The Curse of Dense Low-Dimensional Information Retrieval for Large Index Sizes

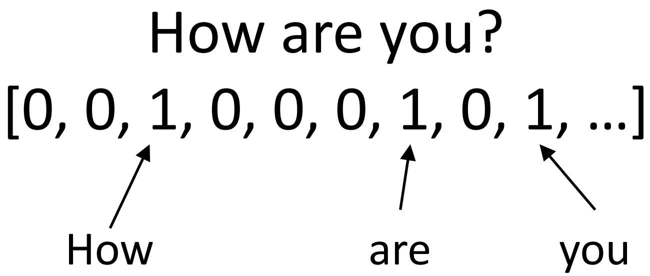
Nils Reimers
Iryna Gurevych

ACL 2021

Dense vs Sparse Representations

Sparse Lexical Representations

- Each word has it own dimension
- Most dimensions are zero



How are you?

[0, 0, 1, 0, 0, 0, 1, 0, 1, …]

Dense Representations

- $f(Text) \rightarrow \mathbb{R}^k$
- k: 100 – 1000
- All dimensions non-zero

- How do dense representations compare to sparse representations for **large index sizes**?

Example

Model	**10k**	**100k**	**1M**	**8.8M**	**100M**
BM25	79.9	63.9	40.1	17.6	?
Dense Model	89.0	71.1	42.2	17.3	?
Difference	9.1	7.2	2.1	-0.3	?

- MS MARCO Passage Retrieval dataset
- MRR@10
- Simple training script for the dense model

Theorem

- The probability for false positives :
    1) increases with the index size n
    2) Increases with fewer dimensions k

- Proof in the paper

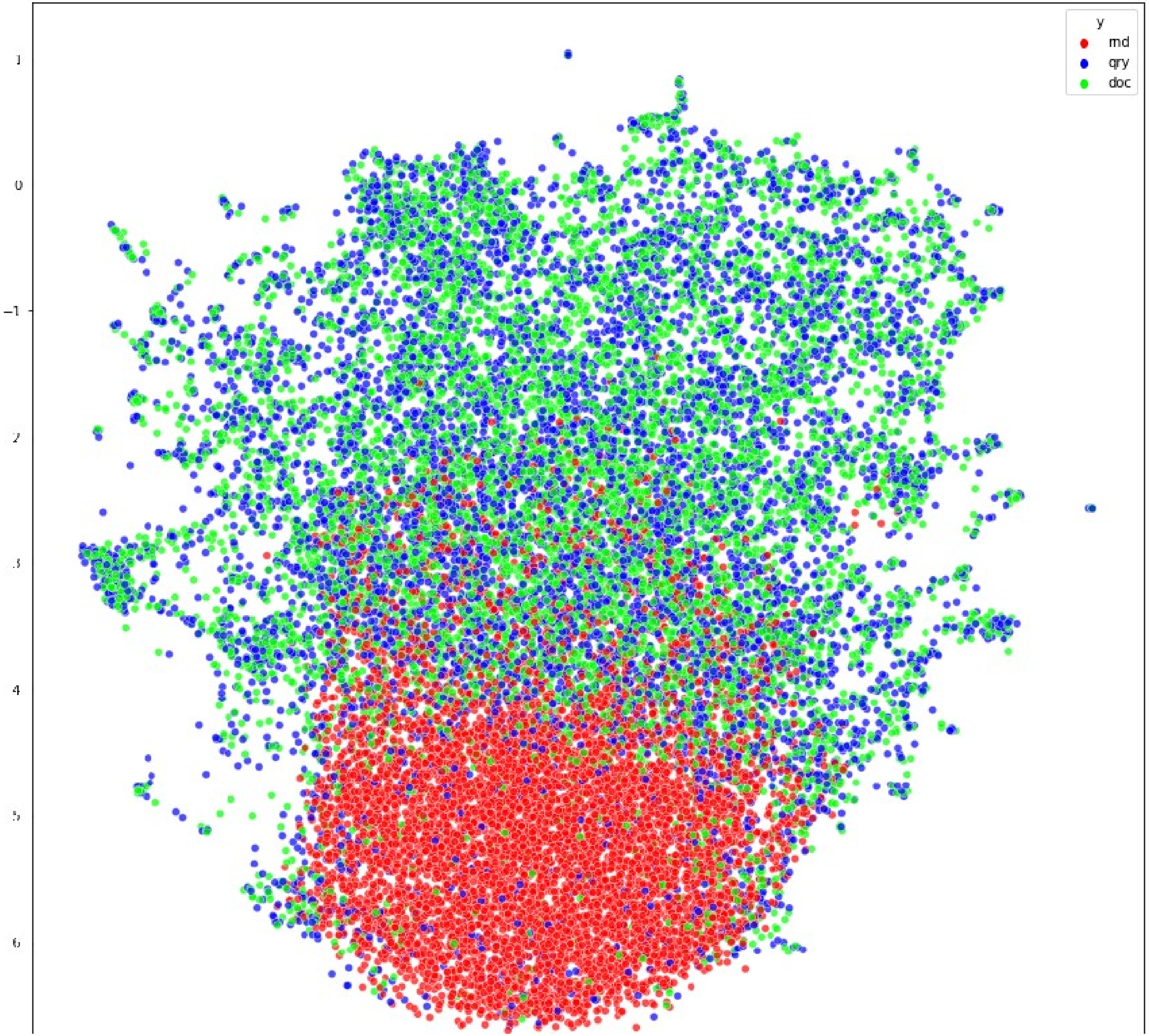
Retrieval of Random Noise

- Eval datasets are sparsely labeled
  - Only 1 or 2 passages marked as relevant
  - Drop in performance due to relevant, but unlabeled passages?

- Add noise (random strings) to the corpus
- How often is this noise retrieved at the top position?

Model	**100k**	**1M**	**10M**	**100M**
BM25 – MS MARCO	0%	0%	0%	0%
Dense Model – 128 dim MS MARCO	2.7%	4.4%	6.7%	9.7%
Dense Model – 768 dim MS MARCO	2.1%	3.7%	5.8%	8.5%
DPR (Karpukhin et al. 2020) - NQ	2.5%	5.6%	9.3%	12.1%

Vector Plot



Conclusion

- Sparse Retrieval: High Precision, low recall
- Dense Retrieval: Low Precision, high recall

- Dense retrieval works better on smaller corpora
- Dense retrieval sensitive to noise in the index
- Fewer dimensions => higher error rates

- **Evaluation results cannot be extrapolated**
  - Best system for 100k docs ≠ Best system for 100M docs