VeriDevOps

NLP for Security Requirements Analysis - Practical Examples

By Andrey Sadovykh

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957212

Agenda

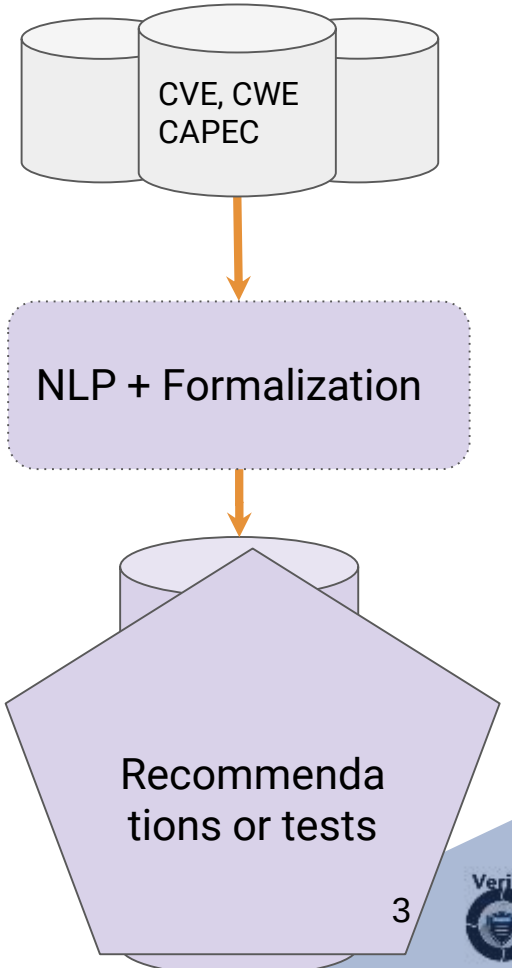
1. Motivation
2. Objectives
3. Related work
4. Practical examples and demo

NLP for security specification modelling

- Main objectives:
  - Extract security requirements from unstructured text
  - Classify security requirements
  - Identify entities and properties
  - Apply formal specification patterns

- Results:
  - Concrete recommendations or tests

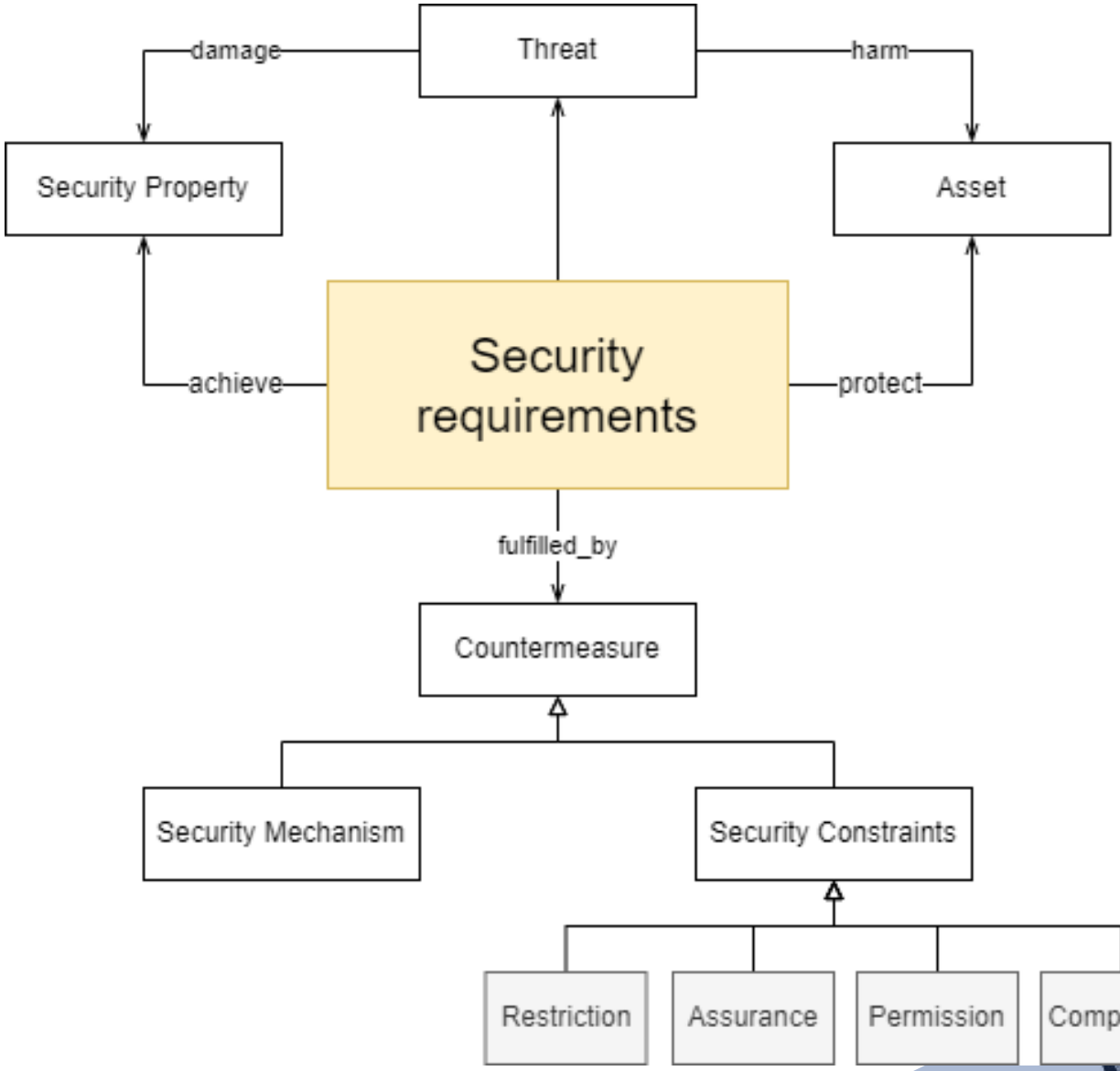
Global security requirements (eg IEC62443), specific security requirements, vulnerability and attack descriptions



NLP For Requirements Engineering

Security requirements taxonomy

Eliminate threat to an asset by achieving security property by implementing a countermeasure.



NLP Methods (ML) - 1

- Classification task in machine learning (ML) - predicting a categorical class

- Extraction - retrieving some specific single or multi-word terms from requirement texts for domain or project glossaries

- Clustering - organizing documents into cohesive subsets or clusters

NLP Methods (ML) - 2

Detection - removing ambiguities in requirements to make them clearer and unequivocal. Main goal - to maintain correctness of requirements texts

- detection of different lexical issues from the debatable usage of grammatical rules
- occurrence of vague phrases (e.g., after some time), weak verbs (e.g., may, might)
- appearance of syntactic ambiguities
- following predefined templates
- recognizing equivalent requirements

NLP Methods (ML) - 3

Vulnerability detection

- identifying vulnerable software components prior to deployment, either by statically analyzing software code, or by executing security testing tools on a running instance of the software.

Vulnerability repair

- transforming a vulnerable code into a non-vulnerable code by learning from a set of source examples.

Specification analysis

- dealing with security risks in products before the code is even written.
- expert methods to automatically process vulnerability descriptions or product specifications to assess security risks.

Performance evaluation

Precision = TP / (TP+FP)

- correctly identified requirements

Recall = TP / (TP+FN)

- missed requirements

F1 = 2TP/(2TP+FN+FP)

- ranking

Deep Learning

Consecutive transformations of representation at one level into a higher, more abstract level. In NLP - Word2Vec for each word by a set of convolution filters.

- Winkler et al. [29] requirements classification with precision of 73% and recall of 89%. F1 = 80%
- Dekhtyar et al. Word2Vec with CNN on SecReq. F1 = 91.34%

Transfer Learning Methods

Trained on huge datasets to capture underlying concepts and meanings of natural language texts

- Bidirectional Encoder Representations from Transformers (BERT) [31]
- Fine-tuned with NFR dataset [25]
- Resulting NoRBERT
  - Functional requirements F1 - 90%
  - Non-functional requirement F1 - 93%
  - Security requirements F1 - 91%

Practical examples

- Requirements Extraction
- Security Requirements Classification
- Mapping to Security Technology Implementation Guidelines

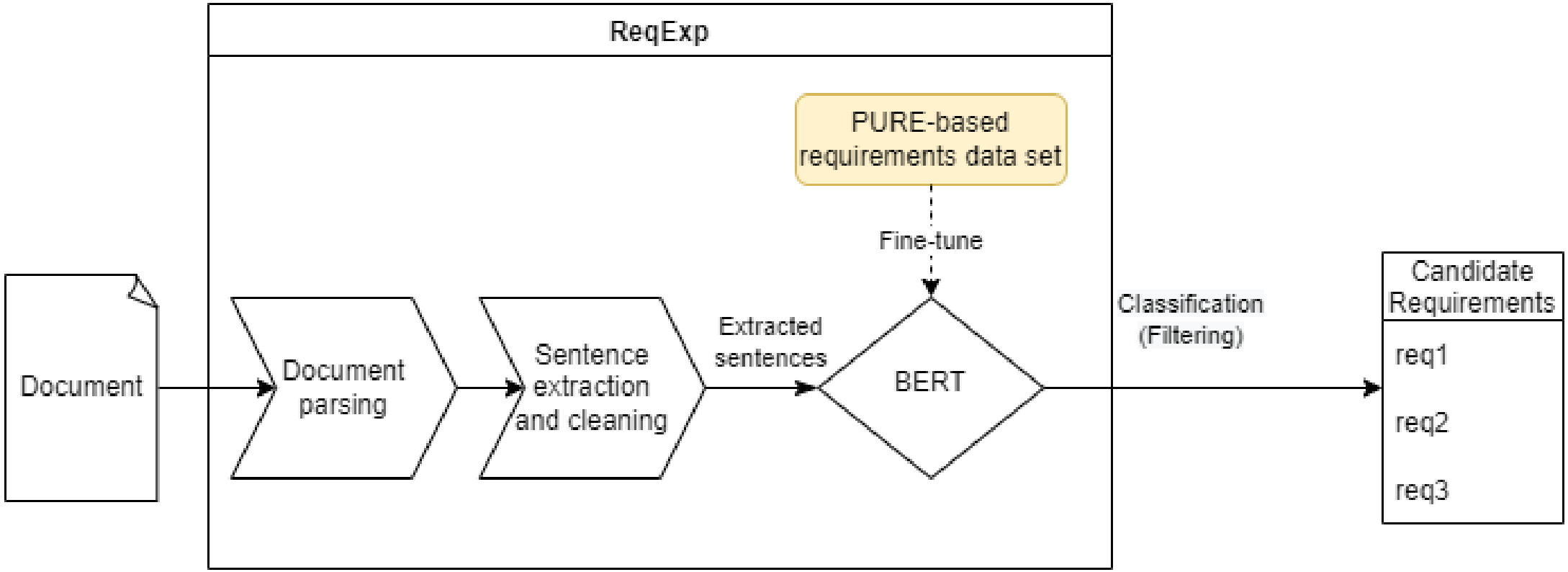
Challenges that we address

1. Requirements are specified in various forms, styles and lexical constructs.
   - What is non-requirement?

2. Security Requirements datasets are relatively small
   - Categorization is difficult or impossible

3. Security Requirements often quite vague, they need to be mapped to concrete practices.

ReqExp prototype

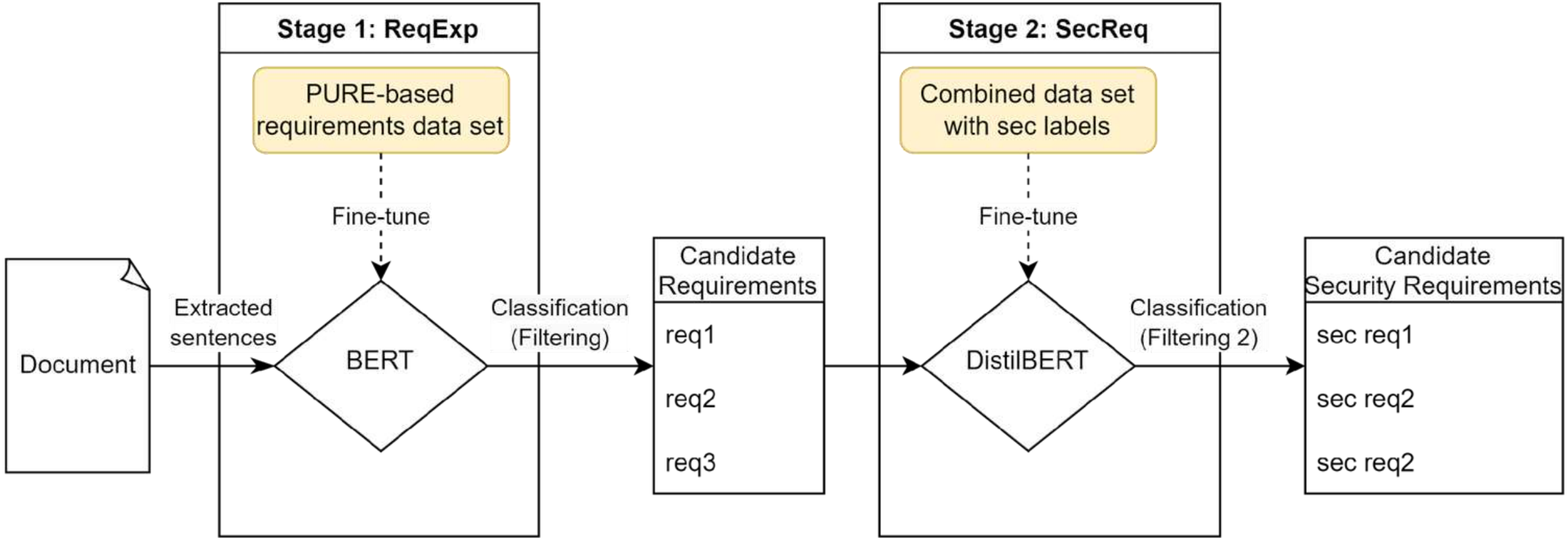


Requirements Extraction

- Dataset: PURE corpus, 79 SRS documents -> 30 documents
- 7745 requirement/non-requirement sentences
- 4145 were requirements and 3600 were non-requirements

Model	F1	P	R	TP	TN	FP	FN
Fasttext	.81	.72	.93	763	419	295	57
ELMO+SVM	.83	.78	.88	827	364	231	112
BERT	.86	.92	.80	841	407	69	217

SecReq Prototype

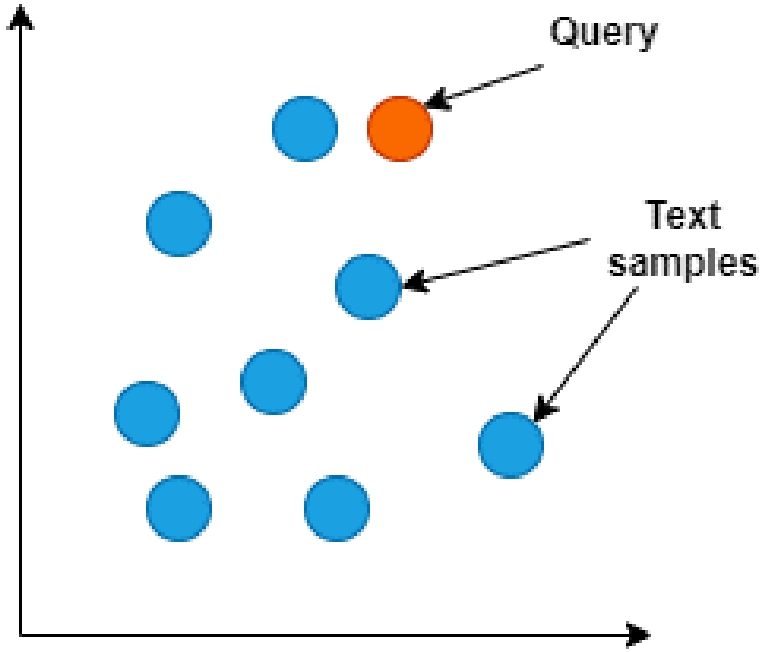


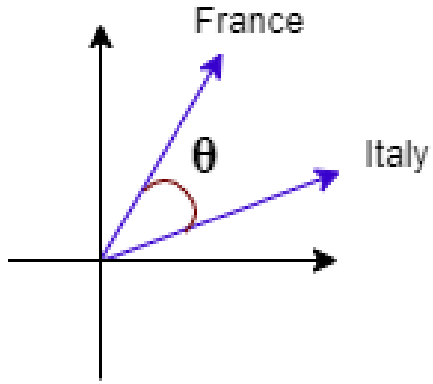
SecReq Prototype

- Datasets
  - Binary: PURE data set relabelled - 7745 of statements including security and non-security
  - Binary: SecReq, PROMISE, CCHIT, Concordia, OWASP - 2328 of security and non security
  - Multiclass: PURE + Secreq + Riaz - 1000 categorized security requirements

Result: Stage 2, F1-score of 0.86

Semantic search

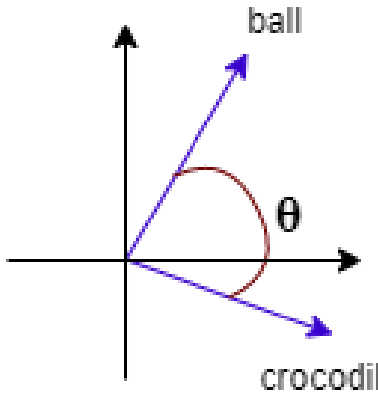




France and Italy are quite similar

θ is close to 0^{o}

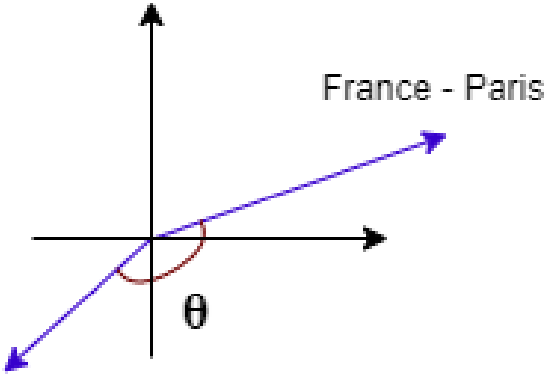
$\cos(\theta) \approx 1$



ball and crocodile are not similar

θ is close to 90^{o}

$\cos(\theta) \approx 0$



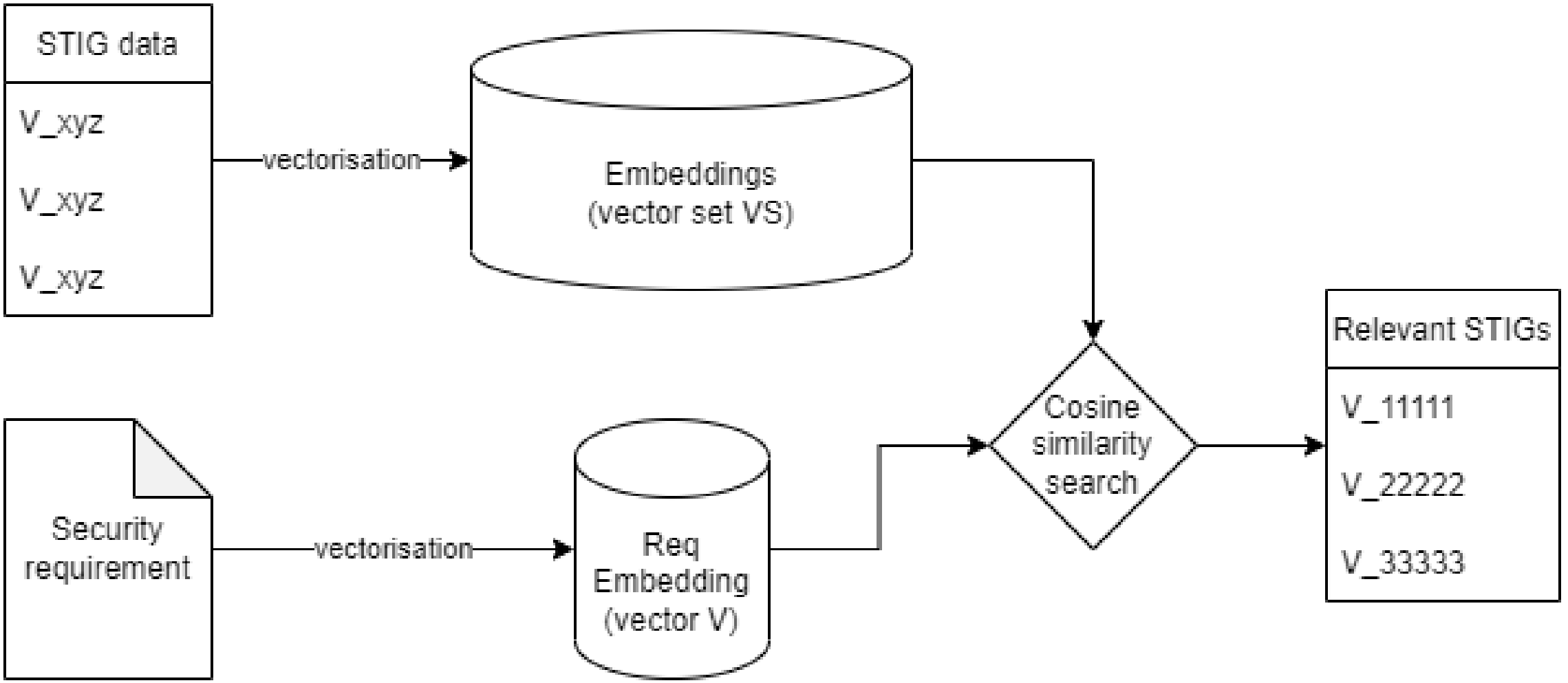
Rome - Italy

The two vectors are similar but opposite
the first one encodes (city - country)
while the second one encodes (country - city)

θ is close to 180^{o}

$\cos(\theta) \approx -1$

STIGsearch Prototype



Challenges ahead

- Performance of ML. High resource demand.
- Evaluating relevance in semantic search
- CI/CD integration
- Further case studies

Next

Time	Duration	Topic	Presenter	Organization
9:30	20 mins	VeriDevOps Technical Introduction	Andrey Sadovykh	SOFTEAM
Part I: Security Requirements Engineering				
9:50	20 mins	A Taxonomy of Vulnerabilities, Attacks, and Security Solutions in Industrial PLCs.	Eduard Paul Enoiu	Mälardalen University
10:10	20 mins	Natural Language Processing with Machine Learning for Security Requirements Analysis - Practical Approaches.	Andrey Sadovykh	SOFTEAM
10:30	20 mins	Security Requirements Formalization with RQCODE.	Andrey Sadovykh	SOFTEAM
10:50	10 mins	break	/	/

Thank You

Contact: Andrey Sadovykh, SOFTEAM

https://www.veridevops.eu/

http://arqan.softeam-rd.eu:8501/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 957212