sentences.
- GRU-RNN => encoder & decoder



Encoder

Encoder. Let $w_i^1, \ldots, w_i^N$ be the words in sentence s_i where N is the number of words in the sentence. At each time step, the encoder produces a hidden state $\mathbf{h}_i^t$ which can be interpreted as the representation of the sequence $w_i^1, \ldots, w_i^t$. The hidden state $\mathbf{h}_i^N$ thus represents the full sentence. To encode a sentence, we iterate the following sequence of equations (dropping the subscript i):



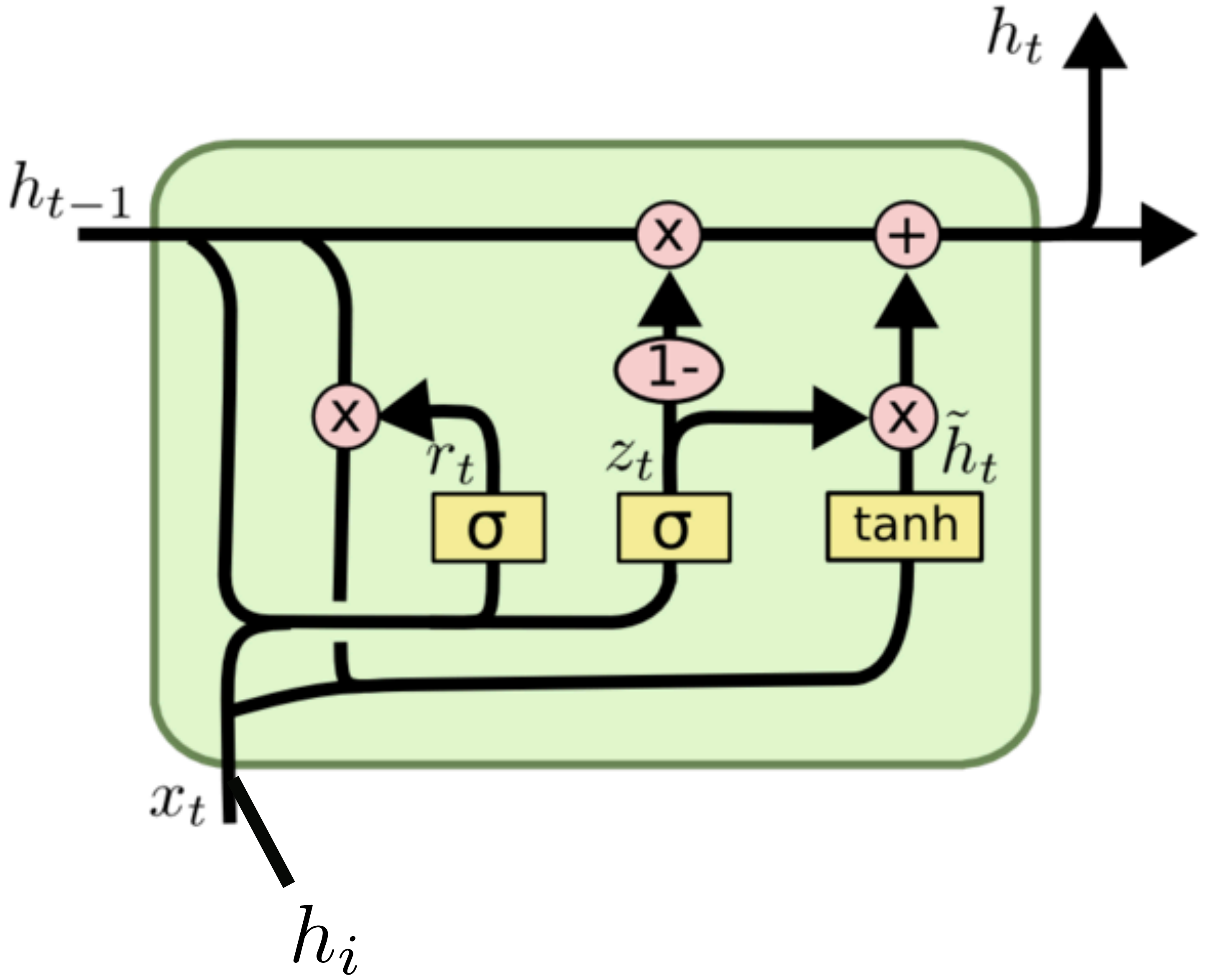
$$\mathbf{r}^t = \sigma(\mathbf{W}_r \mathbf{x}^t + \mathbf{U}_r \mathbf{h}^{t-1}) \tag{1}$$

$$\mathbf{z}^t = \sigma(\mathbf{W}_z \mathbf{x}^t + \mathbf{U}_z \mathbf{h}^{t-1}) \tag{2}$$

$$\bar{\mathbf{h}}^t = \tanh(\mathbf{W} \mathbf{x}^t + \mathbf{U}(\mathbf{r}^t \odot \mathbf{h}^{t-1})) \tag{3}$$

$$\mathbf{h}^t = (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \tag{4}$$

where $\bar{\mathbf{h}}^t$ is the proposed state update at time t, $\mathbf{z}^t$ is the update gate, $\mathbf{r}_t$ is the reset gate $(\odot)$ denotes a component-wise product. Both update gates takes values between zero and one.



Decoder

$$\mathbf{r}^t = \sigma(\mathbf{W}_r^d \mathbf{x}^{t-1} + \mathbf{U}_r^d \mathbf{h}^{t-1} + \mathbf{C}_r \mathbf{h}_i) \quad (5)$$

$$\mathbf{z}^t = \sigma(\mathbf{W}_z^d \mathbf{x}^{t-1} + \mathbf{U}_z^d \mathbf{h}^{t-1} + \mathbf{C}_z \mathbf{h}_i) \quad (6)$$

$$\bar{\mathbf{h}}^t = \tanh(\mathbf{W}^d \mathbf{x}^{t-1} + \mathbf{U}^d (\mathbf{r}^t \odot \mathbf{h}^{t-1}) + \mathbf{C}\mathbf{h}_i) \quad (7)$$

$$\mathbf{h}_{i+1}^t = (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \quad (8)$$

Given $\mathbf{h}_{i+1}^t$, the probability of word w_{i+1}^t given the previous $t-1$ words and the encoder vector is

$$P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) \propto \exp(\mathbf{v}_{w_{i+1}^t} \mathbf{h}_{i+1}^t) \quad (9)$$

where $\mathbf{v}_{w_{i+1}^t}$ denotes the row of $\mathbf{V}$ corresponding to the word of w_{i+1}^t. An analogous computation is performed for the previous sentence s_{i-1}.

Objective

Objective. Given a tuple (s_{i-1}, s_i, s_{i+1}), the objective optimized is the sum of the log-probabilities for the forward and backward sentences conditioned on the encoder representation:

$$\sum_t \log P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) + \sum_t \log P(w_{i-1}^t | w_{i-1}^{<t}, \mathbf{h}_i) \quad (10)$$

Vocabulary expansion

$$f : \mathcal{V}_{w2v} \rightarrow \mathcal{V}_{rnn} \qquad \mathbf{v}' = \mathbf{W}\mathbf{v} \qquad \mathbf{v} \in \mathcal{V}_{w2v} \text{ and } \mathbf{v}' \in \mathcal{V}_{rnn}$$

- learned linear mappings between translation word spaces, we solve an un-regularized L2 linear regression loss for the matrix W.
- Thus, any word from V_w2v can now be mapped into V_rnn for encoding sentences

Vocabulary expansion

choreograph	**modulation**	**vindicate**	**neuronal**	**screwy**	**Mykonos**	**Tupac**
choreography	transimpedance	vindicates	synaptic	wacky	Glyfada	2Pac
choreographs	harmonics	exonerate	neural	nutty	Santorini	Cormega
choreographing	Modulation	exculpate	axonal	iffy	Dubrovnik	Biggie
rehearse	##QAM	absolve	glial	loopy	Seminyak	Gridlock'd
choreographed	amplitude	undermine	neuron	zany	Skiathos	Nas
Choreography	upmixing	invalidate	apoptotic	kooky	Hersonissos	Cent
choreographer	modulations	refute	endogenous	dodgy	Kefalonia	Shakur

Table 3: Nearest neighbours of words after vocabulary expansion. Each query is a word that does not appear in our 20,000 word training vocabulary.

Dataset

- **BookCorpus**
- These are free books written by ye unpublished authors
- The dataset has books in 16 different genres

# of books	# of sentences	# of words	# of unique words	mean # of words per sentence
11,038	74,004,228	984,846,357	1,316,420	13

Experiments

We extract and evaluate our vectors with linear models on 8 tasks:

- semantic relatedness
- paraphrase detection
- image-sentence ranking
- question-type classification
- 4 benchmark sentiment and subjectivity datasets.

Experiments

- Using the learned encoder as a feature extractor, extract skip-thought vectors for all sentences.
- If the task involves computing scores between pairs of sentences, compute component-wise features between pairs. This is described in more detail specifically for each experiment.
- Train a linear classifier on top of the extracted features, with no additional fine-tuning or back-propagation through the skip-thoughts model.

Features Vectors

- uni-skip: unidirectional encoder with 2400 dimensions
- bi-skip: a bidirectional model with forward and backward encoders of 1200 dimensions each. The outputs are then concatenated to form a 2400 dimensions
- combine-skip: the concatenation of the vectors from uni-skip and bi-skip, resulting in a 4800 dimensions vector

Semantic relatedness

Method	r	ρ	MSE
Illinois-LH [18]	0.7993	0.7538	0.3692
UNAL-NLP [19]	0.8070	0.7489	0.3550
Meaning Factory [20]	0.8268	0.7721	0.3224
ECNU [21]	0.8414	–	–
Mean vectors [22]	0.7577	0.6738	0.4557
DT-RNN [23]	0.7923	0.7319	0.3822
SDT-RNN [23]	0.7900	0.7304	0.3848
LSTM [22]	0.8528	0.7911	0.2831
Bidirectional LSTM [22]	0.8567	0.7966	0.2736
Dependency Tree-LSTM [22]	**0.8676**	**0.8083**	**0.2532**
uni-skip	0.8477	0.7780	0.2872
bi-skip	0.8405	0.7696	0.2995
combine-skip	0.8584	0.7916	0.2687
combine-skip+COCO	0.8655	0.7995	0.2561

Method	Acc	F1
feats [24]	73.2	
RAE+DP [24]	72.6	
RAE+feats [24]	74.2	
RAE+DP+feats [24]	76.8	83.6
FHS [25]	75.0	82.7
PE [26]	76.1	82.7
WDDP [27]	75.6	83.0
MTMETRICS [28]	**77.4**	**84.1**
uni-skip	73.0	81.9
bi-skip	71.2	81.2
combine-skip	73.0	82.0
combine-skip + feats	75.8	83.0

Table 4: **Left:** Test set results on the SICK semantic relatedness subtask. The evaluation metrics are Pearson's r, Spearman's ρ, and mean squared error. The first group of results are SemEval 2014 submissions, while the second group are results reported by [22]. **Right:** Test set results on the Microsoft Paraphrase Corpus. The evaluation metrics are classification accuracy and F1 score. Top: recursive autoencoder variants. Middle: the best published results on this dataset.

Classification benchmarks

Method	MR	CR	SUBJ	MPQA	TREC
NB-SVM [41]	79.4	81.8	93.2	86.3	
MNB [41]	79.0	80.0	93.6	86.3	
cBoW [6]	77.2	79.9	91.3	86.4	87.3
GrConv [6]	76.3	81.3	89.5	84.5	88.4
RNN [6]	77.2	82.3	93.7	90.1	90.2
BRNN [6]	82.3	82.6	94.2	90.3	91.0
CNN [4]	81.5	85.0	93.4	89.6	**93.6**
AdaSent [6]	**83.1**	**86.3**	**95.5**	**93.3**	92.4
Paragraph-vector [7]	74.8	78.1	90.5	74.2	91.8
uni-skip	75.5	79.3	92.1	86.9	91.4
bi-skip	73.9	77.9	92.5	83.3	89.4
combine-skip	76.5	80.1	93.6	87.1	92.2
combine-skip + NB	80.4	81.3	93.6	87.5	

Table 7: Classification accuracies on several standard benchmarks. Results are grouped as follows: (a): bag-of-words models; (b): supervised compositional models; (c) Paragraph Vector (unsupervised learning of sentence representations); (d) ours. Best results overall are **bold** while best results outside of group (b) are underlined.

Generation

she grabbed my hand . “ come on . ” she fluttered her bag in the air . “ i think we ’re at your place . i ca n’t come get you . ” he locked himself back up . “ no . she will . ” kyrian shook his head . “ we met ... that congratulations ... said no . ” the sweat on their fingertips ’s deeper from what had done it all of his flesh hard did n’t fade . cassie tensed between her arms suddenly grasping him as her sudden her senses returned to its big form . her chin trembled softly as she felt something unreadable in her light . it was dark . my body shook as i lost what i knew and be betrayed and i realize just how it ended . it was n’t as if i did n’t open a vein . this was all my fault , damaged me . i should have told toby before i was screaming . i should ’ve told someone that was an accident . never helped it . how can i do this , to steal my baby ’s prints ? ”

Conclusion

Many variations have yet to be explored

- deep encoders and decoders
- larger context windows
- encoding and decoding paragraphs,
- other encoders, such as convnets