A Practical Approach for Few Shot Learning with SetFit for Scaling Up Search and Relevance Ranking o a Large Text Database

[n2vec]
Documents Search Engine

HAYSTACK
the search relevance conference
www.haystackconf.com

About me

Fernando Vieira da Silva

CEO of N2VEC

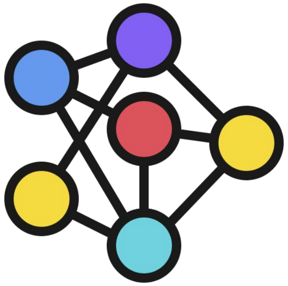
PhD in Artificial Inteligence (NLP)

+17 Years of experience in R&D

+ 8 Years of working experience in NLP

About N2VEC

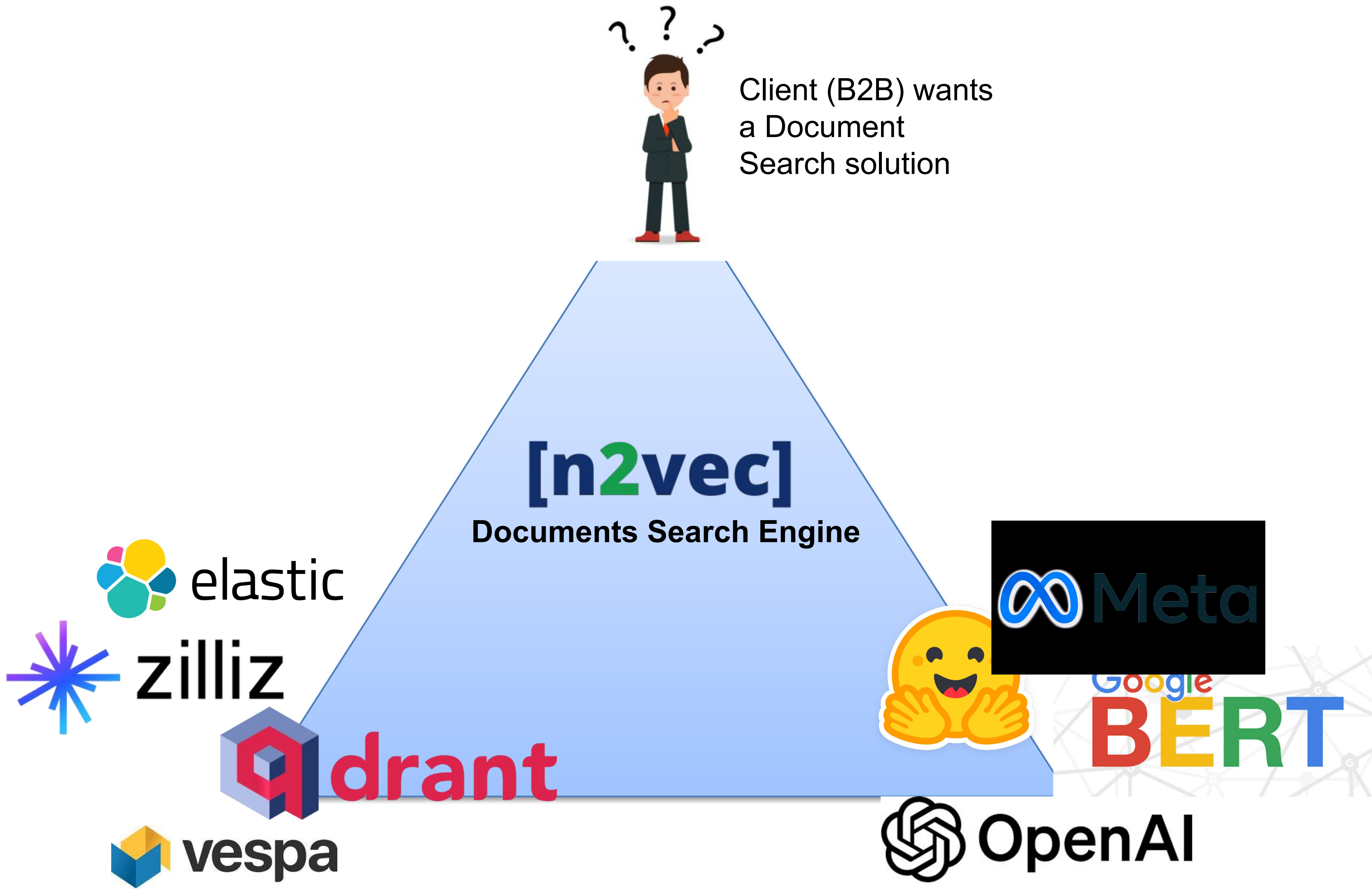
We developed a Search API for Documents or knowledge database



Our API uses Vector Databases and BERT for searching and ranking

We are based near São Paulo, Brazil

N2VEC in the ecosystem



The Problem

Handling data with few labels...

SetFit

Few-shot Fine tuning of SentenceTransformers

Competitive results compared to GPT and others

Light and fast to train (you can train in your laptop)

Multilingual support

SetFit

Efficient Few-Shot Learning Without Prompts

Lewis Tunstall[1], Nils Reimers[2], Unso Eun Seo Jo[1], Luke Bates[3],
Daniel Korat[4], Moshe Wasserblat[4], Oren Pereg[4]
[1]Hugging Face [2]cohere.ai
[3]Ubiquitous Knowledge Processing Lab, Technical University of Darmstadt
[4]Emergent AI Lab, Intel Labs
[1]firstname@huggingface.com [2]info@nils-reimers.de
[3]bates@ukp.informatik.tu-darmstadt.de
[4]firstname.lastname@intel.com

Abstract

Recent few-shot methods, such as parameter-efficient fine-tuning (PEFT) and pattern exploiting training (PET), have achieved impressive results in label-scarce settings. However, they are difficult to employ since they are subject to high variability from manually crafted prompts, and typically require billion-parameter language models to achieve high accuracy. To address these shortcomings, we propose SETFIT (**Se**ntence **T**ransformer **Fi**ne-**t**uning), an efficient and prompt-free framework for few-shot fine-tuning of Sentence Transformers (ST). SETFIT works by first fine-tuning a pretrained ST on a small number of

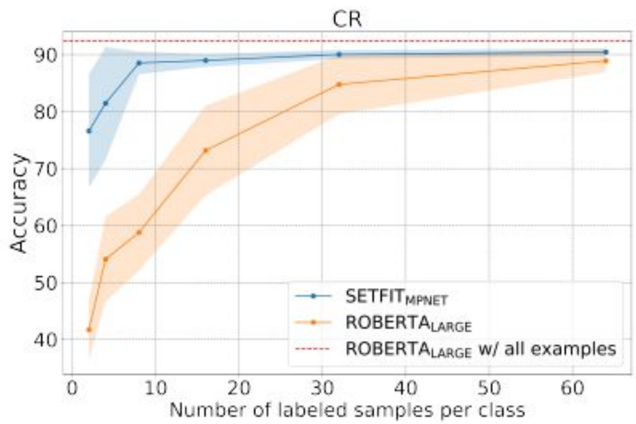
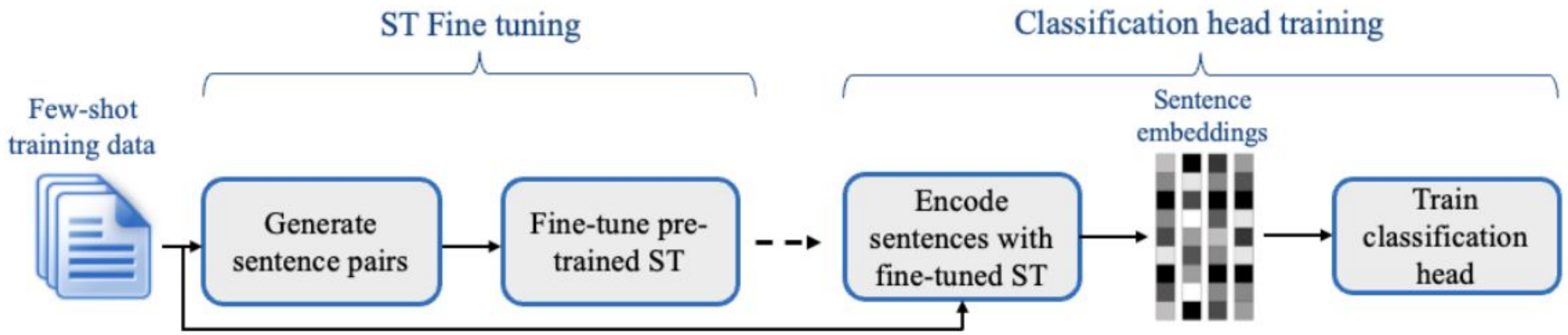


Figure 1: Compared to standard fine-tuning, SETFIT is more sample efficient and exhibits less variability when trained on a small number of labeled examples.

SetFit for Classification problems



SetFit Fine tuning and training blocks (extracted from the original article)

SetFit – Generate Sentence pairs

Original data

X	y
[0.1, 0.2, 0.3]	A
[0.4, 0.5, 0.6]	B
[0.1, 0.3, 0.3]	A
[0.1, 0.5, 0.6]	A
[0.4, 0.4, 0.6]	C

Positive triplets

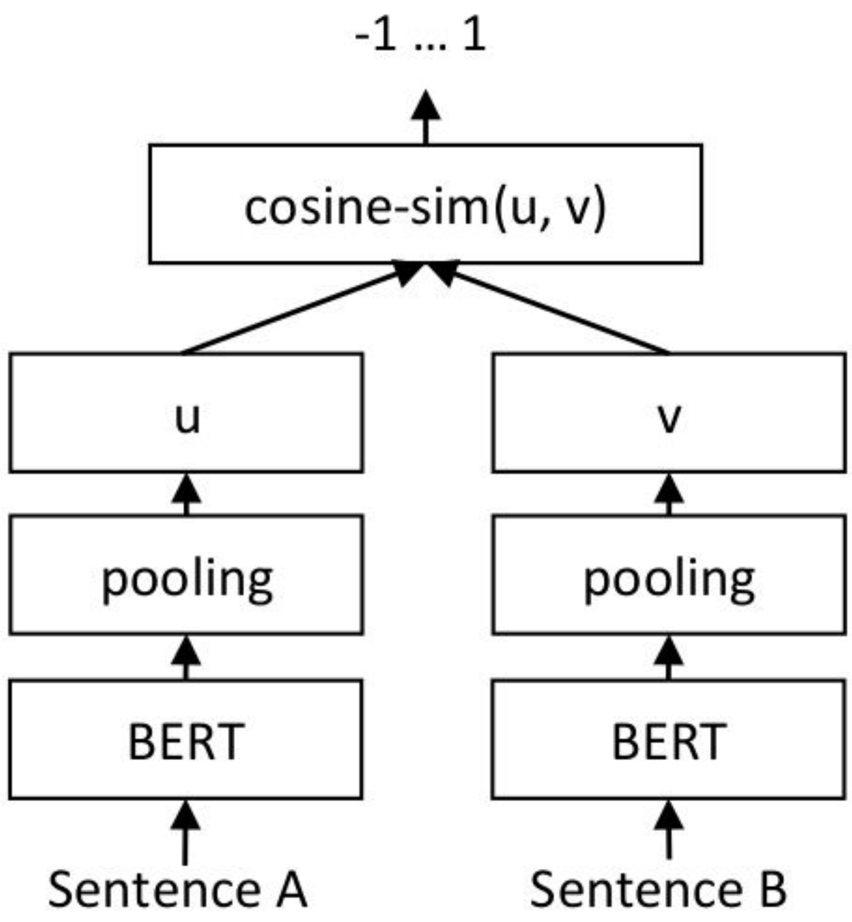
X1	X2	Y
[0.1, 0.2, 0.3]	[0.1, 0.3, 0.3]	1
[0.1, 0.3, 0.3]	[0.1, 0.5, 0.6]	1

Negative triplets

X1	X2	Y
[0.1, 0.2, 0.3]	[0.4, 0.4, 0.6]	0
[0.1, 0.3, 0.3]	[0.4, 0.5, 0.6]	0

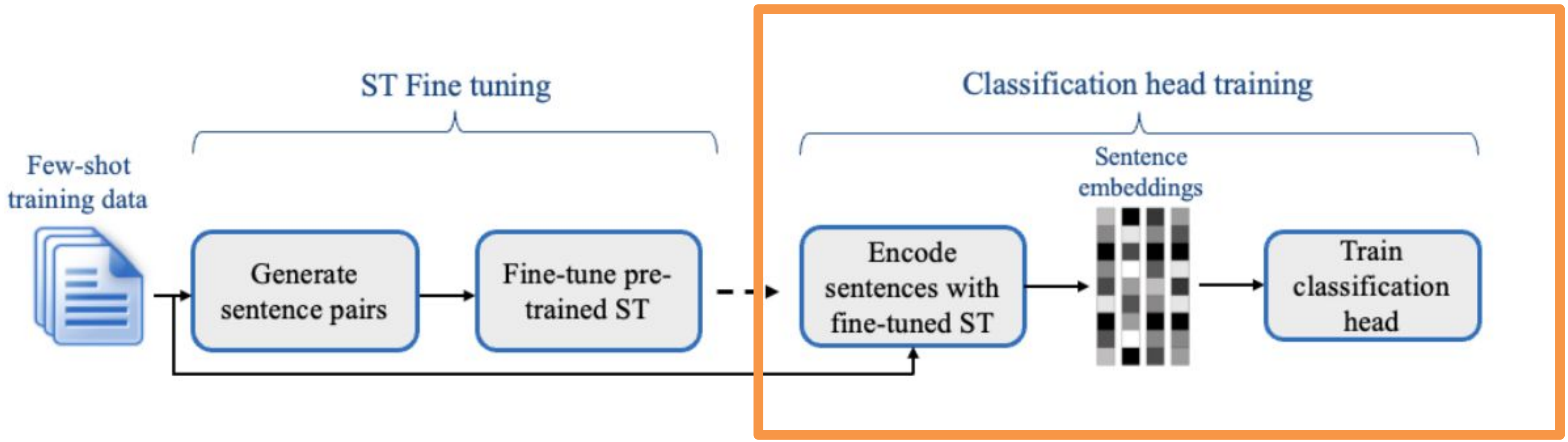
Total: K(K-1) / 2

SetFit – Sentence Transformers



Reference: https://www.sbert.net/docs/training/overview.html

SetFit for Classification Problems



SetFit Fine tuning and training blocks (extracted from the original article)

SetFit - results

Method	SST-5	AmazonCF	CR	Emotion	EnronSpam	AGNews	Average[†]
			$\lvert N\rvert = 8^*$				
FINETUNE	$33.5_{2.1}$	$9.2_{4.9}$	$58.8_{6.3}$	$28.7_{6.8}$	$85.0_{6.0}$	$81.7_{3.8}$	$43.0_{5.2}$
PERFECT	$34.9_{3.1}$	$18.1_{5.3}$	$81.5_{8.6}$	$29.8_{5.7}$	$79.3_{7.4}$	$80.8_{5.0}$	$48.7_{6.0}$
ADAPET	$50.0_{1.9}$	$19.4_{7.3}$	$91.0_{1.3}$	$46.2_{3.7}$	$85.1_{3.7}$	$\mathbf{85.1}_{2.7}$	$58.3_{3.6}$
T-FEW 3B	$\mathbf{55.0}^{\star}_{1.4}$	$19.0_{3.9}$	$\mathbf{92.1}_{1.0}$	$\mathbf{57.4}_{1.8}$	$\mathbf{93.1}_{1.6}$	–	$\mathbf{63.4}_{1.9}$
$\text{SETFIT}_{\text{MPNET}}$	$43.6_{3.0}$	$\mathbf{40.3}_{11.8}$	$88.5_{1.9}$	$48.8_{4.5}$	$90.1_{3.4}$	$82.9_{2.8}$	$62.3_{4.9}$
			$\lvert N\rvert = 64^*$				
FINETUNE	$45.9_{6.9}$	$52.8_{12.1}$	$88.9_{1.9}$	$65.0_{17.2}$	$95.9_{0.8}$	$88.4_{0.9}$	$69.7_{7.8}$
PERFECT	$49.1_{0.7}$	$\mathbf{65.1}_{5.2}$	$92.2_{0.5}$	$61.7_{2.7}$	$95.4_{1.1}$	$\mathbf{89.0}_{0.3}$	$72.7_{1.9}$
ADAPET	$54.1_{0.8}$	$54.1_{6.4}$	$92.6_{0.7}$	$72.0_{2.2}$	$96.0_{0.9}$	$88.0_{0.6}$	$73.8_{2.2}$
T-FEW 3B	$\mathbf{56.0}_{0.6}$	$34.7_{4.5}$	$\mathbf{93.1}_{1.0}$	$70.9_{1.1}$	$\mathbf{97.0}_{0.3}$	–	$70.3_{1.5}$
$\text{SETFIT}_{\text{MPNET}}$	$51.9_{0.6}$	$61.9_{2.9}$	$90.4_{0.6}$	$\mathbf{76.2}_{1.3}$	$96.1_{0.8}$	$88.0_{0.7}$	$\mathbf{75.3}_{1.3}$
			$\lvert N\rvert = Full^{**}$				
FINETUNE	59.8	80.1	92.4	92.6	99.0	93.8	84.8

Table 2: SETFIT performance score and standard deviation compared to the baselines across 6 test datasets for three training set sizes $\lvert N\rvert$. *Number of training samples per class. **Entire available training data used. [†]The AGNews dataset is excluded from the average score to enable fair comparison with T-FEW (which has AGNews in its training set). ⋆The inputs of SST-5 (but not its labels) appeared in T-FEW's training set, as part of Rotten Tomatoes dataset.

Rank	Method	Score	Size*
1	YIWISE	76.8	-
2	T-FEW 11B	75.8	11B
4	Human baseline	73.5	-
6	$\text{SETFIT}_{\text{RoBERTa}}$	71.3	355M
9	PET	69.6	235M
11	$\text{SETFIT}_{\text{MPNET}}$	66.9	110M
12	GPT-3	62.7	175B

Table 3: SETFIT compared to prominent methods on the RAFT leaderboard (as of Sept. 5, 2022). *Number of parameters.

Tables extracted from the original article

Could SetFit be Applied for Search Relevance?

Legal Research

Good precedents more chances to WIN!!!

Legal Research

65% of lawyers spend most of their **TIME** doing **Legal Research**

75% of lawyers say **Legal Research** is the **HARDEST** task

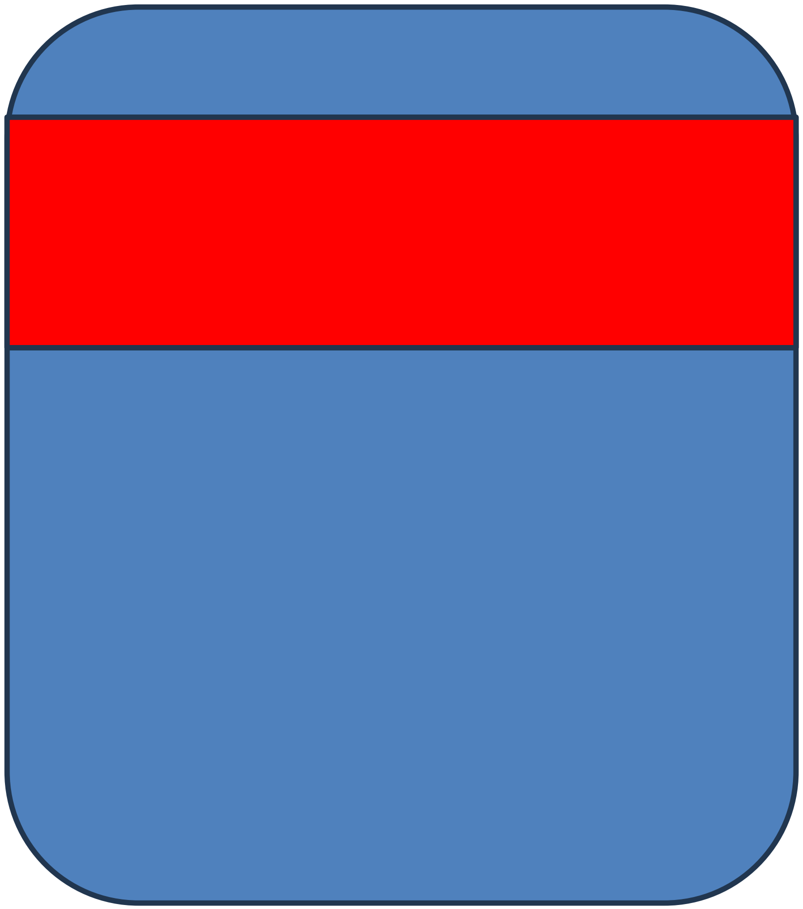
Our Legal Research Database

+ 60 million sentences extracted from legal decisions

Precedents from 8 courts in Brazil

+ 7,800 results annotated by lawyers

First Step: Using SetFit for sentences Classification



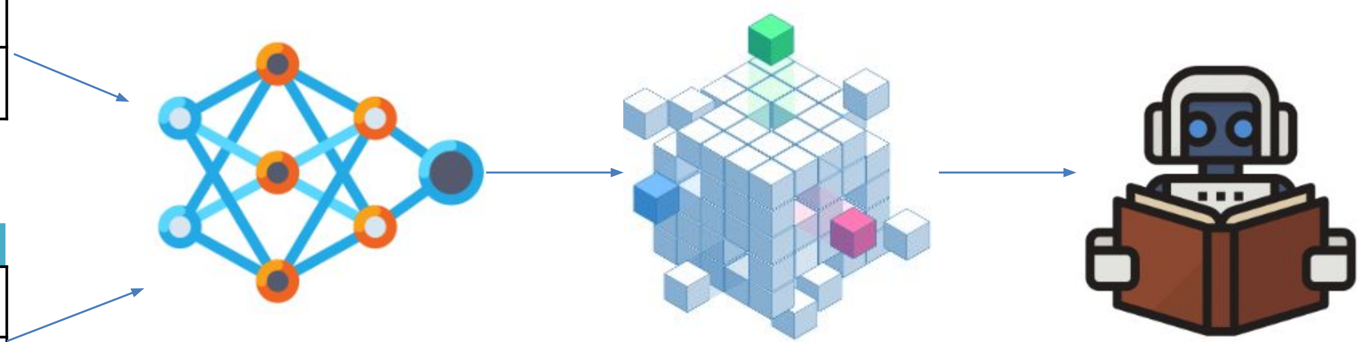
A fraction of the dataset was labeled according to the theme (i.e. "traffic accident", "consumer rights", "health")

But too many classes with few labels each… (9K split in 138 classes)

First Step: Using SetFit for sentences Classification

X1	X2	Y
[0.1, 0.2, 0.3]	[0.1, 0.3, 0.3]	1
[0.1, 0.3, 0.3]	[0.1, 0.5, 0.6]	1

X1	X2	Y
[0.1, 0.2, 0.3]	[0.4, 0.4, 0.6]	0
[0.1, 0.3, 0.3]	[0.4, 0.5, 0.6]	0

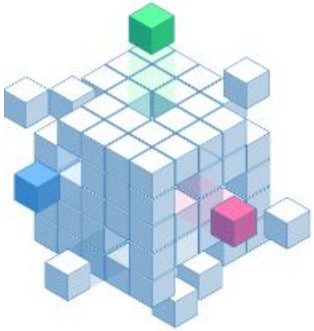


Fine tune
Sentence
Transformer

Fine-tuned
ST

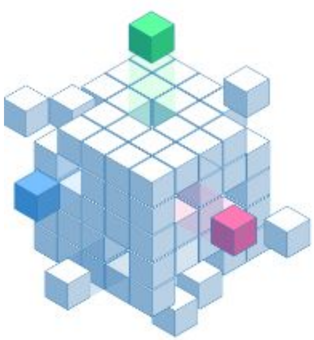
Classifier

First Step: Using SetFit for sentences Classification



Fine-tuned
ST

Second Step: Fine tuning with queries/results

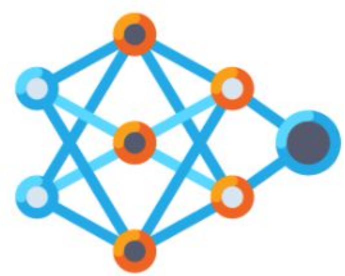


Fine-tuned
ST

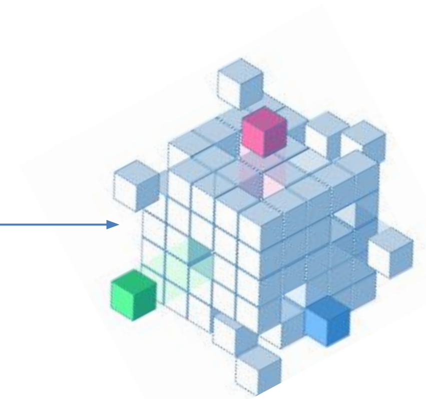
X1	X2	Y
[0.1, 0.2, 0.3]	[0.1, 0.3, 0.3]	1
[0.1, 0.3, 0.3]	[0.1, 0.5, 0.6]	1

X1	X2	Y
[0.1, 0.2, 0.3]	[0.4, 0.4, 0.6]	0
[0.1, 0.3, 0.3]	[0.4, 0.5, 0.6]	0

Annotated triplets
augmented by
negative triplets

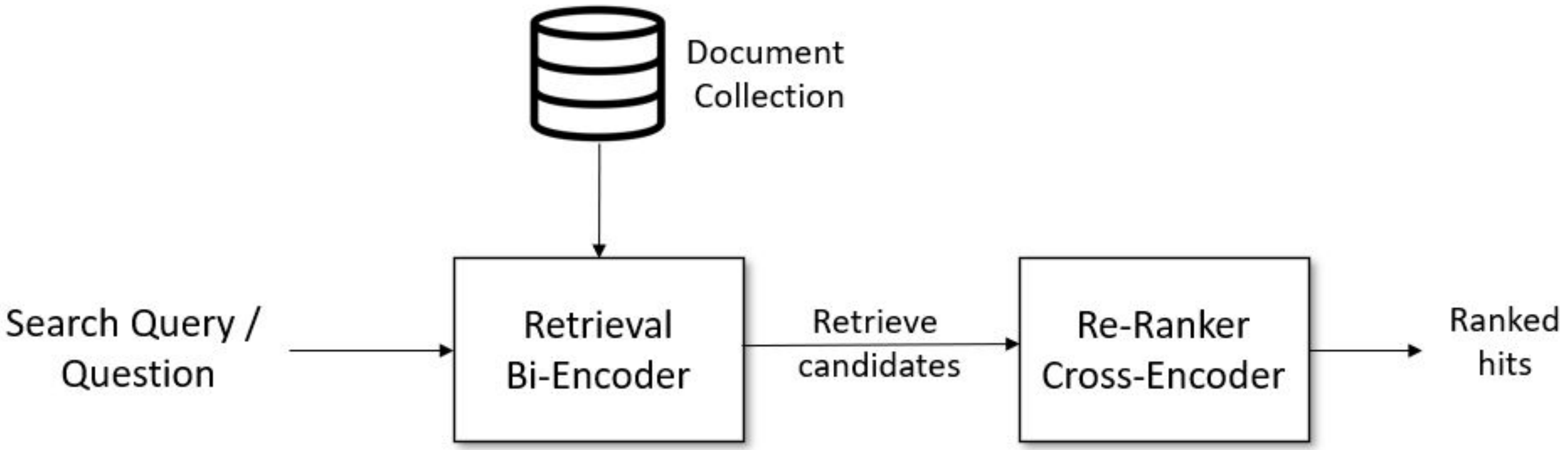


Fine tune
Sentence
Transformer



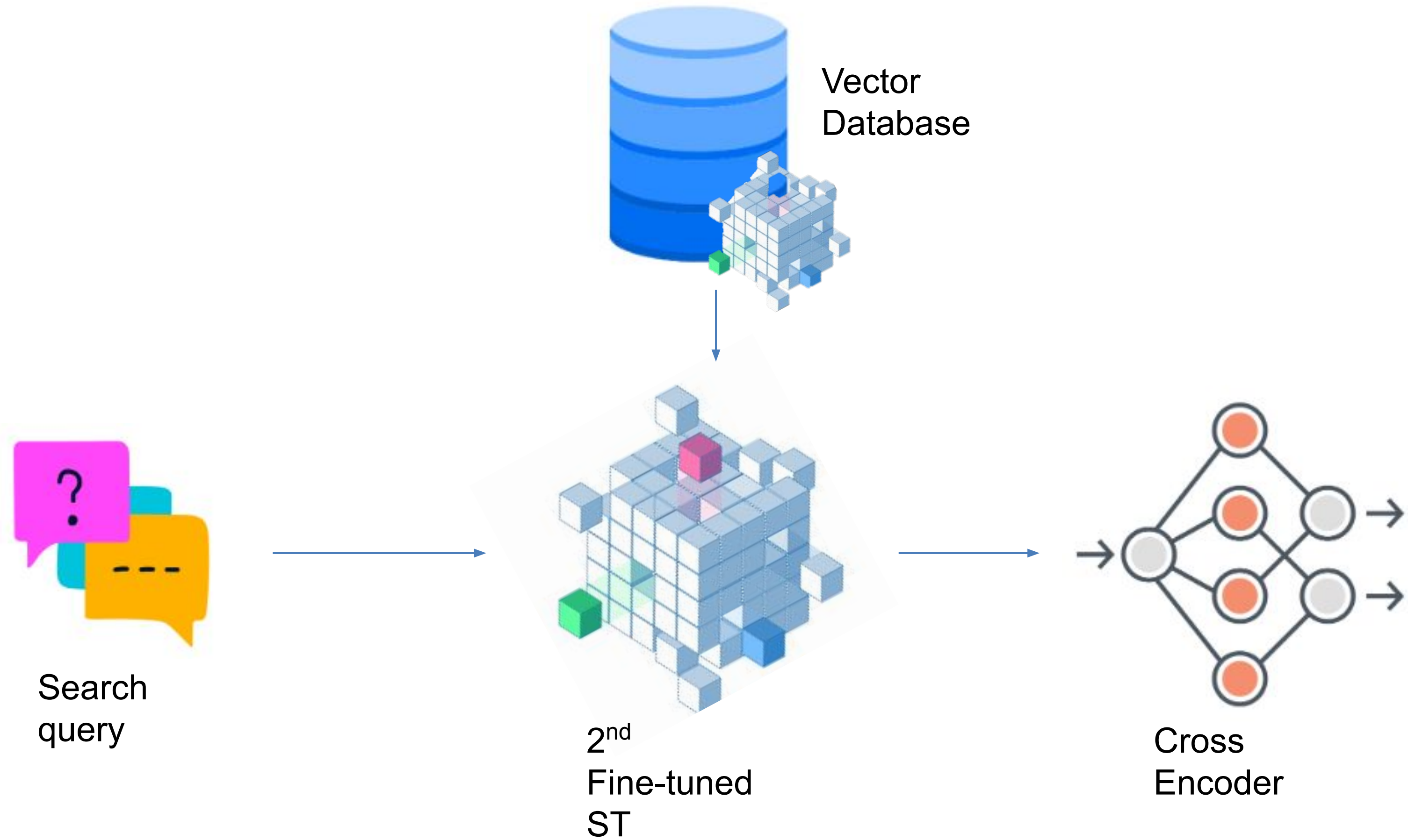
2nd
Fine-tuned
ST

Third Step: Implement Search and Ranking



Reference: https://www.sbert.net/examples/applications/information-retrieval/README.html

Third Step: Implement Search and Ranking



Results

ST + Cross Encoder

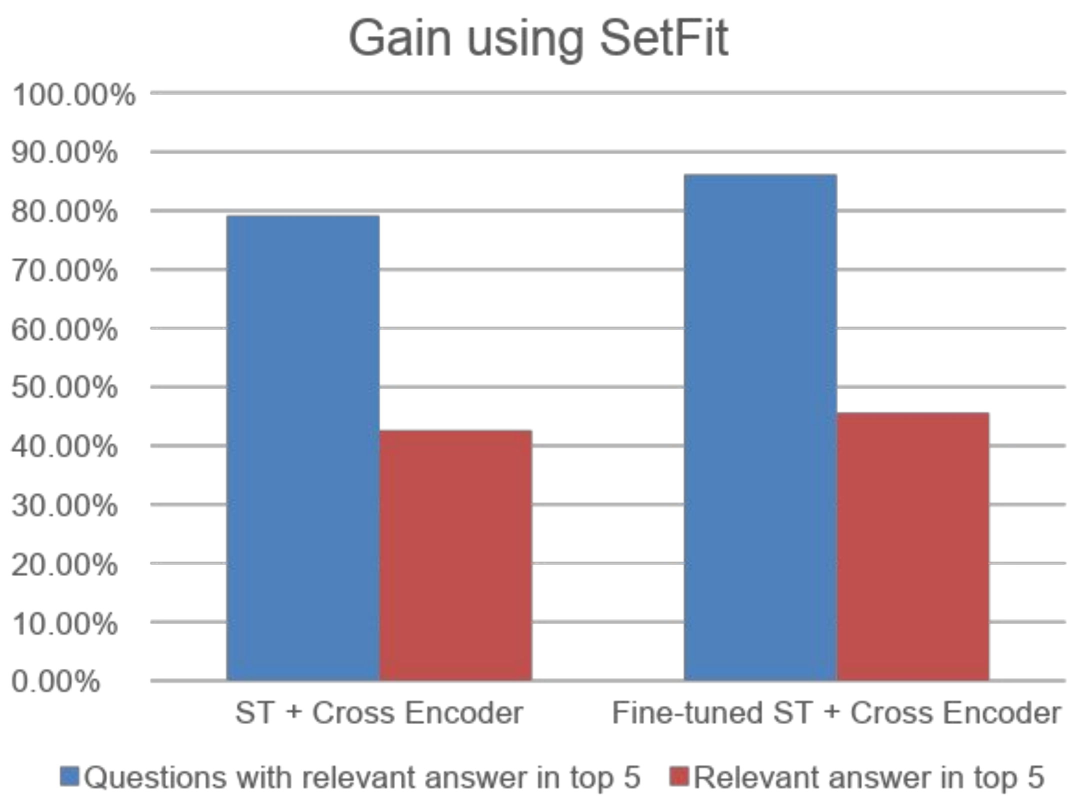
79.1% of questions with relevant answers in top 5

42.6% of relevant answers in top 5

Fine-tuned ST + CrossEncoder

86.1% of questions with relevant answers in top 5

45.6% of relevant answers in top 5

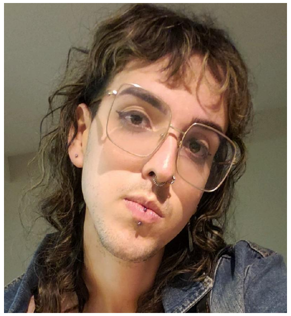


Acknowledgement

N2VEC Team:

Pedro Kim
Machine Learning Engineer

Gustavo Sarti
Machine Learning Engineer



Yara Campos
Data Enginer

Alfonso Guerra
Python SW Engineer

fernando@n2vec.com
+55 15 98801 2165