# Continual Learning for Sentence Representations Using Conceptors

**Tianlin Liu**
Department of Computer Science and
Electrical Engineering
Jacobs University Bremen
28759 Bremen, Germany
t.liu@jacobs-university.de

**Lyle Ungar** and **João Sedoc**
Department of Computer and
Information Science
University of Pennsylvania
Philadelphia, PA 19104
{ungar, joao}@cis.upenn.edu

## Abstract

Distributed representations of sentences have become ubiquitous in natural language processing tasks. In this paper, we consider a continual learning scenario for sentence representations: Given a sequence of corpora, we aim to optimize the sentence encoder with respect to the new corpus while maintaining its accuracy on the old corpora. To address this problem, we propose to initialize sentence encoders with the help of corpus-independent features, and then sequentially update sentence encoders using Boolean operations of *conceptor* matrices to learn corpus-dependent features. We evaluate our approach on semantic textual similarity tasks and show that our proposed sentence encoder can continually learn features from new corpora while retaining its competence on previously encountered corpora.

## 1 Introduction

Distributed representations of sentences are essential for a wide variety of natural language processing (NLP) tasks. Although recently proposed sentence encoders have achieved remarkable results (e.g., (Yin and Schütze, 2015; Arora et al., 2017; Cer et al., 2018; Pagliardini et al., 2018)), most, if not all, of them are trained on *a priori* fixed corpora. However, in open-domain NLP systems such as conversational agents, oftentimes we are facing a dynamic environment, where training data are accumulated sequentially over time and the distributions of training data vary with respect to external input (Lee, 2017; Mathur and Singh, 2018). To effectively use sentence encoders in such systems, we propose to consider the following *continual sentence representation learning task*: Given a sequence of corpora, we aim to train sentence encoders such that they can continually learn features from new corpora while retaining strong performance on previously encountered corpora.

Toward addressing the continual sentence representation learning task, we propose a simple sentence encoder that is based on the summation and linear transform of a sequence of word vectors aided by matrix conceptors. Conceptors have their origin in reservoir computing (Jaeger, 2014) and recently have been used to perform continual learning in deep neural networks (He and Jaeger, 2018). Here we employ Boolean operations of conceptor matrices to update sentence encoders over time to meet the following desiderata:

1. *Zero-shot learning*. The initialized sentence encoder (no training corpus used) can effectively produce sentence embeddings.

2. *Resistant to catastrophic forgetting*. When the sentence encoder is adapted on a new training corpus, it retains strong performances on old ones.

The rest of the paper is organized as follows. We first briefly review a family of linear sentence encoders. Then we explain how to build upon such sentence encoders for continual sentence representation learning tasks, which lead to our proposed algorithm. Finally, we demonstrate the effectiveness of the proposed method using semantic textual similarity tasks.[1]

**Notation** We assume each word $w$ from a vocabulary set $V$ has a real-valued word vector $v_w \in \mathbb{R}^n$. Let $p(w)$ be the monogram probability of a word $w$. A corpus $D$ is a collection of sentences, where each sentence $s \in D$ is a multiset of words (word order is ignored here). For a collection of vectors $Y = \{y_i\}_{i \in I}$, where $y_i \in \mathbb{R}^l$

[1] Our codes are available on GitHub https://github.com/liutianlin0121/contSentEmbed

*Proceedings of NAACL-HLT 2019*, pages 3274–3279
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

for $i$ in an index set $I$ with cardinality $|I|$, we let $[y_i]_{i\in I} \in \mathbb{R}^{l\times|I|}$ be a matrix whose columns are vectors $y_1, \cdots, y_{|I|}$. An identity matrix is denoted by $\mathbf{I}$.

## 2 Linear sentence encoders

We briefly overview "linear sentence encoders" that are based on linear algebraic operations over a sequence of word vectors. Among different linear sentence encoders, the smoothed inverse frequency (SIF) approach (Arora et al., 2017) is a prominent example – it outperforms many neural-network based sentence encoders on a battery of NLP tasks (Arora et al., 2017).

Derived from a generative model for sentences, the SIF encoder (presented in Algorithm 1) transforms a sequence of word vectors into a sentence vector with three steps. First, for each sentence in the training corpus, SIF computes a weighted average of word vectors (line 1-3 of Algorithm 1); next, it estimates a "common discourse direction" of the training corpus (line 4 of Algorithm 1); thirdly, for each sentence in the testing corpus, it calculates the weighted average of the word vectors and projects the averaged result away from the learned common discourse direction (line 5-8 of Algorithm 1). Note that this 3-step paradigm is slightly more general than the original one presented in (Arora et al., 2017), where the training and the testing corpus is assumed to be the same.

```
Algorithm 1: SIF sentence encoder.
  Input  : A training corpus $D$; a testing
           corpus $G$; parameter $a$, monogram
           probabilities $\{p(w)\}_{w\in V}$ of words
1 for sentence $s \in D$ do
2     $q_s \leftarrow \frac{1}{|s|}\sum_{w\in s}\frac{a}{p(w)+a}v_w$
3 end
4 Let $u$ be the first singular vector of $[q_s]_{s\in D}$.
5 for sentence $s \in G$ do
6     $q_s \leftarrow \frac{1}{|s|}\sum_{w\in s}\frac{a}{p(w)+a}v_w$
7     $f_s^{\text{SIF}} \leftarrow q_s - uu^\top q_s$.
8 end
  Output: $\{f_s^{\text{SIF}}\}_{s\in G}$
```

Building upon SIF, recent studies have proposed further improved sentence encoders (Khodak et al., 2018; Pagliardini et al., 2018; Yang et al., 2018). These algorithms roughly share the core procedures of SIF, albeit using more refined methods (e.g., softly remove more than one common discourse direction).

## 3 Continual learning for linear sentence encoders

In this section, we consider how to design a linear sentence encoder for continual sentence representation learning. We observe that common discourse directions used by SIF-like encoders are estimated from the training corpus. However, incrementally estimating common discourse directions in continual sentence representation learning tasks might not be optimal. For example, consider that we are sequentially given training corpora of `tweets` and `news article`. When the first `tweets` corpus is presented, we can train a SIF sentence encoder using `tweets`. When the second `news article` corpus is given, however, we will face a problem on how to exploit the newly given corpus for improving the *trained* sentence encoder. A straightforward solution is to first combine the `tweets` and `news article` corpora and then train a new encoder from scratch using the combined corpus. However, this paradigm is not efficient or effective. It is not efficient in the sense that we will need to re-train the encoder from scratch every time a new corpus is added. Furthermore, it is not effective in the sense that the common direction estimated from scratch reflects a compromise between tweets and news articles, which might not be optimal for either of the stand-alone corpus. Indeed, it is possible that larger corpora will swamp smaller ones.

To make the common discourse learned from one corpus more generalizable to another, we propose to use the conceptor matrix (Jaeger, 2017) to characterize and update the common discourse features in a sequence of training corpora.

### 3.1 Matrix conceptors

In this section, we briefly introduce matrix conceptors, drawing heavily on (Jaeger, 2017; He and Jaeger, 2018; Liu et al., 2019). Consider a set of vectors $\{x_1, \cdots, x_n\}$, $x_i \in \mathbb{R}^N$ for all $i \in \{1, \cdots, n\}$. A conceptor matrix is a regularized identity map that minimizes

$$\frac{1}{n}\sum_{i=1}^{n}\|x_i - Cx_i\|_2^2 + \alpha^{-2}\|C\|_{\mathrm{F}}^2. \tag{1}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm and $\alpha^{-2}$ is a scalar parameter called *aperture*. It can be shown that $C$ has a closed form solution:

$$C = \frac{1}{n}XX^\top(\frac{1}{n}XX^\top + \alpha^{-2}I)^{-1}, \qquad (2)$$

where $X = [x_i]_{i\in\{1,\cdots,n\}}$ is a data collection matrix whose columns are vectors from $\{x_1, \cdots, x_n\}$. In intuitive terms, $C$ is a soft projection matrix on the linear subspace where the typical components of $x_i$ samples lie. For convenience in notation, we may write $C(X, \alpha)$ to stress the dependence on $X$ and $\alpha$.

Conceptors are subject to most laws of Boolean logic such as NOT $\neg$, AND $\wedge$ and OR $\vee$. For two conceptors $C$ and $B$, we define the following operations:

$$\neg C := \mathbf{I} - C, \qquad (3)$$

$$C \wedge B := (C^{-1} + B^{-1} - \mathbf{I})^{-1} \qquad (4)$$

$$C \vee B := \neg(\neg C \wedge \neg B) \qquad (5)$$

Among these Boolean operations, the OR operation $\vee$ is particularly relevant for our continual sentence representation learning task. It can be shown that $C \vee B$ is the conceptor computed from the union of the two sets of sample points from which $C$ and $B$ are computed. Note that, however, to calculate $C \vee B$, we only need to know two matrices $C$ and $B$ and do not have to access to the two sets of sample points from which $C$ and $B$ are computed.

### 3.2 Using conceptors to continually learn sentence representations

We now show how to sequentially characterize and update the common discourse of corpora using the Boolean operation of conceptors. Suppose that we are sequentially given $M$ training corpora $D^1, \cdots, D^M$, presented one after another. Without using any training corpus, we first initialize a conceptor which characterizes the corpus-independent common discourse features. More concretely, we compute $C^0 := C([v_w]_{w\in Z}, \alpha)$, where $[v_w]_{w\in Z}$ is a matrix of column-wisely stacked word vectors of words from a stop word list $Z$ and $\alpha$ is a hyper-parameter. After initialization, for each new training corpus $D^i$ ($i = 1, \cdots, M$) coming in, we compute a new conceptor $C^{\text{temp}} := C([q_s]_{s\in D^i}, \alpha)$ to characterize the common discourse features of corpus $D^i$, where those $q_s$ are defined in the SIF Algorithm 1. We can then use Boolean operations of conceptors to compute $C^i := C^{\text{temp}} \vee C^{i-1}$, which characterizes common discourse features from the new corpus as well as the old corpora. After all $M$ corpora are presented, we follow the SIF paradigm and use $C^M$ to remove common discourse features from (potentially unseen) sentences. The above outlined conceptor-aided (CA) continual sentence representation learning method is presented in Algorithm 2.

```
Algorithm 2: CA sentence encoder.
   Input : A sequence of $M$ training corpora
           $\mathcal{D} = \{D^1, \cdots, D^M\}$; a testing
           corpus $G$; hyper-parameters $a$ and $\alpha$;
           word probabilities $\{p(w)\}_{w\in V}$; stop
           word list $Z$.
1  $C^0 \leftarrow C([v_w]_{w\in Z}, \alpha)$ .
2  for corpus index $i = 1, \cdots, M$ do
3      for sentence $s \in D^i$ do
4          $q_s \leftarrow \frac{1}{|s|}\sum_{w\in s}\frac{a}{p(w)+a}v_w$
5      end
6      $C^{\text{temp}} \leftarrow C([q_s]_{s\in D^i}, \alpha)$
7      $C^i \leftarrow C^{\text{temp}} \vee C^{i-1}$
8  end
9  for $s \in G$ do
10     $q_s \leftarrow \frac{1}{|s|}\sum_{w\in s}\frac{a}{p(w)+a}v_w$
11     $f_s^{\text{CA}} \leftarrow q_s - C^M q_s$
12 end
   Output: $\{f_s^{\text{CA}}\}_{s\in G}$
```

A simple modification of Algorithm 2 yields a “zero-shot” sentence encoder that requires only pre-trained word embeddings and no training corpus: we can simply skip those corpus-dependent steps (line 2-8) and use $C^0$ in place of $C^M$ in line 11 in Algorithm 2 to embed sentences. This method will be referred to as “zero-shot CA.”

## 4 Experiment

We evaluated our approach for continual sentence representation learning using semantic textual similarity (STS) datasets (Agirre et al., 2012, 2013, 2014, 2015, 2016). The evaluation criterion for such datasets is the Pearson correlation coefficient (PCC) between the predicted sentence similarities and the ground-truth sentence similarities. We split these datasets into five corpora by their genre: news, captions, wordnet, forums, tweets (for details see appendix). Throughout this section, we use publicly available 300-

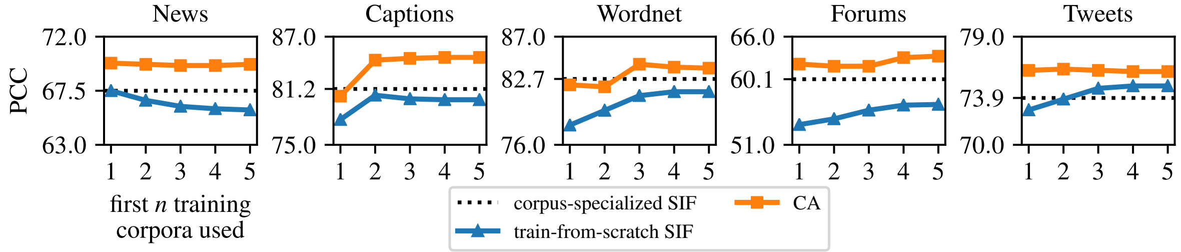

Figure 1: PCC results of STS datasets. Each panel shows the PCC results of a testing corpus (specified as a subtitle) as a function of increasing numbers of training corpora used. The setup of this experiment mimics (Zenke et al., 2017, section 5.1).

| | News | Captions | WordNet | Forums | Tweets |
|---|---|---|---|---|---|
| av. train-from-scratch SIF | 66.5 | 79.7 | 80.3 | 55.5 | 74.2 |
| zero-shot CA | 65.6 | 79.8 | 82.5 | 61.5 | 75.2 |
| av. CA | **69.7** | **83.8** | **83.2** | **62.5** | **76.2** |

Table 1: Time-course averaged PCC of train-from-scratch SIF and conceptor-aided (CA) methods, together with the result of zero-shot CA. Best results are in boldface and the second best results are underscored.

dimensional GloVe vectors (trained on the 840 billion token Common Crawl) (Pennington et al., 2014). Additional experiments with Word2Vec (Mikolov et al., 2013), Fasttext (Bojanowski et al., 2017), Paragram-SL-999 (Wieting et al., 2015) are in the appendix.

We use a standard continual learning experiment setup (cf. (Zenke et al., 2017, section 5.1)) as follows. We sequentially present the five training datasets in the order[2] of news, captions, wordnet, forums, and tweets, to train sentence encoders. Whenever a new training corpus is presented, we train a SIF encoder from scratch[3] (by combining all available training corpora which have been already presented) and then test it on each corpus. At the same time, we incrementally adapt a CA encoder[4] using the newly presented corpus and test it on each corpus. The lines of each panel of Figure 1 show the test results of SIF and CA on each testing corpus (specified as the panel subtitle) as a function of the number of training corpora used (the first $n$ corpora of news, captions, wordnet, forums, and tweets for this experiment). To give a concrete example, consider the blue line in the first panel of Figure 1. This line shows the test PCC scores ($y$-axis) of SIF encoder on the news corpus when the number of training corpora increases ($x$-axis). Specifically, the left-most blue dot indicates the test result of SIF encoder on news corpus when trained on news corpus itself (that is, the first training corpus is used); the second point indicates the test results of SIF encoder on news corpus when trained on news and captions corpora (i.e., the first *two* training corpora are used); the third point indicates the test results of SIF encoder on news corpus when trained on news, captions, and wordnet corpora (that is, the first *three* training corpora are used), so on and so forth. The dash-lines in panels show the results of a corpus-specialized SIF, which is trained and tested on the same corpus, i.e., as done in (Arora et al., 2017, section 4.1). We see that the PCC results of CA are better and more "forgetting-resistant" than train-from-scratch SIF throughout the time course where more training data are incorporated. Consider, for example, the test result of news corpus (first panel) again. As more and more training corpora are used, the performance of train-from-scratch SIF drops with a noticeable slope; by contrast, the performance CA drops only slightly.

As remarked in the section 3.2, with a simple modification of CA, we can perform zero-shot sentence representation learning without using any training corpus. The zero-shot learning results are

[2] The order can be arbitrary. Here we ordered the corpora from the one with the largest size (news) to the smallest size (tweets). The results from reversely ordered corpora are reported in the appendix.

[3] We use $a = 0.001$ as in (Arora et al., 2017). The word frequencies are available at the GitHub repository of SIF.

[4] We used hyper-parameter $\alpha = 1$. Other parameters are set to be the same as SIF.

presented in Table 1, together with the time-course averaged results of CA and train-from-scratch SIF (i.e., the averaged values of those CA or SIF scores in each panel of Figure 1). We see that the averaged results of our CA method performs the best among these three methods. Somewhat surprisingly, the results yielded by zero-shot CA are better than the averaged results of train-from-scratch SIF in most of the cases.

We defer additional experiments to the appendix, where we compared CA against more baseline methods and use different word vectors other than GloVe to carry out the experiments.

## 5 Conclusions and future work

In this paper, we formulated a continual sentence representation learning task: Given a consecutive sequence of corpora presented in a time-course manner, how can we extract useful sentence-level features from new corpora while retaining those from previously seen corpora? We identified that the existing linear sentence encoders usually fall short at solving this task as they leverage on "common discourse" statistics estimated based on a priori fixed corpora. We proposed two sentence encoders (CA encoder and zero-shot CA encoder) and demonstrate their the effectiveness at the continual sentence representation learning task using STS datasets.

As the first paper considering continual sentence representation learning task, this work has been limited in a few ways – it remains for future work to address these limitations. First, it is worthwhile to incorporate more benchmarks such as GLUE (Wang et al., 2019) and SentEval (Conneau and Kiela, 2018) into the continual sentence representation task. Second, this work only considers the case of linear sentence encoder, but future research can attempt to devise (potentially more powerful) non-linear sentence encoders to address the same task. Thirdly, the proposed CA encoder operates at a corpus level, which might be a limitation if boundaries of training corpora are ill-defined. As a future direction, we expect to lift this assumption, for example, by updating the common direction statistics at a sentence level using Autoconceptors (Jaeger, 2014, section 3.14). Finally, the continual learning based sentence encoders should be applied to downstream applications in areas such as open domain NLP systems.

## Acknowledgements

The authors thank anonymous reviewers for their helpful feedback. This work was partially supported by João Sedoc's Microsoft Research Dissertation Grant.

## References

E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigaua, L. Uriaa, and J. Wiebeg. 2015. Semeval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation*, pages 252–263.

E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation*, pages 81–91.

E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California.

E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. 2013. Sem 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics*, volume 1, pages 32–43.

E. Agirre, M. Diab, D. Cer, and A. Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval '12, pages 385–393, Stroudsburg, PA, USA. Association for Computational Linguistics.

S. Arora, Y. Liang, and T. Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

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

D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

A. Conneau and D. Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

X. He and H. Jaeger. 2018. Overcoming catastrophic interference using conceptor-aided backpropagation. In *International Conference on Learning Representations*.

H. Jaeger. 2014. Controlling recurrent neural networks by conceptors. Technical report, Jacobs University Bremen.

H. Jaeger. 2017. Using conceptors to manage neural long-term memories for temporal patterns. *Journal of Machine Learning Research*, 18(13):1–43.

M. Khodak, N. Saunshi, Y. Liang, T. Ma, B. Stewart, and S. Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. *In the Proceedings of ACL*.

S. Lee. 2017. Toward continual learning for conversational agents. Technical report, Microsoft Research AI - Redmond.

T. Liu, L. Ungar, and J. Sedoc. 2019. Unsupervised post-processing of word vectors via conceptor negation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-2019), Honolulu*.

V. Mathur and A. Singh. 2018. The rapidly changing landscape of conversational agents.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

M. Pagliardini, P. Gupta, and M. Jaggi. 2018. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *Proceedings of the NAACL 2018*.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

J. Wieting, M. Bansal, K. Gimpel, K. Livescu, and D. Roth. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Z. Yang, C. Zhu, and W. Chen. 2018. Zero-training sentence embedding via orthogonal basis.

W. Yin and H. Schütze. 2015. Convolutional neural network for paraphrase identification. In *Proceedings of the NAACL HLT 2015*, pages 901–911.

F. Zenke, B. Poole, and S. Ganguli. 2017. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3987–3995, International Convention Centre, Sydney, Australia. PMLR.