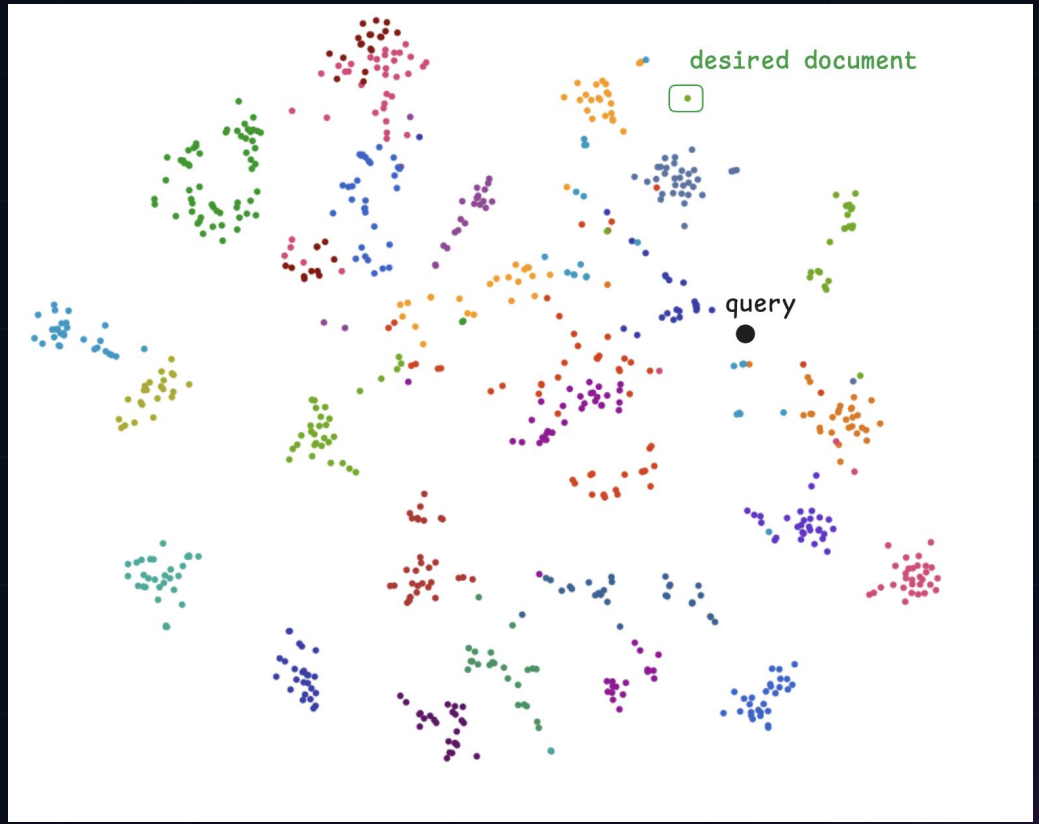
Relevance Feedback in Semantic Search at Scale

Relevance of results is model, dataset & query dependent; and we retrieve st scale with “cheaper” embedding models



How to aim for the higher relevance of the retrieved set?

There are 2 options:
(1) Query adjusting (popular) vs (2) **adjusting the scoring function**.

Why not query adjusting:

- To utilize only very small amount of feedback
- To use also uncertain feedback (who said that retrieved results will be strictly relevant or irrelevant)
- Because we have access to the whole collection of documents

Idea

Big smart model (LLM, cross-encoder, colBERT) + small retriever model

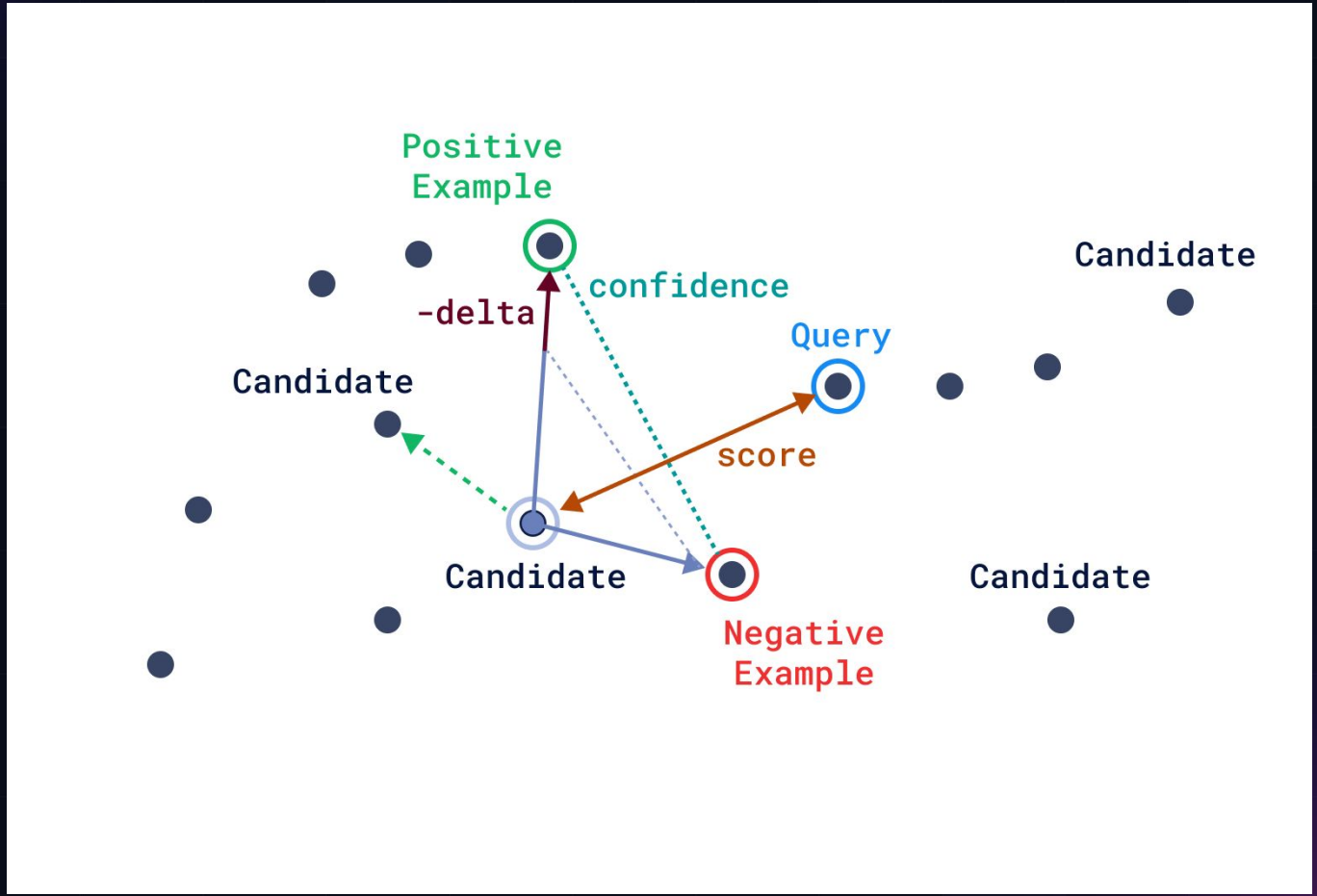
Big smart model sees top 2-5 results from the first retrieval
Forms pairs of “this is more relevant”, “this is less relevant”

We extract direction signal from these pairs and use it while traversing HNSW on the 2nd retrieval
Direction signal is extracted to the new similarity scoring formula, which is based on the feedback of the big model

Retrieval is done only with the small retriever.

$$F = a * score + confidence \hat{} b * c * delta,$$

a, b, c - trained weights per dataset & retriever-feedback-er pair on 500-1000 queries



Recall@10 with Relevance Feedback

Feedback window – top 5 retrieved results of the initial retrieval, from which we build relevance context pairs used in the rescoring

	qwen0.6B + ColBERT	mxbread + ColBERT
Msmarco	+23.2%	+2.4%
scidocs	+38.7%	+9.6%
Quora	+5.0%	+0.0%
Nfcorpus	+10.3%	+21.6%
FiQa	+6.5%	+12.2%

Interface (Qdrant 1.16.0?)

```
POST /collections/{name}/points/query
{
  "query": {
    "relevance_feedback": {
      "vector": [0.12, ..., 0.99],
      "feedback": [
        {"vector": [0.77, ..., -0.88], "score": 0.85},
        {"point_id": 9918, "score": 0.96},
        {"point_id": 12311, "score": 0.97},
        ....
      ],
      "coeffs": {
        "a": 0.12,
        "b": 1.25,
        "c": 0.99
      }
    }
  },
  "using": "my-small-vector"
}
```

Evgeniya Sukhodolskaya
Dev Advocate at Qdrant

Thank You