S2-P13 高性能な文類似度を用いた事前学習モデルの対照学習

山際宏明[1], 石橋陽一[1], 下平英寿[1,2] 1. 京都大学大学院 2. 理化学研究所革新知能総合研究センター

背景:対照学習（SimCSE）により，優れた文類似度性能を達成するBERTの埋め込みを獲得できる.
やったこと: 対照学習の際に使用する類似度はコサイン類似度．他の様々な類似度では性能が向上するか？
結果: 実験した類似度の中でコサイン類似度が最も優れていた.

SimCSE

対照学習

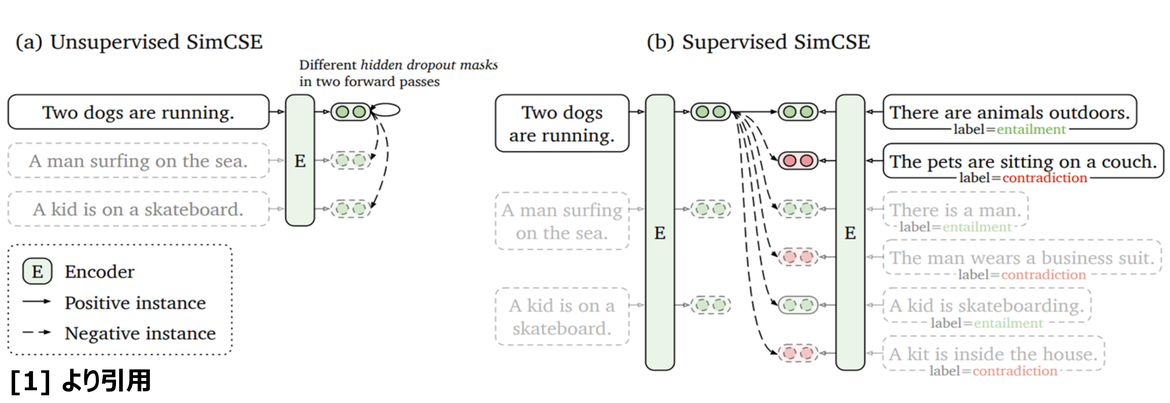
- 埋め込みを学習する際に，意味が近いものを近づけ，そうでないものは遠ざけることで効率的に学習する手法.

SimCSE [1]

- 対照学習を用いることで，BERT[2] などの事前学習モデルを unsupervised (unsup), supervised (sup）の2種類の設定で fine-tuning するモデル.
- [CLS] を文の分散表現とみなし，文類似度にはコサイン類似度を用いる.
- loss は温度パラメータ (τ) 付き softmax cross-entropy.

unsup の fine-tuing の設定

batch をモデルに 2 回入力すると，dropout により異なる分散表現を出力し，同じ文の組を positive，異なる文の組を negative とする.



[1] より引用

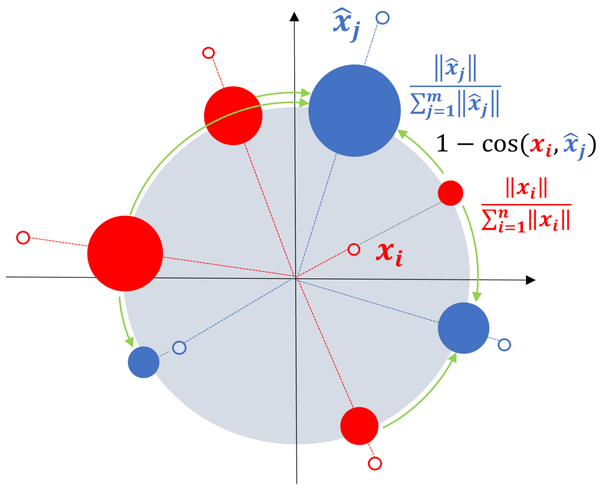
提案手法

背景

SimCSE の fine-tuning 時に [CLS] を用いたコサイン類似度ではなく，文中の単語の分散表現を用いて WRD, BERTScore, DynaMax のような高性能な文類似度を計算することで文類似度タスクの性能の向上が期待できる.
$w_i, \hat{w}_i$ を単語とし，2つの文を $s = (w_1, \cdots, w_n), \hat{s} = (\hat{w}_1, \cdots, \hat{w}_m)$ とし，BERT-based model の分散表現を用いて $x = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n \times d}, \hat{x} = (\hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_m)^\top \in \mathbb{R}^{m \times d}$ と表す．以下の手法の入力は単語ベクトル集合.

Word Rotator's Distance (WRD) [3]

- 最適輸送距離に基づく文類似度.
- ベクトルの長さとコサイン類似度を用いて分布，コストを定義.



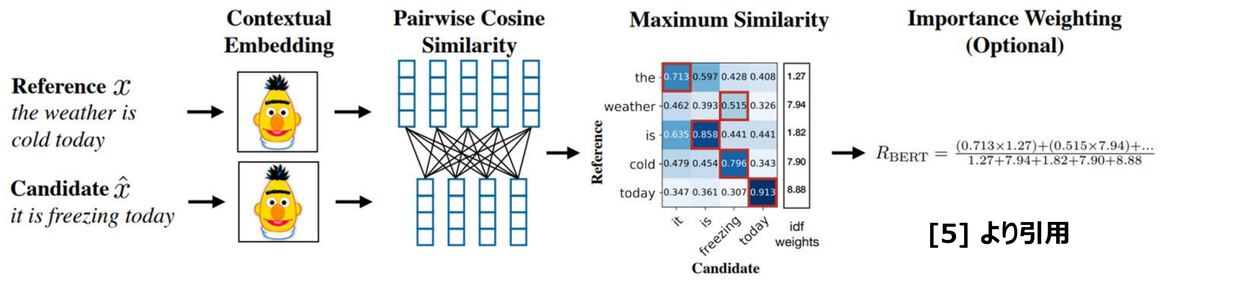
BERTScore [4]

- 生成文が参照文とどれだけ近いかを自動的に測る手法.
- 規格化された分散表現のコサイン類似度から以下で Recall, Precision, F値を計算.

$$R_{\mathrm{BERT}} = \frac{\sum_{i=1}^n a_i \max_{j \in \{1,\cdots,m\}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\sum_{i=1}^n a_i}$$

$$P_{\mathrm{BERT}} = \frac{\sum_{j=1}^m \hat{a}_j \max_{i \in \{1,\cdots,n\}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j}{\sum_{j=1}^m \hat{a}_j}$$

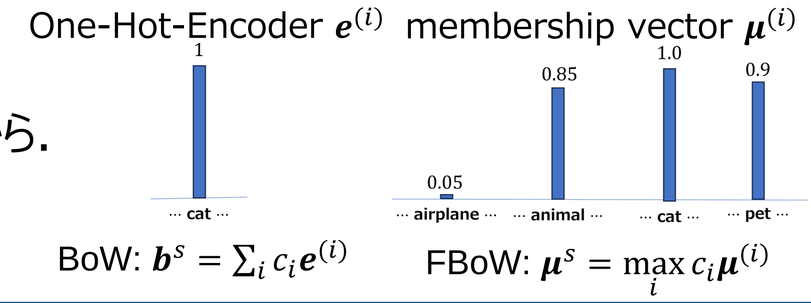
$$F_{\mathrm{BERT}} = 2\frac{P_{\mathrm{BERT}} \cdot R_{\mathrm{BERT}}}{P_{\mathrm{BERT}} + R_{\mathrm{BERT}}}$$



[5] より引用

DynaMax [5]

- BoW を連続化した Fussy BoW を定義.
- $\boldsymbol{\mu}^{(i)} \in \mathbb{R}^{(n+m)}$ は2文中のコサイン類似度から.
- 文類似度は $\mathrm{Fjaccard}(\boldsymbol{\mu}^s, \boldsymbol{\mu}^{\hat{s}}) = \sum_{k=1}^{n+m} \min(\mu_k^s, \mu_k^{\hat{s}}) / \max(\mu_k^s, \mu_k^{\hat{s}})$ で測る.



実験

実験設定

- unsup で bert-base-uncased で比較.
- [1] と同様 Wikipedia の 10^6 の文を使用.
- デフォルトで $\tau = 0.05$, BERTScore, Dynamax のみ $\tau = 0.025$.
- その他はデフォルトと同じハイパーパラメータ.
- STS12-16, STSB, SICK-R ついて，文類似度と人間のスコアの順位相関係数を計算.
- 学習時の eval は STSB と SICK-R の dev set の平均.
- WRD, BERTScore では idf を重みとした場合も試す.

Model	Similarity	Weight	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
$\mathrm{BERT}_{\mathrm{base}}$	CLS-cos	-	**68.40**	**82.41**	**74.38**	**80.91**	**78.56**	76.85	**72.23**	**76.25**
	WRD	norm	67.61	80.39	69.98	78.14	76.14	75.34	69.20	73.83
		idf	63.95	78.55	70.03	80.02	77.01	77.78	69.28	73.80
	BERTScore	uniform	60.85	77.95	69.32	79.13	76.28	76.58	65.61	72.25
		idf	62.76	79.43	70.79	80.64	77.05	**78.70**	68.83	74.03
	DynaMax	-	63.07	79.25	70.71	80.54	75.52	76.78	67.43	73.33

表1: それぞれのデータセットについて，1番目に良いスコアを太字，2番目に良いスコアを下線で示す.

結果

- コサイン類似度が1番目，BERTScore が2番目に良い.
- WRD の STS12, STS13 を除いて，idf は uniform よりも良い性能を発揮.

考察

- (コサイン類似度の学習のしやすさ)>(他の類似度の性能の良さ)?
- BERTScore が2番目に良いのはコサイン類似度を WRD, DynaMax より直接的に用いた手法であるから？

今後の展望

- 他の事前学習モデルでの実験.
- WRD, BERTScore で新たな重みを試す.
- 例えば重みも fine-tuing で学習など.

参考文献

[1] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In EMNLP.
[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In NAACL.
[3] Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. Word rotator's distance. In EMNLP.
[4] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In ICLR.
[5] Vitalii Zhelezniak, Aleksandar Savkov, April Shen, Francesco Moramarco, Jack Flann, and Nils Y Hammerla. 2019. Don't Settle for Average, Go for the Max: Fuzzy Sets and Max-Pooled Word Vectors. In ICLR.