	doc_1		doc_2		decision	id
cases	authors	• Ansel MacLaughlin • Jwala Dhamala • Anoop Kumar • Sriram Venkatapathy • Ragav Venkatesan • Rahul Gupta	authors	• Ansel MacLaughlin • Jwala Dhamala • Anoop Kumar • Sriram Venkatapathy • Ragav Venkatesan • Rahul Gupta	NOT DUPLICATES	420
	title	Evaluating the Effectiveness of Efficient Neural Architecture Search for Sentence-Pair Tasks	title	Evaluating the Effectiveness of Efficient Neural Architecture Search for Sentence-Pair Tasks		
	publication_date	2020-10-08 00:00:00	publication_date	2020-10-08 20:26:34+00:00		
	source	SupportedSources.OPENALEX	source	SupportedSources.ARXIV		
	journal	arXiv (Cornell University)	journal	None		
	volume		volume			
	doi	10.48550/arxiv.2010.04249	doi			
	urls	• https://openalex.org/W4287644805 • https://doi.org/10.48550/arxiv.2010.04249 • http://arxiv.org/pdf/2010.04249	urls	• http://arxiv.org/pdf/2010.04249v1 • http://arxiv.org/abs/2010.04249v1 • http://arxiv.org/pdf/2010.04249v1		
	id	id1261178150777643172	id	id-4910804529046475945		
	abstract		abstract	Neural Architecture Search (NAS) methods, which automatically learn entire neural model or individual neural cell architectures, have recently achieved competitive or state-of-the-art (SOTA) performance on variety of natural language processing and computer vision tasks, including language modeling, natural language inference, and image classification. In this work, we explore the applicability of a SOTA NAS algorithm, Efficient Neural Architecture Search (ENAS) (Pham et al., 2018) to two sentence pair tasks, paraphrase detection and semantic textual similarity. We use ENAS to perform a micro-level search and learn a task-optimized RNN cell architecture as a drop-in replacement for an LSTM. We explore the effectiveness of ENAS through experiments on three datasets (MRPC, SICK, STS-B), with two different models (ESIM, BiLSTM-Max), and two sets of embeddings (Glove, BERT). In contrast to prior work applying ENAS to NLP tasks, our results are mixed -- we find that ENAS architectures sometimes, but not always, outperform LSTMs and perform similarly to random architecture search.		
	versions		versions			