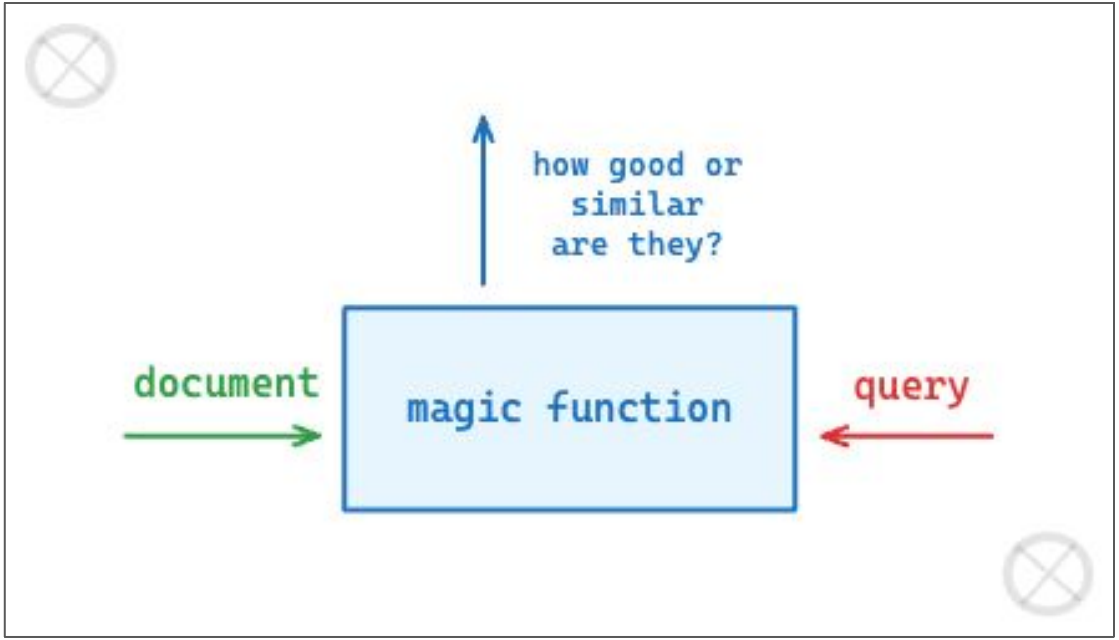
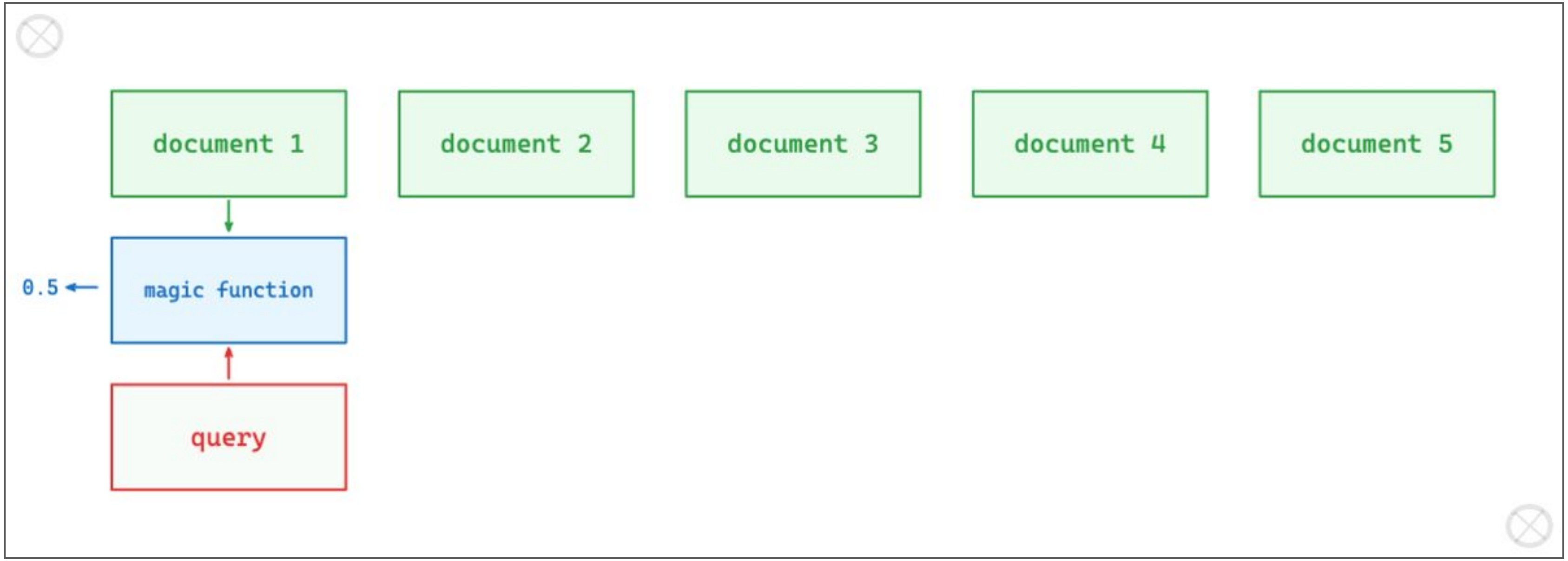
Week 2

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

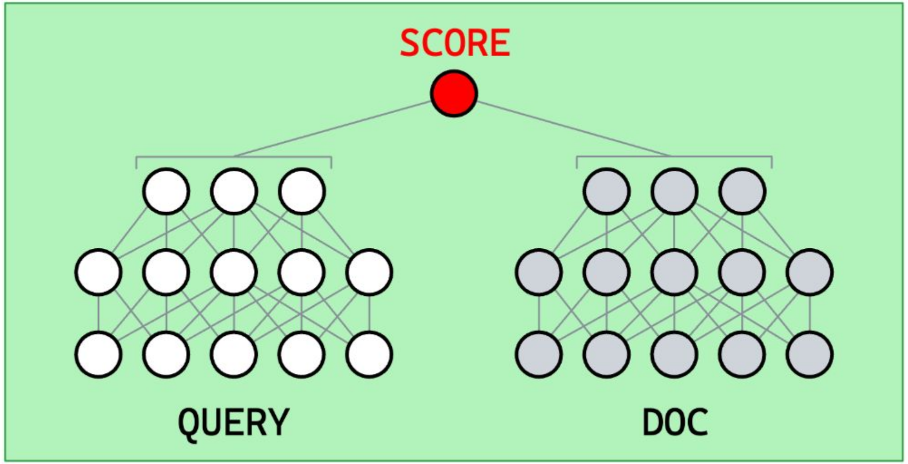
Learn To Rank



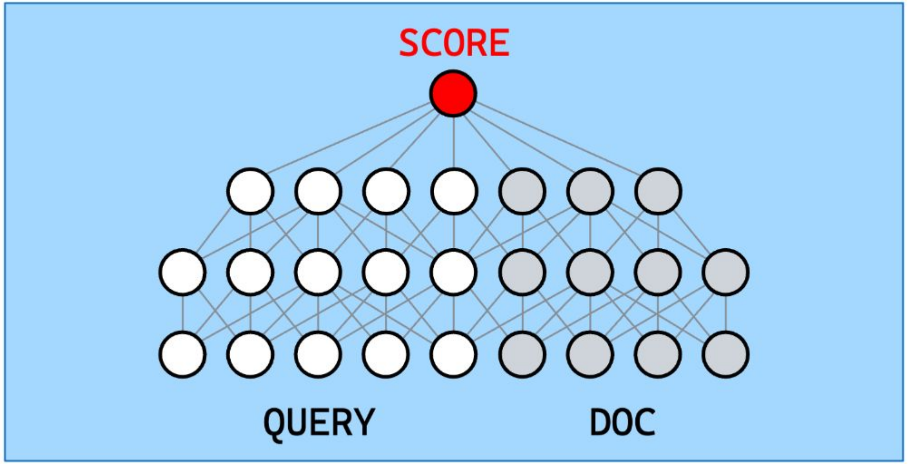
Learn To Rank



Architectures

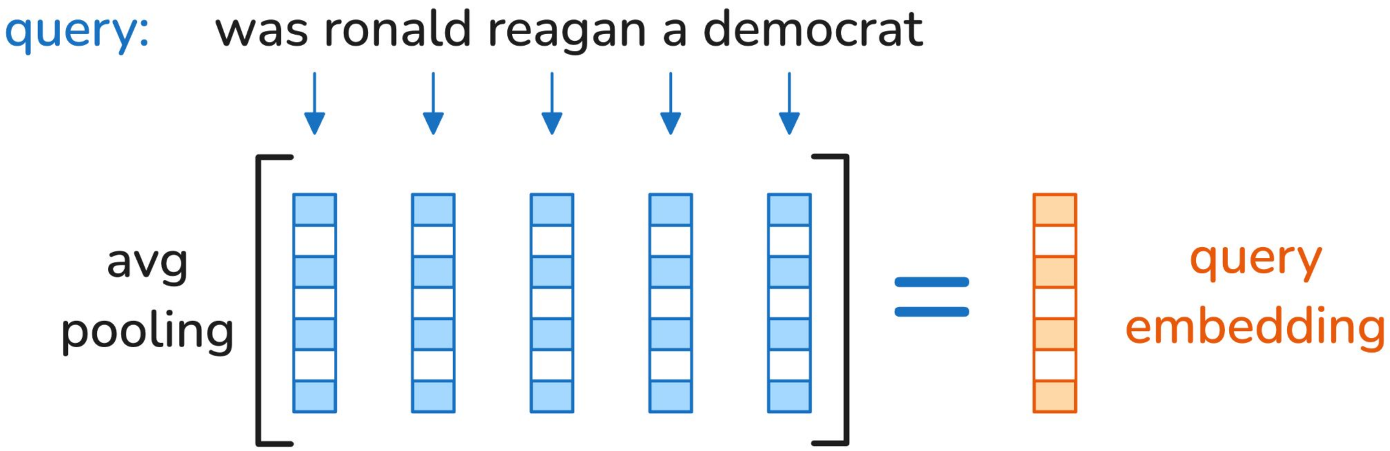


Dual-Encoder



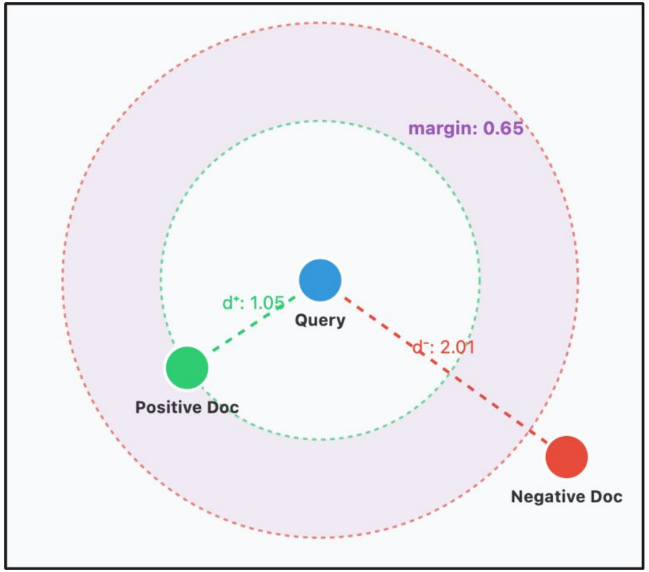
Cross-Encoder

Start Simple

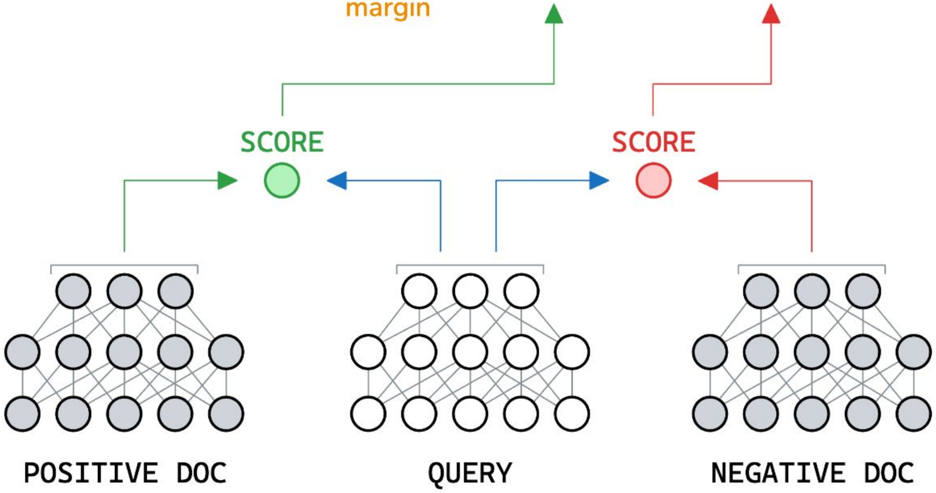


Do the same for the document, avg pool both document and query

Triplet Loss



$$\mathcal{L}_{\text{contrastive}} = \max\left(0, \underbrace{\boxed{m}}_{\text{margin}} - \boxed{s_{\text{cosine}}(q, d^{+})} + \boxed{s_{\text{cosine}}(q, d^{-})}\right)$$



https://claude.ai/public/artifacts/16d7e462-bfc4-4229-ae32-0b5a45a1c5a4

Tiny Example

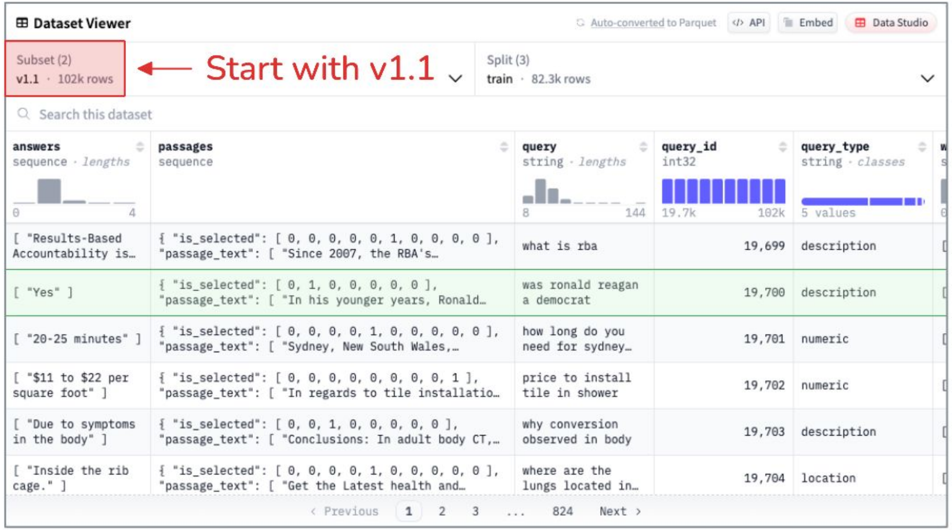
FOUNDERS AND CODERS

Machine Learning Institute

```python

import torch


class QryTower(torch.nn.Module):
  def __init__(self):
    super().__init__()
    self.fc = torch.nn.Linear(10, 1)

  def forward(self, x):
    x  = self.fc(x)
    return x


class DocTower(torch.nn.Module):
  def __init__(self):
    super().__init__()
    self.fc = torch.nn.Linear(10, 1)

  def forward(self, x):
    x  = self.fc(x)
    return x

```

```python

qryTower = QryTower()
docTower = DocTower()


qry = torch.randn(1, 10) # 1 query, 10-dim embedding
pos = torch.randn(1, 10) # 1 positive doc, 10-dim embedding
neg = torch.randn(1, 10) # 1 negative doc, 10-dim embedding


qry = qryTower(qry)
pos = docTower(pos)
neg = docTower(neg)


dst_pos = torch.nn.functional.cosine_similarity(qry, pos)
dst_neg = torch.nn.functional.cosine_similarity(qry, neg)
dst_dif = dst_pos - dst_neg
dst_mrg = torch.tensor(0.2)


loss = torch.max(torch.tensor(0.0), dst_mrg - dst_dif)
loss.backward()

```

MS MARCO



answers	passages	query	query_id	query_type
sequence · lengths	sequence	string · lengths	int32	string · classes
0 – 4		8 – 144	19.7k – 102k	5 values
["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], "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 installatio…	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], "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 ›

positive

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

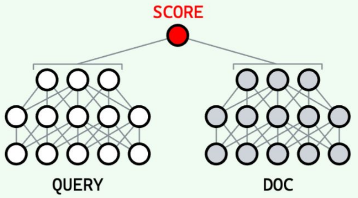
positive

maybe hard negative

Stages

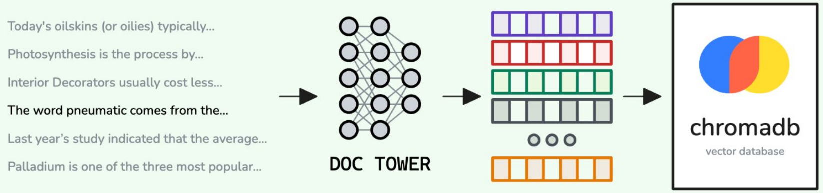
1

Train both query and document tower



OFFLINE

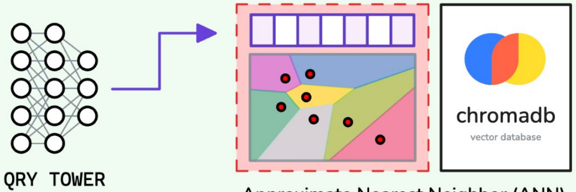
2



OFFLINE

3

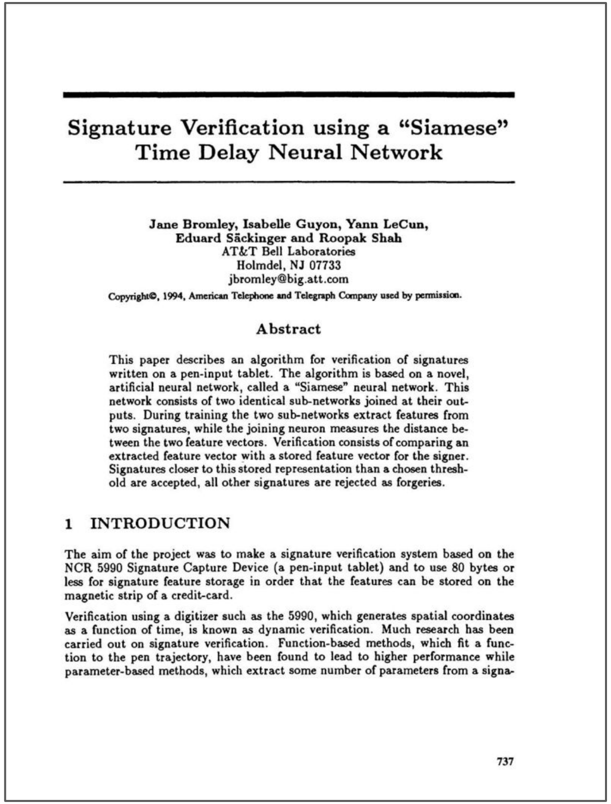
was ronald reagan a democrat



Approximate Nearest Neighbor (ANN)

ONLINE

Papers



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

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-

Dense Passage Retrieval for Open-Domain Question Answering

Vladimir Karpukhin*, Barlas Oğuz*, Sewon Min[†], Patrick Lewis,
Ledell Wu, Sergey Edunov, Danqi Chen[‡], Wen-tau Yih
Facebook AI [†]University of Washington [‡]Princeton University
{vladk, barlaso, plewis, ledell, edunov, scottyih}@fb.com
sewon@cs.washington.edu
danqic@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 (Ferrucci (2012); Moldovan et al. (2003), *inter alia*), 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 practice[2], indicating the needs of improving retrieval.

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 be 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

*Equal contribution

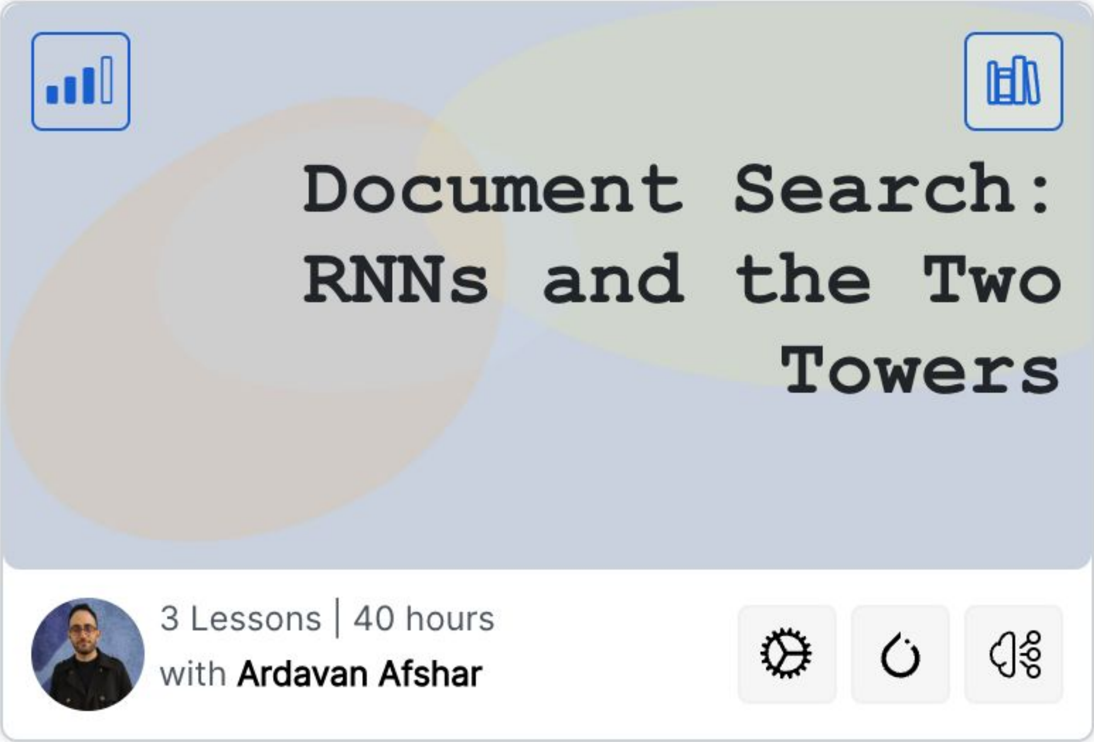
[1]The code and trained models have been released at https://github.com/facebookresearch/DPR.

[2]For instance, the exact match score on SQuAD v1.1 drops from above 80% to less than 40% (Yang et al., 2019a).

FOUNDERS AND CODERS

Cortex

Machine Learning Institute



FOUNDERS AND CODERS

Good luck!

Machine Learning Institute

Recurrent Rebels

Marcin Tolysz
Jacob Jenner
Ayman Abbas
Rasched Haidari

Gradient Gigglers

Kadriye Turkcan
Andrew
Nikolas Kuhn
Helen Zhou

Overfitting Overlords

Jingyan Chen
Ethan Edwards
Yali Pan
Dan Goss

Hyperparameter Hippies

David Edev
Tao Zamorano
Prima Gouse
Miguel Parracho

Perceptron Party

Charles Cai
Maria Sharif
Arjuna James
Ben Liong

Backprop Bunch

Aparna Pillai
Peter O'Keeffe
Ben Bethell
James Yan

Dropout Disco

Anton Dergunov
Andrei Zhirnov
Esperanza Shi
Hikaru Tsujimura

Kernel Kittens

Joao Esteves
Tomas Krajcoviech
Clement Ha
Ben Williams

Bayesian Buccaneers

Tyrone Nicholas
Halil Serkan Uz
Umut Sagir
Adam Beedell

Feature Fiestas

Rosh Beed
Ewan Beattie
James Carter
Melanie Wong