CON-S2V: A Generic Framework for Incorporating Extra-Sentential Context into Sen2Vec

Tanay Kumar Saha[1] Shafiq Joty[2] Mohammad Al Hasan[1]

[1]Indiana University Purdue University Indianapolis, Indianapolis, IN 46202, USA

[2]Nanyang Technological University, Singapore

September 22, 2017

Outline

Outline

Sen2Vec (Model for representation of Sentences)

- Learn distributed representation of sentences from unlabeled data
  - v_1: I eat rice $\rightarrow$ [0.2 0.3 0.4]
  - $\phi : V \rightarrow \mathbb{R}^d$
- For many text processing tasks that involve classification, clustering, or ranking of sentences, vector representation of sentences is a prerequisite
- Distributed Representation has been shown to perform better than Bag-of-Words (BOW) based vector representation
- Proposed by Mikolov et. al

CON-S2V (Our Model)

- A novel approach to learn distributed representation of sentences from unlabeled data by jointly modeling both content and context of a sentence
  - v_1: I have an NEC multisync 3D monitor for sale
  - v_2: Looks new
  - v_3: Great Condition
- In contrast to the existing works, we consider context sentences as atomic linguistic units.
- We consider two types of context: discourse and similarity. However, our model can take any arbitrary type of context
- Our evaluation on these tasks across multiple datasets shows impressive results for our model, which outperforms the best existing models by up to 7.7 F_1-score in classification, 15.1 V-score in clustering, 3.2 ROUGE-1 score in summarization.
- Build on top of Sen2Vec

Context Types of a Sentence

- Discourse Context of a Sentence
  - It is formed by the previous and the following sentences in the text
  - Adjacent sentences in a text are logically connected by certain coherence relations (e.g., elaboration, contrast) to express the meaning
  - Lactose is a milk sugar. The enzyme lactase breaks it down. Here, the second sentence is an elaboration of the first sentence.
- Similarity Context of a Sentence
  - Based on more direct measures of similarity
  - Considers relations between all possible sentences in a document and possibly across multiple documents

Related Work

- Sen2Vec
  - Uses Sentence ID as a special token and learn the representation of the sentence by predicting all the words in a sentence
  - For example, for a sentence, v_1 : I eat rice, it will learn representation for v_1 by learning to predict each of the words, i.e. I, eat, and rice correctly
  - Shown to perform better than tf-idf
- W2V-avg
  - Uses word vector averaging
  - A tough-to-beat baseline for most downstream tasks
- SDAE
  - Employs an encoder-decoder framework, similar to neural machine translation (NMT) to de-noise an original sentence (target) from its corrupted version (source)
  - SAE is similar in spirit to SDAE but does not corrupt source

Related Work

- C-Phrase
  - C-PHRASE is an extension of CBOW (Continuous Bag of Words Model)
  - The context of a word is extracted from a syntactic parse of the sentence
  - Syntax tree for a sentence, *A sad dog is howling in the park* is: (S (NP A sad dog) (VP is (VP howling (PP in (NP the park)))))
  - C-PHRASE will optimize context prediction for dog, sad dog, a sad dog, a sad dog is howling, etc., but not, for example, for howling in, as these two words do not form a *syntactic constituent* by themselves
  - Uses word vector addition for representing sentences

Related Work

- Skip-Thought (Context Sensitive)
  - Uses the NMT framework to predict adjacent sentences (target) given a sentence (source)
- FastSent (Context Sensitive)
  - An additive model to learn sentence representation from word vectors
  - It predicts the words of its adjacent sentences in addition to its own words

CON-S2V

- A novel model to learn distributed representation of sentences by considering content as well as context of a sentence
- It treats the context sentences as an atomic unit
- Efficient to train compared to *compositional* methods like encoder-decoder models (e.g., SDAE, Skip-Thought) that compose a sentence vector from the word vectors

Outline

Con-S2V Model

- The model for learning the vector representation of a sentence comprises three components
- The first component models the content by asking the sentence vector to predict its constituent words (modeling content)
- The second component models the distributional hypotheses of a context (modeling context)
- Third component models the proximity hypotheses of a context, which also suggests that sentences that are proximal should have similar representations (modeling context)

CON-S2V Model

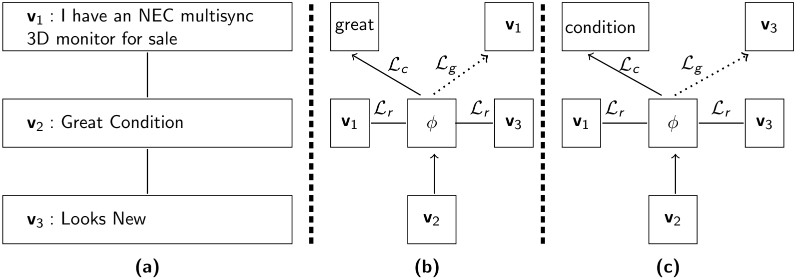

Figure: Two instances (see **(b)** and **(c)**) of our model for learning representation of sentence $\mathbf{v}_2$ within a context of two other sentences: $\mathbf{v}_1$ and $\mathbf{v}_3$ (see **(a)**). Directed and undirected edges indicate prediction loss and regularization loss, respectively, and dashed edges indicate that the node being predicted is randomly sampled. (Collected from: 20news-bydate-train/misc.forsale/74732. The central topic is "forsale".)

- We minimize the following loss function for learning representation of sentences:

$$J(\phi) = \sum_{\mathbf{v}_i \in V} \sum_{\substack{v \in \langle v_i \rangle_t^l \\ j \sim \mathcal{U}(1, C_i)}} \big[\mathcal{L}_c(\mathbf{v}_i, v) + \mathcal{L}_g(\mathbf{v}_i, \mathbf{v}_j) + \mathcal{L}_r(\mathbf{v}_i, \mathcal{N}(\mathbf{v}_i)) \big] \tag{1}$$

- $\mathcal{L}_c$: Modeling Content (First Component)
- $\mathcal{L}_g$: Modeling Context with Distributional Hypothesis (Second Component). The distributional hypothesis conveys that the sentences occurring in similar contexts should have similar representations
- $\mathcal{L}_r$: Modeling Context with Proximity Hypothesis (Third Component). Proximity hypotheses of a context, which also suggests that sentences that are proximal should have similar representations

Modeling Content

- Our approach for modeling content of a sentence is similar to the distributed bag-of-words (DBOW) model of Sen2Vec
- Given an input sentence $\mathbf{v}_i$, we first map it to a unique vector $\phi(\mathbf{v}_i)$ by looking up the corresponding vector in the sentence embedding matrix ϕ
- We then use $\phi(\mathbf{v}_i)$ to predict each word v sampled from a window of words in $\mathbf{v}_i$. Formally, the loss for modeling content using negative sampling is:

$$\mathcal{L}_c(\mathbf{v}_i, v) = -\log\sigma\left(\mathbf{w}_v^T\phi(\mathbf{v}_i)\right) - \log\sum_{s=1}^{S}\mathbb{E}_{v^s\sim\psi_c}\,\sigma\left(-\mathbf{w}_{v^s}^T\phi(\mathbf{v}_i)\right) \tag{2}$$

Modeling Distributional Similarity

- Our sentence-level distributional hypothesis is that if two sentences share many neighbors in the graph, their representations should be similar
- We formulate this in our model by asking the sentence vector to predict its neighboring nodes
- Formally, the loss for predicting a neighboring node $\mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i)$ using the sentence vector $\phi(\mathbf{v}_i)$ is:

$$\mathcal{L}_g(\mathbf{v}_i, \mathbf{v}_j) = -\log \sigma\left(\mathbf{w}_j^T \phi(\mathbf{v}_i)\right) - \log \sum_{s=1}^{S} \mathbb{E}_{j^s \sim \psi_g} \sigma\left(-\mathbf{w}_{j^s}^T \phi(\mathbf{v}_i)\right) \tag{3}$$

Modeling Proximity

- According to our proximity hypothesis, sentences that are proximal in their contexts, should have similar representations
- We use a Laplacian regularizer to model this
- The regularization loss for modeling proximity for a sentence $\mathbf{v}_i$ in its context $\mathcal{N}(\mathbf{v}_i)$ is

$$\mathcal{L}_r(\mathbf{v}_i, \mathcal{N}(\mathbf{v}_i)) = \frac{\lambda}{C_i} \sum_{\mathbf{v}_k \in \mathcal{N}(\mathbf{v}_i)} ||\phi(\mathbf{v}_i) - \phi(\mathbf{v}_k)||^2 \tag{4}$$

Training CON-S2V

Algorithm 1: Training CON-S2V with SGD

Input : set of sentences V, graph $G = (V, E)$

Output: learned sentence vectors ϕ

1. Initialize model parameters: ϕ and $\mathbf{w}$'s;
2. Compute noise distributions: ψ_c and ψ_g
3. **repeat**

  for *each sentence* $\mathbf{v}_i \in V$ **do**

    for *each content word* $v \in \mathbf{v}_i$ **do**

      a) Generate a positive pair $(\mathbf{v}_i, v)$ and S negative pairs $\{(\mathbf{v}_i, v^s)\}_{s=1}^{S}$ using ψ_c;

      b) Take a gradient step for $\mathcal{L}_c(\mathbf{v}_i, v)$;

      c) Sample a neighboring node $\mathbf{v}_j$ from $\mathcal{N}(\mathbf{v}_i)$;

      d) Generate a positive pair $(\mathbf{v}_i, \mathbf{v}_j)$ and S negative pairs $\{(\mathbf{v}_i, \mathbf{v}_j^s)\}_{s=1}^{S}$ using ψ_g;

      e) Take a gradient step for $\mathcal{L}_g(\mathbf{v}_i, \mathbf{v}_j)$;

      f) Take a gradient step for $\mathcal{L}_r(\mathbf{v}_i, \mathcal{N}(\mathbf{v}_i))$;

    end

  end

until *convergence*;

Training Details

- CON-S2V is trained with stochastic gradient descent (SGD), where the gradient is obtained via backpropagation
- The number of noise samples (S) in negative sampling was 5
- In all our models, the embeddings vectors (ϕ, ψ) were of 600 dimensions, which were initialized with random numbers sampled from a small uniform distribution, $\mathcal{U}(-0.5/d, 0.5/d)$
- The weight vectors ω's were initialized with zero

Outline

Evaluation Tasks and Dataset

- We evaluate CON-S2V on Summarization, Classification and Clustering Task
- CON-S2V learns representation of a sentence by exploiting contextual information in addition to the content
- For this reason, we did not evaluate our models on tasks (Sentiment Classification) previously used to evaluate sentence representation models
- For Classification and Clustering evaluation, it require a corpora of annotated sentences with ordering and document boundaries preserved, i.e., documents with sentence-level annotations

Evaluation Tasks (Summarization)

- The goal is to select the most important sentences to form an abridged version of the source document(s)
- We use the popular graph-based algorithm LexRank
- The input to LexRank is a graph, where nodes represent sentences and edges represent cosine similarity between *vector representations* (learned by models) of the two corresponding sentences
- We use the benchmark datasets from DUC-2001 and DUC-2002 dataset for evaluation

Dataset	#Doc.	#Avg. Sen.	#Avg. Sum.
DUC 2001	486	40	2.17
DUC 2002	471	28	2.04

Table: Basic statistics about the DUC datasets

Evaluation Tasks (Classification and Clustering)

- We evaluate our models by measuring how effective the learned vectors are when they are used as features for classifying or clustering the sentences into topics
- We use a MaxEnt classifier and a K-means++ clustering algorithm for classification and clustering tasks, respectively
- We use the standard text categorization corpora: *Reuters-21578* and *20-Newsgroups*.
- Reuters-21578 (henceforth Reuters) is a collection of 21,578 news documents covering 672 topics.
- 20-Newsgroups is a collection of about 20,000 news articles organized into 20 different topics.

Classification and Clustering (Generating Sentence-level Topic Annotations)

- One option is to assume that all the sentences of a document share the same topic label as the document
- This naive assumption induces a lot of noise
- Although sentences in a document collectively address a common topic, not all sentences are directly linked to that topic, rather they play supporting roles
- To minimize this noise, we employ our extractive summarizer to select the top 20% sentences of each document as representatives of the document, and assign them the same topic label as the topic of the document
- Note that the sentence vectors are learned independently from an entire dataset

DataSet Statistics for Classification and Clustering

Dataset	#Doc.	Total #sen.	Annot. #sen	Train #sen.	Test #sen.	#Class
Reuters	9,001	42,192	13,305	7,738	3,618	8
Newsgroups	7,781	95,809	22,374	10,594	9,075	8

Table: Statistics about Reuters and Newsgroups.

Metrics for Evaluation

- For Summarization, We use the widely used automatic evaluation metric ROUGE to evaluate the system-generated summaries.
- ROUGE computes *n*-gram recall between a system-generated summary and a set of human-authored reference summaries
- We report raw **acc**uracy, macro-averaged $\mathbf{F_1}$-score, and Cohen's $\boldsymbol{\kappa}$ for comparing classification performance
- For clustering, we report **V**-measure and adjusted mutual information or **AMI**

Models Compared

- Existing Distributed Models: Sen2Vec, W2V-avg, C-PHRASE, FastSent, and Skip-Thought
- Non-distributed Model: Tf-Idf
- Retrofitted Models: RET-dis, RET-sim
- Regularized Models: REG-dis, REG-sim: We compare with a variant of our model, where the loss to capture distributional similarity $\mathcal{L}_g(\mathbf{v}_i, \mathbf{v}_j)$ is turned off
- Our Model: CON-S2V-dis, CON-S2V-sim

Similarity Network Construction

- Our similarity context allows any other sentence in the corpus to be in the context of a sentence depending on how similar they are
- we first represent the sentences with vectors learned by Sen2Vec , then we measure the cosine distance between the vectors
- We restrict the context size of a sentence for computational efficiency
- First, we set thresholds for intra- and across-document connections: sentences in a document are connected only if their similarity value is above a pre-specified threshold δ, and sentences across documents are connected only if their similarity value is above another pre-specified threshold γ
- we allow up to 20 most similar neighbors. We call the resulting network *similarity network*

Optimal Parameter Settings

- For each dataset that we describe earlier, we randomly selected 20% documents from the training set to form a held-out validation set on which we tune the hyper-parameters
- we optimized F_1 for classification, AMI for clustering, and ROUGE-1 for summarization
- For RET-sim, and RET-dis, the number of iteration was set to 20
- For the similarity context, the intra- and across-document thresholds δ and γ were set to 0.5 and 0.8
- Optimal Parameter values are given in the following table:

Dataset	Task	Sen2Vec	FastSent (*win. size*)	W2V-avg	REG-sim (*win. size, reg. str.*)	REG-dis (*win. size, reg. str.*)	CON-S2V-sim (*win. size, reg. str.*)	CON-S2V-dis (*win. size, reg. str.*)
Reuters	clas.	8	10	10	(8, 1.0)	(8, 1.0)	(8, 0.8)	(8, 1.0)
	clus.	12	8	12	(12, 0.3)	(12, 1.0)	(12,0.8)	(12, 0.8)
Newsgroups	clas.	10	8	10	(10, 1.0)	(10, 1.0)	(10, 1.0)	(10, 1.0)
	clus.	12	12	12	(12, 1.0)	(12, 1.0)	(12, 0.8)	(10, 1.0)
DUC 2001	sum.	10	12	12	(10, 0.8)	(10, 0.5)	(10, 0.3)	(10, 0.3)
DUC 2002	sum.	8	8	10	(8, 0.8)	(8, 0.3)	(8, 0.3)	(8, 0.3)

Table: Optimal values of the hyper-parameters for different models on different tasks.

Outline

Classification and Clustering Performance

	Topic Classification Results						Topic Clustering Results			
	Reuters			*Newsgroups*			*Reuters*		*Newsgroups*	
	F_1	Acc	κ	F_1	Acc	κ	V	AMI	V	AMI
Sen2Vec	83.25	83.91	79.37	79.38	79.47	76.16	42.74	40.00	35.30	34.74
W2V-avg	(+) 2.06	(+) 1.91	(+) 2.51	(−) 0.42	(−) 0.44	(−) 0.50	(−) 11.96	(−) 10.18	(−) 17.90	(−) 18.50
C-PHRASE	(−) 2.33	(−) 2.01	(−) 2.78	(−) 2.49	(−) 2.38	(−) 2.86	(−) 11.94	(−) 10.80	(−) 1.70	(−) 1.44
FastSent	(−) 0.37	(−) 0.29	(−) 0.41	(−) 12.23	(−) 12.17	(−) 14.21	(−) 15.54	(−) 13.06	(−) 34.40	(−) 34.16
Skip-Thought	(−) 19.13	(−) 15.61	(−) 21.8	(−) 13.79	(−) 13.47	(−)15.76	(−) 29.94	(−) 28.00	(−) 27.50	(−) 27.04
Tf-Idf	(−) 3.51	(−) 2.68	(−) 3.85	(−) 9.95	(−) 9.72	(−) 11.55	(−) 21.34	(−) 20.14	(−) 29.20	(−) 30.60
RET-sim	(+) 0.92	(+) 1.28	(+) 1.65	(+) 2.00	(+) 1.97	(+) 2.27	(+) 3.72	(+) 3.34	(+) 5.22	(+) 5.70
RET-dis	(+) 1.66	(+) 1.79	(+) 2.30	(+) 5.00	(+) 4.91	(+) 5.71	(+) 4.56	(+) 4.12	(+) 6.28	(+) 6.76
REG-sim	(+) 2.53	(+) 2.53	(+) 3.28	(+) 3.31	(+) 3.29	(+) 3.81	(+) 4.76	(+) 4.40	(+) 12.78	(+) 12.18
REG-dis	(+) 2.52	(+) 2.43	(+) 3.17	(+) 5.41	(+) 5.34	(+) 6.20	(+) 7.40	(+) 6.82	(+) 12.54	(+) 12.44
CON-S2V-sim	(+) 3.83	(+) 3.55	(+) 4.62	(+) 4.52	(+) 4.50	(+) 5.21	**(+) 14.98**	**(+) 14.38**	(+) 13.68	(+) 13.56
CON-S2V-dis	**(+) 4.29**	**(+) 4.04**	**(+) 5.22**	**(+) 7.68**	**(+) 7.56**	**(+) 8.80**	(+) 9.30	(+) 8.36	**(+) 15.10**	**(+) 15.20**

Table: Performance of our models on topic classification and clustering tasks in comparison to Sen2Vec.

Summarization Performance

	DUC'01	DUC'02
Sen2Vec	43.88	54.01
W2V-avg	(−) 0.62	(+) 1.44
C-PHRASE	(+) 2.52	(+) 1.68
FastSent	(−) 4.15	(−) 7.53
Skip-Thought	(+) 0.88	(−) 2.65
Tf-Idf	**(+) 4.83**	(+) 1.51
RET-sim	(−) 0.62	(+) 0.42
RET-dis	(+) 0.45	(−) 0.37
REG-sim	(+) 2.90	(+) 2.02
REG-dis	(−) 1.92	(−) 8.77
CON-S2V-sim	(+) 3.16	**(+) 2.71**
CON-S2V-dis	(+) 1.15	(−) 4.46

Table: ROUGE-1 scores of the models on DUC datasets in comparison with Sen2Vec.

Outline

Conclusion and Future Work

- We have presented a novel model to learn distributed representation of sentences by considering content as well as context of a sentence
- One important property of our model is that it encodes a sentence directly, and it considers neighboring sentences as atomic units
- Apart from the improvements that we achieve in various tasks, this property makes our model quite efficient to train compared to *compositional* methods like encoder-decoder models (e.g., SDAE, Skip-Thought) that compose a sentence vector from the word vectors

Conclusion and Future Work

- It would be interesting to see how our model compares with compositional models on sentiment classification task
- However, this would require creating a new dataset of comments with sentence-level sentiment annotations
- We intend to create such datasets and evaluate the models in the future