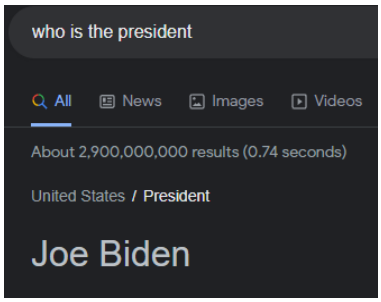
ACL Paper Summary

Paper Title + Authors

The paper is titled **“Learning Dense Representations of Phrases at Scale”**
Its authors are **Jinhyuk Lee1**, **Mujeen Sung**, **Jaewoo Kang**, and **Danqi Chen**
Affiliations: **Korea University**, **Princeton University**

Problem Addressed by the Paper



This paper proposes a solution for **open domain question answering (OpenQA)**. In OpenQA, the user enters a question in natural language. Based on knowledge gained from a very large and unstructured text corpus spanning potentially several topics, the system generates a response in natural language. OpenQA systems are evaluated on their ability to provide quick, accurate, and appropriate responses to the user’s questions. The specific OpenQA solution in this paper is intended to provide an instant answer to a given question by highlighting the most appropriate response phrase in the corpus. This particular solution emphasizes speed and scale rather than raw accuracy.

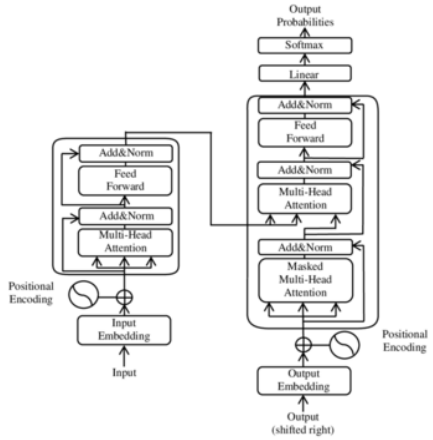
Prior Work

This paper builds upon and distinguishes itself from prior work, including:

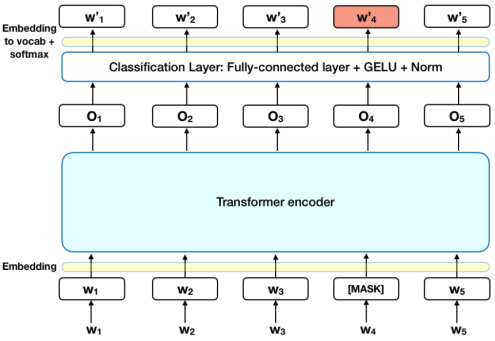
- **Reader-retriever models**
  This approach to OpenQA is two-stage: first, find the most relevant documents. Then, process those top documents to extract the best response to the question. The second phase is quite slow because the vector representations of each phrase of the retrieved documents depends on the user’s query. Building and evaluating these vectors at inference time is slow, so this approach is not ideal. This paper intends to achieve similar accuracy to reader-retriever models in a fraction of the time.

- **Phrase retrieval models**
  These encode phrases from documents independently from the question, which allows all the phrases from the corpus to be encoded before inference time. This is more performant than reader-retriever models because an efficient search algorithm, **Maximum Inner-Product Search (MIPS)**, can be used to quickly find which phrase vector best corresponds to the question vector. In prior work, **sparse vectors** were used to represent phrases. Examples of sparse vectors are word buckets or tf-idf vectors. A drawback of using sparse vectors is that they do not lend themselves to efficient search. This paper improves on prior work by encoding phrases using purely **dense vectors**, which can encode richer semantics using less data.

- **Transformer Architecture**
  The transformer architecture was a landmark achievement in ML & NLP. It has several mechanisms that distinguish it from earlier models such as RNNs and LTSMs, including **self-attention** and **positional encoding**. These mechanisms improve its ability to embed semantics of words into word vectors, which ultimately lets it perform well on various NLP tasks such as translation.



- **BERT (Bidirectional Encoder Representations from Transformers)**
  BERT takes the encoder part from the transformer architecture and trains it on two unsupervised learning tasks simultaneously: predicting masked words in a passage and determining the correct order of randomly swapped sentences. BERT achieves remarkably well on language modeling tasks, and it is often used to encode the meaning of words in passages.



- **SpanBERT**
  BERT maps each word in a sentence to a vector representation. This is not ideal for phrase extraction tasks, since we need to identify *phrases* in a passage, not individual *words*. SpanBERT is similar to BERT, but it maps each word to both a *start* and *end* vector and is trained by trying to "guess" the phrase that appears between a start and end vector when the actual phrase is masked. Like BERT, this training is unsupervised. The result of this training is a model that does a great job of encoding the phrases in a passage.

Architectural Overview

The model described in this paper is called **DensePhrases**. The general flow is like this:

- **Phrases** are sequences of words in a passage not exceeding length L=20.
- A SpanBERT-like model called the **phrase encoder** passes through the knowledge base corpus and generates a *start* and *end* vector for each word. A phrase is identified by a pair of (start, end) vectors not separated by more than 20 words.
- When the user enters a question, the question gets encoded into a (q-start, q-end) pair of vectors by the **question encoder**. During the inference process,

the phrase from the corpus with representation (p-start, p-end) that has a maximum inner product with (q-start, q-end) is selected as the correct answer.

An objective of this model is for a vector representation of a question to closely resemble the vector representation of an answer. Then, MIPS can be used to find which phrase in the text is the best answer for a question.

Unique Contributions of this Paper

The major contributions of this paper relate to how the question encoder and phrase encoder are trained. Better training objectives means smarter encoders, and smarter encoders means smarter dense representations. Smarter representations means better accuracy at QA tasks.

The paper outlines the following techniques used to train the encoders:

- **Single-Passage Training**
  The model is trained on data sets of question/answer pairs created by humans. For every question in a training batch, both the question and the passage in which the correct answer is found is given to the model. The loss function is updated in proportion to the inverse likelihood that the model predicts the correct answer from the given passage. (Not the entire corpus)

- **Data Augmentation**
  A common issue for models trained on Q/A sets is that there is often an imbalance where a single question could have multiple correct answers in the passage, but a given answer only has one instance of a question in the training set. The authors of this paper attempt to counteract this issue by using another model to generate additional questions for the training set. This is an example of ***data augmentation***, a technique to reduce variance by generating slightly modified copies of existing data.

- **Distillation**
  The model parameters of the phrase encoder were fine-tuned to resemble those of established pre-trained SpanBERT models. Particularly, the *KL-Divergence* between the outputs of the pre-trained and question encoder models on a given training phrase contributed to the loss function value.

- **In-Batch & Pre-Batch Negatives**
  Above we explained how the model is training *on a single passage*. But the model needs to know how to find the right answer in an *entire corpus*, not just a single passage. The authors describe in-batch and pre-batch negatives, which are used to train the model to discriminate against incorrect answers appearing in irrelevant passages.

The overall loss function is a linear combination of the single-passage loss, the distillation loss, and the in/pre-batch negative loss.

After the model is trained on the loss function described above, the authors propose a technique called **query-side fine tuning** to make the question encoder even smarter. It works like this: For every question in the training set, the model predicts the top answers from the corpus. The new loss function objective is to maximize the likelihood that the model picks the *truly* correct answer phrase instead of all the other phrases from the corpus that the model thought could be correct.

How the Authors Evaluated Their Work

Before evaluating, the authors trained the models on the following previously established data sets:

- SQuAD
- Natural Questions (NQ) [training]

Then they evaluated the performance of the model on the following sets:

- Natural Questions (NQ) [testing]
- WebQuestions (WQ)
- CuratedTREC (TREC)
- TriviaQA (TQA)
- SQuAD

Using a metric called exact match (EM), the authors evaluated their model on 3 different types of tasks: reading comprehension, open domain QA, and slot filling. The table below summarizes the results for Open Domain QA:

Model		NQ	WQ	TREC	TQA	SQuAD
Retriever-reader	$\mathcal{C}_{\text{retr}}$: (Pre-)Training					
DrQA (Chen et al., 2017)	-	-	20.7	25.4	-	29.8
BERT + BM25 (Lee et al., 2019)	-	26.5	17.7	21.3	47.1	**33.2**
ORQA (Lee et al., 2019)	{Wiki.}†	33.3	36.4	30.1	45.0	20.2
$\text{REALM}_{\text{News}}$ (Guu et al., 2020)	{Wiki., CC-News}†	40.4	40.7	42.9	-	-
DPR-multi (Karpukhin et al., 2020)	{NQ, WQ, TREC, TQA}	**41.5**	**42.4**	**49.4**	**56.8**	24.1
Phrase retrieval	$\mathcal{C}_{\text{phrase}}$: Training					
DenSPI (Seo et al., 2019)	{SQuAD}	8.1*	11.1*	31.6*	30.7*	36.2
DenSPI + Sparc (Lee et al., 2020)	{SQuAD}	14.5*	17.3*	35.7*	34.4*	**40.7**
DenSPI + Sparc (Lee et al., 2020)	{NQ, SQuAD}	16.5	-	-	-	-
DensePhrases (ours)	{SQuAD}	31.2	36.3	50.3	**53.6**	39.4
DensePhrases (ours)	{NQ, SQuAD}	**40.9**	**37.5**	**51.0**	50.7	38.0

DensePhrases doesn't do quite as well as the best retriever-reader model in terms of accuracy, but it is far more efficient. And for a query-agnostic model, its accuracy is quite impressive, since it far surpasses the other phrase retrieval models and almost achieves the same accuracy as the best Reader-retriever.

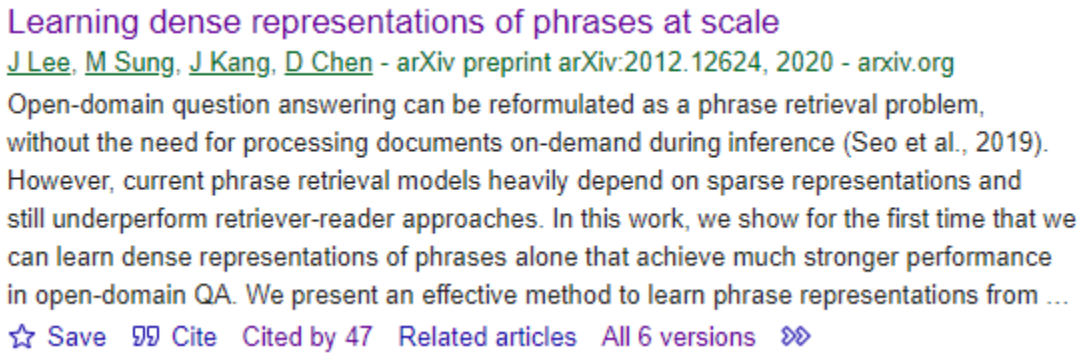
Why Is This Paper Important?

In our opinion, this paper is important primarily for two reasons:

Firstly, the model it explores is useful by itself as a powerful tool for question answering. The techniques it describes could very likely be used for search engines or other information services that need to provide instant answers to users, even when the knowledge base is enormous.

Another reason why this paper is important is that it lays a theoretical groundwork for future developments in natural language processing. Researchers could gain tremendous insight from knowing the strengths and limitations of using dense representations of phrases for phrase retrieval & OpenQA problems.

Number of Citations the Authors Received on Google Scholar



The authors have received 47 citations for this paper

Times authors have been cited for other papers:

1. Jinhyuk Lee has been cited 3673 times
2. Mujeen Sung has been cited 303 times
3. Jaewoo Kang has been cited 10154 times
4. Danqi Chen has been cited 24592 times

Complete Citation

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning Dense Representations of Phrases at Scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, Online. Association for Computational Linguistics.