cases	doc_1		doc_2		decision	id
			authors	• Ansel MacLaughlin • J. Dhamala • Anoop Kumar • Sriram Venkatapathy • Ragav Venkatesan • Rahul Gupta	DUPLICATES	275
	authors	• Ansel MacLaughlin • Jwala Dhamala • Anoop Kumar • Sriram Venkatapathy • Ragav Venkatesan • Rahul Gupta	**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 00:00:00	**source**	SupportedSources.SEMANTIC_SCHOLAR		
	source	SupportedSources.OPENALEX	**journal**	ArXiv		
	journal	arXiv (Cornell University)	**volume**	abs/2010.04249		
	volume		**doi**	10.18653/v1/2020.insights-1.4		
	doi	None	**urls**	• https://www.semanticscholar.org/paper/421d52a9aa9f52c17f9f4af090e5a0e4a79d3f4f		
	urls	• https://openalex.org/W3092155366	**id**	id9119261069938251721		
	id	id-7163506145219965697	**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 â€“ we find that ENAS architectures sometimes, but not always, outperform LSTMs and perform similarly to random architecture search.		
	abstract		**versions**			
	versions					