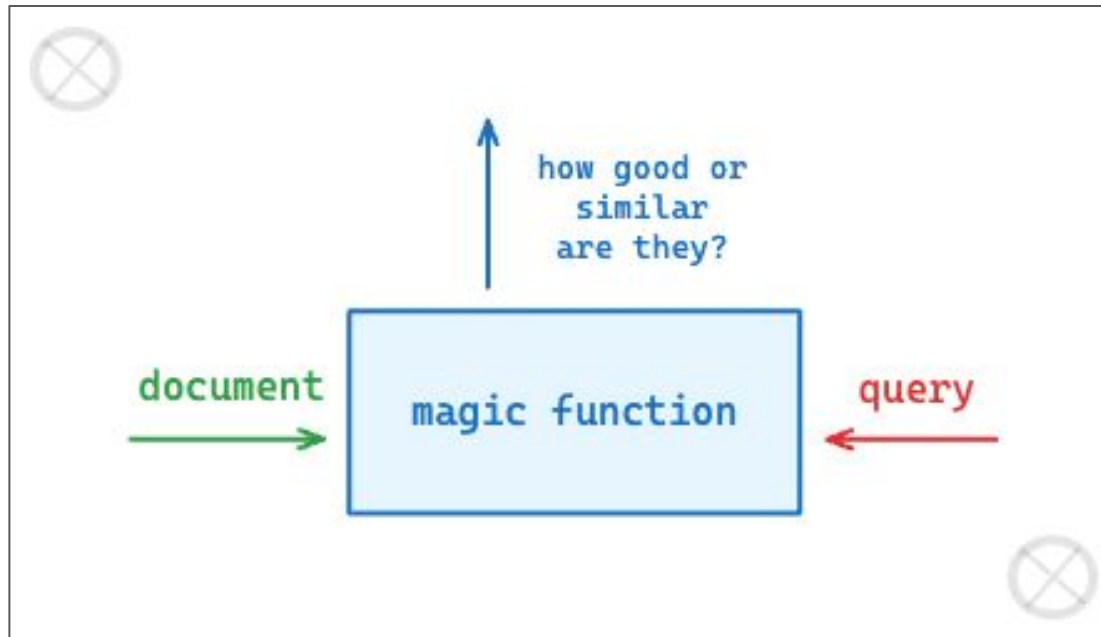
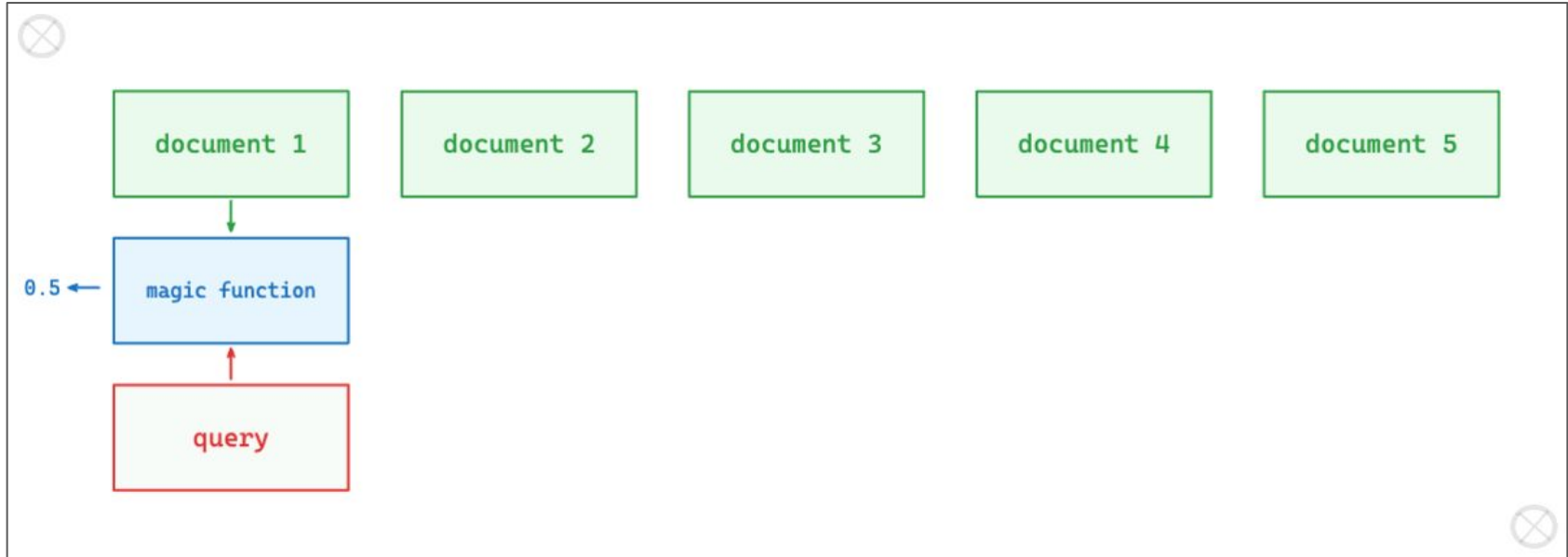


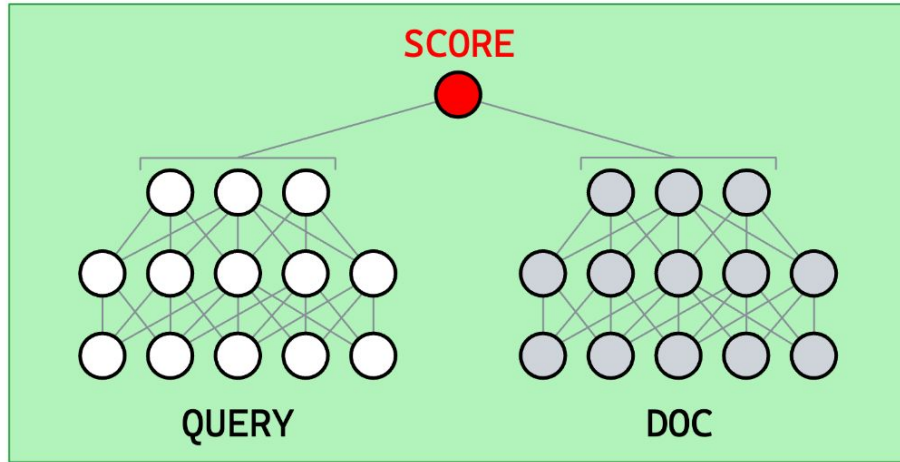
Learn To Search
Learn To Recommend
Learn To Rank
Learn To Compare

Learn To Rank

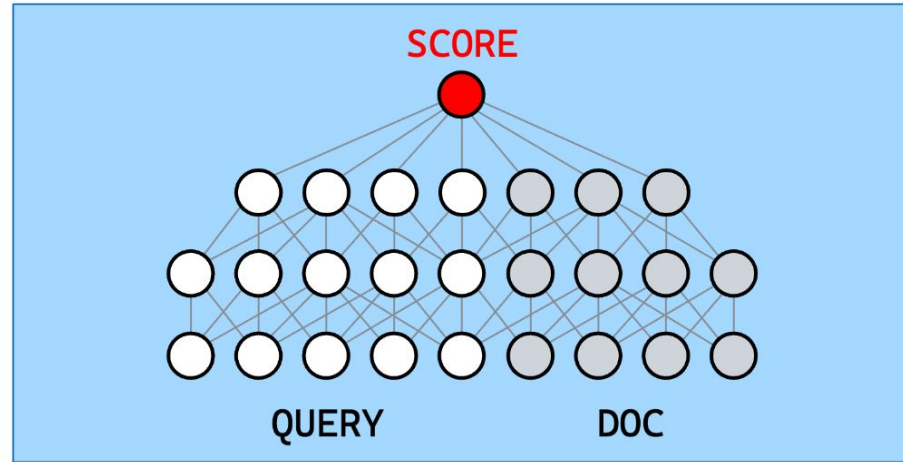


Learn To Rank



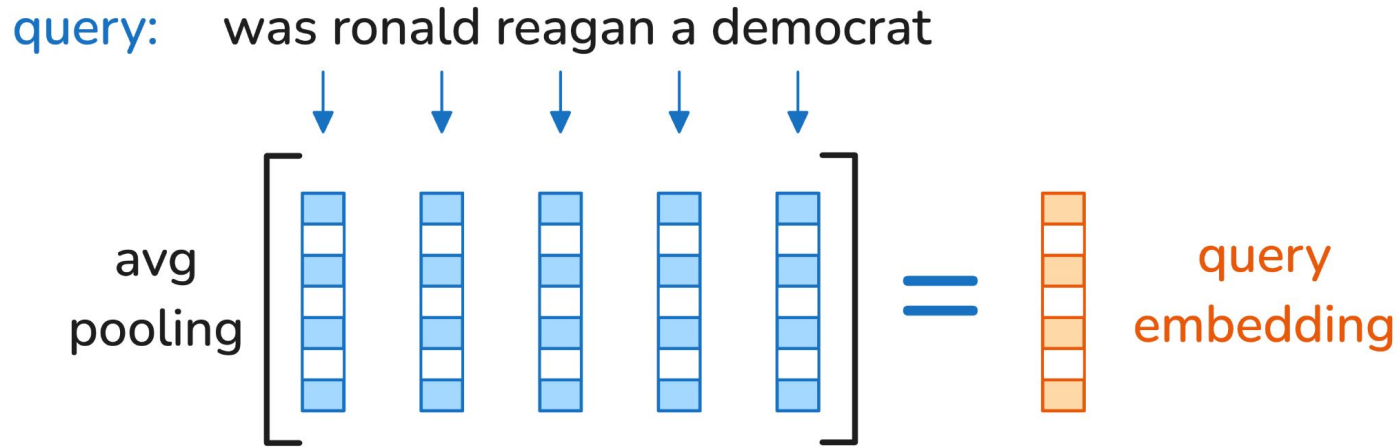


Dual-Encoder



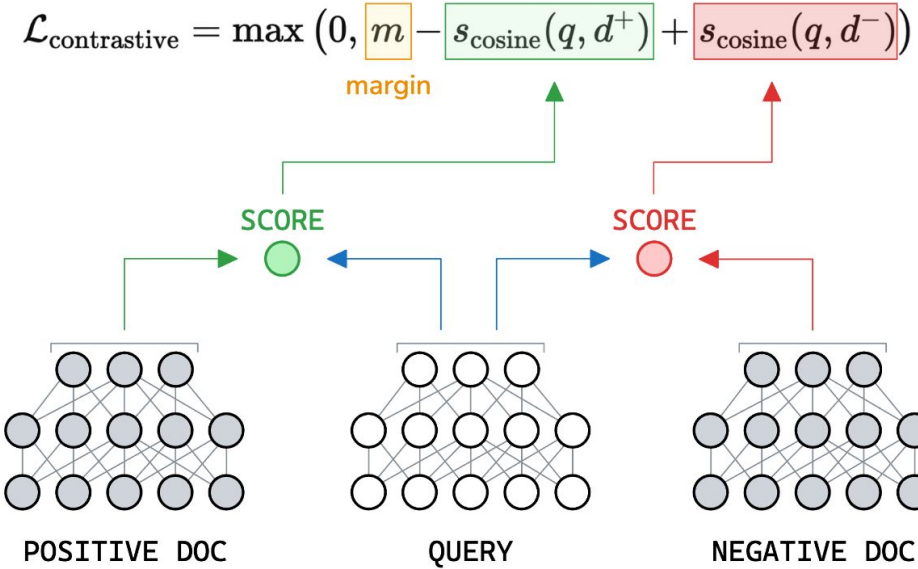
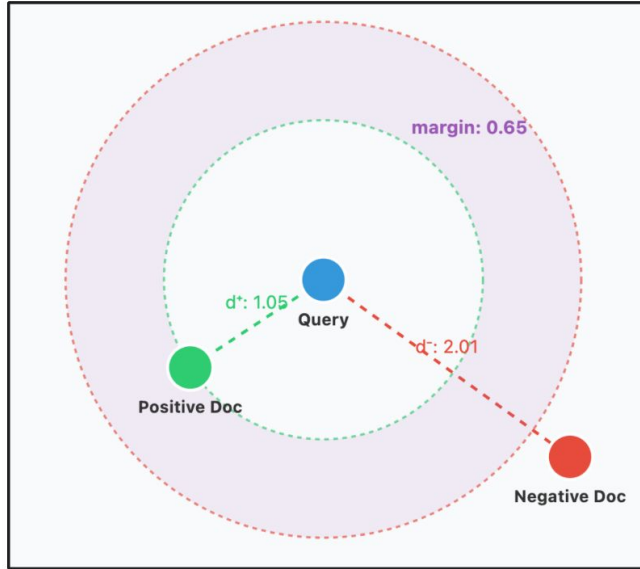
Cross-Encoder

Start Simple



Do the same for the document, avg pool both document and query

Triplet Loss



<https://claude.site/artifacts/16d7e462-bfc4-4229-ae32-0b5a45a1c5a4>

Tiny Example

```

1
2 import torch
3
4
5 class QryTower(torch.nn.Module):
6     def __init__(self):
7         super().__init__()
8         self.fc = torch.nn.Linear(10, 1)
9
10    def forward(self, x):
11        x = self.fc(x)
12        return x
13
14
15 class DocTower(torch.nn.Module):
16     def __init__(self):
17         super().__init__()
18         self.fc = torch.nn.Linear(10, 1)
19
20    def forward(self, x):
21        x = self.fc(x)
22        return x
23

```

```

24
25 qryTower = QryTower()
26 docTower = DocTower()
27
28
29 qry = torch.randn(1, 10) # 1 query, 10-dim embedding
30 pos = torch.randn(1, 10) # 1 positive doc, 10-dim embedding
31 neg = torch.randn(1, 10) # 1 negative doc, 10-dim embedding
32
33
34 qry = qryTower(qry)
35 pos = docTower(pos)
36 neg = docTower(neg)
37
38
39 dst_pos = torch.nn.functional.cosine_similarity(qry, pos)
40 dst_neg = torch.nn.functional.cosine_similarity(qry, neg)
41 dst_dif = dst_pos - dst_neg
42 dst_mrg = torch.tensor(0.2)
43
44
45 loss = torch.max(torch.tensor(0.0), dst_mrg - dst_dif)
46 loss.backward()
47

```

Dataset Viewer Auto-converted to Parquet API Embed Data Studio

Subset (2)
v1.1 · 102k rows ← Start with v1.1

Split (3)
train · 82.3k rows

Search this dataset

answers sequence · lengths	passages sequence	query string · lengths	query_id int32	query_type string · classes
["Results-Based Accountability is...	{ "is_selected": [0, 0, 0, 0, 0, 1, 0, 0, 0, 0], "passage_text": ["Since 2007, the RBA's...	what is rba	19,699	description
["Yes"]	{ "is_selected": [0, 1, 0, 0, 0, 0, 0, 0], "passage_text": ["In his younger years, Ronald...	was ronald reagan a democrat	19,700	description
["20-25 minutes"]	{ "is_selected": [0, 0, 0, 0, 1, 0, 0, 0, 0, 0], "passage_text": ["Sydney, New South Wales,...	how long do you need for sydney...	19,701	numeric
["\$11 to \$22 per square foot"]	{ "is_selected": [0, 0, 0, 0, 0, 0, 0, 0, 1], "passage_text": ["In regards to tile installation...	price to install tile in shower	19,702	numeric
["Due to symptoms in the body"]	{ "is_selected": [0, 0, 1, 0, 0, 0, 0, 0, 0], "passage_text": ["Conclusions: In adult body CT,...	why conversion observed in body	19,703	description
["Inside the rib cage."]	{ "is_selected": [0, 0, 0, 0, 1, 0, 0, 0, 0, 0], "passage_text": ["Get the Latest health and...	where are the lungs located in...	19,704	location

< Previous 1 2 3 ... 824 Next >

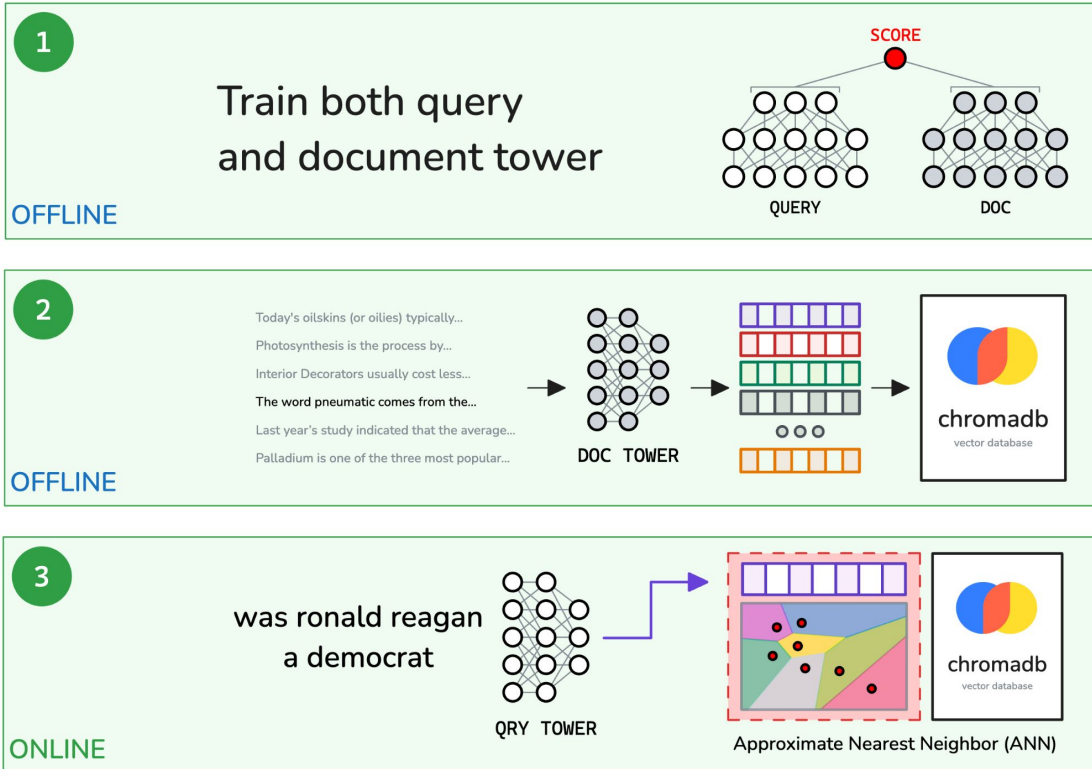
```
{
  "is_selected": [ 0, 1, 0, 0, 0, 0, 0, 0 ],
  "passage_text": ["A", "B", "C", "D", "E", "F", "G"],
  "url": [ ... ]
}
```

positive

positive

maybe hard negative

Stages



Signature Verification using a “Siamese” Time Delay Neural Network

Jane Bromley, Isabelle Guyon, Yann LeCun,
Eduard Säckinger and Roopak Shah
AT&T Bell Laboratories
Holmdel, NJ 07733
jbromley@big.att.com

Copyright©, 1994, American Telephone and Telegraph Company used by permission.

Abstract

This paper describes an algorithm for verification of signatures written on a pen-input tablet. The algorithm is based on a novel, artificial neural network, called a “Siamese” neural network. This network consists of two identical sub-networks joined at their outputs. During training the two sub-networks extract features from two signatures, while the joining neuron measures the distance between the two feature vectors. Verification consists of comparing an extracted feature vector with a stored feature vector for the signer. Signatures closer to this stored representation than a chosen threshold are accepted, all other signatures are rejected as forgeries.

1 INTRODUCTION

The aim of the project was to make a signature verification system based on the NCR 5990 Signature Capture Device (a pen-input tablet) and to use 80 bytes or less for signature feature storage in order that the features can be stored on the magnetic strip of a credit-card.

Verification using a digitizer such as the 5990, which generates spatial coordinates as a function of time, is known as dynamic verification. Much research has been carried out on signature verification. Function-based methods, which fit a function to the pen trajectory, have been found to lead to higher performance while parameter-based methods, which extract some number of parameters from a signa-

737

Dense Passage Retrieval for Open-Domain Question Answering

Vladimir Karpukhin¹, Barlas Öğuz¹, Sewon Min¹, Patrick Lewis,
Liedel Wu, Sergey Edunov, Danqi Chen¹, Wen-tau Yih
Facebook AI ¹University of Washington ²Princeton University
{vladk, barlaso, p.lewis, liedel, edunov, scotttyh}@fb.com
sewon@cs.washington.edu danqi@cs.princeton.edu

Abstract

Open-domain question answering relies on efficient passage retrieval to select candidate contexts, where traditional sparse vector space models, such as TF-IDF or BM25, are the de facto method. In this work, we show that retrieval can be practically implemented using dense representations alone, where embeddings are learned from a small number of questions and passages by a simple dual-encoder framework. When evaluated on a wide range of open-domain QA datasets, our dense retriever outperforms a strong Lucene-BM25 system greatly by 9%-19% absolute in terms of top-20 passage retrieval accuracy, and helps our end-to-end QA system establish new state-of-the-art on multiple open-domain QA benchmarks.¹

1 Introduction

Open-domain question answering (QA) (Voorhees, 1999) is a task that answers factoid questions using a large collection of documents. While early QA systems are often complicated and consist of multiple components (Ferruci (2012), Moldovan et al. (2003), *Inter-ale*), the advances of reading comprehension models suggest a much simplified two-stage framework: (1) a context retriever first selects a small subset of passages where some of them contain the answer to the question, and then (2) a machine reader can thoroughly examine the retrieved contexts and identify the correct answer (Chen et al., 2017). Although reducing open-domain QA to machine reading is a very reasonable strategy, a huge performance degradation is often observed in practice², indicating the needs of improving retrieval.

¹Equal contribution
²The code and trained models have been released at <https://github.com/facebookresearch/DPR>.
For instance, the exact match score on SQuAD v1.1 drops from above 80% to less than 40% (Yang et al., 2019a).

Retrieval in open-domain QA is usually implemented using TF-IDF or BM25 (Robertson and Zaragoza, 2009), which matches keywords efficiently with an inverted index and can be seen as representing the question and context in high-dimensional, sparse vectors (with weighting). Conversely, the dense, latent semantic encoding is complementary to sparse representations by design. For example, synonyms or paraphrases that consist of completely different tokens may still be mapped to vectors close to each other. Consider the question “Who is the bad guy in lord of the rings?”, which can be answered from the context “Sala Baker is best known for portraying the villain Sauron in the Lord of the Rings trilogy.” A term-based system would have difficulty retrieving such a context, while a dense retrieval system would be able to better match “bad guy” with “villain” and fetch the correct context. Dense encodings are also learnable by adjusting the embedding functions, which provides additional flexibility to have a task-specific representation. With special in-memory data structures and indexing schemes, retrieval can be done efficiently using maximum inner product search (MIPS) algorithms (e.g., Shrivastava and Li (2014); Guo et al. (2016)).

However, it is generally believed that learning a good dense vector representation needs a large number of labeled pairs of question and contexts. Dense retrieval methods have thus never been shown to outperform TF-IDF/BM25 for open-domain QA before ORQA (Lee et al., 2019), which proposes a sophisticated inverse cloze task (ICT) objective, predicting the blocks that contain the masked sentence, for additional pretraining. The question encoder and the reader model are then fine-tuned using pairs of questions and answers jointly. Although ORQA successfully demonstrates that dense retrieval can outperform BM25, setting new state-of-the-art results on multiple open-domain

arXiv:2004.04906v3 [cs.CL] 30 Sep 2020



Document Search: RNNs and the Two Towers



3 Lessons | 40 hours
with **Ardavan Afshar**



Good luck!



Teams

Recurrent Rebels

Coline
Nnamdi
Evelyn
Jason

Perceptron Party

Pry
Dimitris
Yurii
Dimitar

Activation Aces

Aygun
Alexander
Milo
Gaurav

Gradient Gigglers

Andrea
David
Stanley
Kori

Backprop Bunch

Maxime
Kenton
Ollie
Filippo

Batchnorm Bandits

Amy
Jack
Liam
Josh
Daniel

Overfitting Overlords

Neville
Artemis
Guillaume
James

Dropout Disco

Emil
Ardrit
Loredana
Maxi