TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

AG·SE

Automatic Extraction of Conceptual Interoperability Constraints from API Documentation

"Master Thesis"

Mohammed Abujayyab

First supervisor: Prof. Dr. Dr. h.c. H. Dieter Rombach

Second supervisor: Hadil Abukwaik, MSc.

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Outline

- Background
- Motivation scenario
- Problem
- Research methodology
- Research part one
- Research part two
- Conclusion and future work

Background

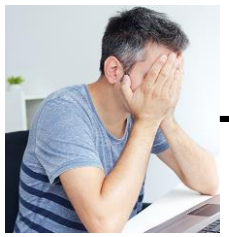
Conceptual Interoperability Constraints (COINs)

are the restrictions on interoperable software units and their related data elements at different conceptual levels (i.e., syntax, semantics, structure, dynamics, context, and quality) [1].

Category	COIN name	Examples of value
Syntax	Lexical references	Dictionary, thesaurus, glossary, etc.
	Modeling lang.	XML, UML, ADL, WSDL, etc.
Semantic	Semantic references	Reference ontologies
	Semantic constraints	Data units and scale ratio
Structure	Data structural constraints	Invariants, inherited constraints, and multiplicity constraints
	Data storage	Relational database, flat files, etc.
	Distribution	Distributed, centralized
	Encapsulation	Encapsulated , not encapsulated
	Concurrency	Single-threaded, multi-threaded
	Layering	Layered, not layered
Dynamic	Data change	Periodic, irregular, continuous, etc.
	Service conditions	Pre, post, and time conditions
	Interaction property	State(ful/less), (a)synchronous, etc.
	Interaction time constraints	Session timeline, acknowledgment timeline, response timeline, etc.
	Communication style	Messaging, procedure call, blackboard, streaming
Context	Usage context	device type, wired/wireless, access rate, time, location, etc.
	Intended users	Human/machine, gender, age, etc.
Quality	Data quality	Security, trust, accuracy, etc.
	Service quality	Safety, availability, efficiency, etc.

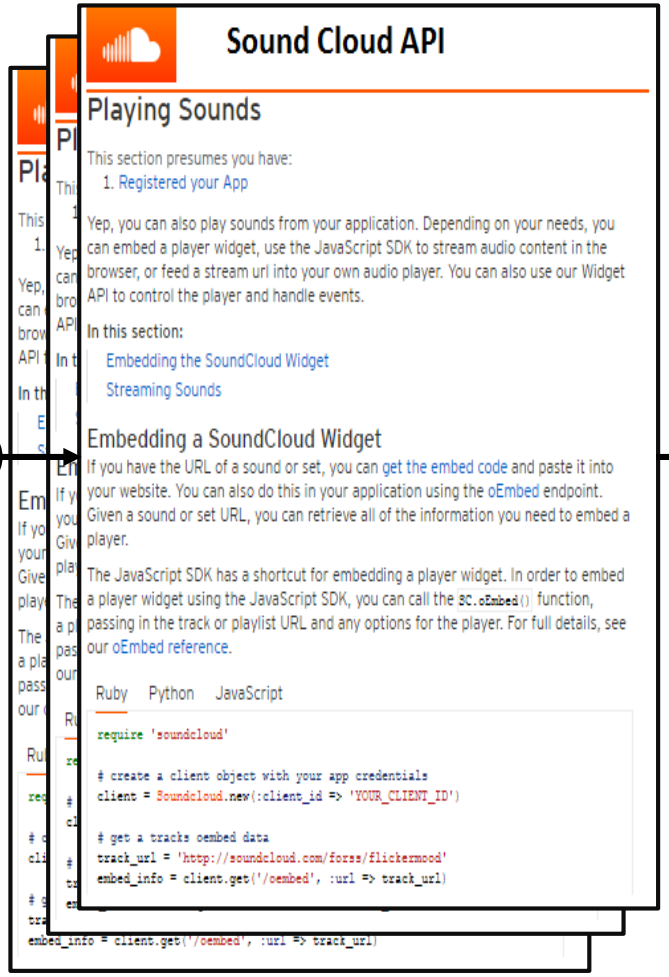
Conceptual Interoperability Constraints [1]

Motivation scenario



Software Architects / Analysts

Read



Sound Cloud API documentation
<Input>

Find COINs *manually*

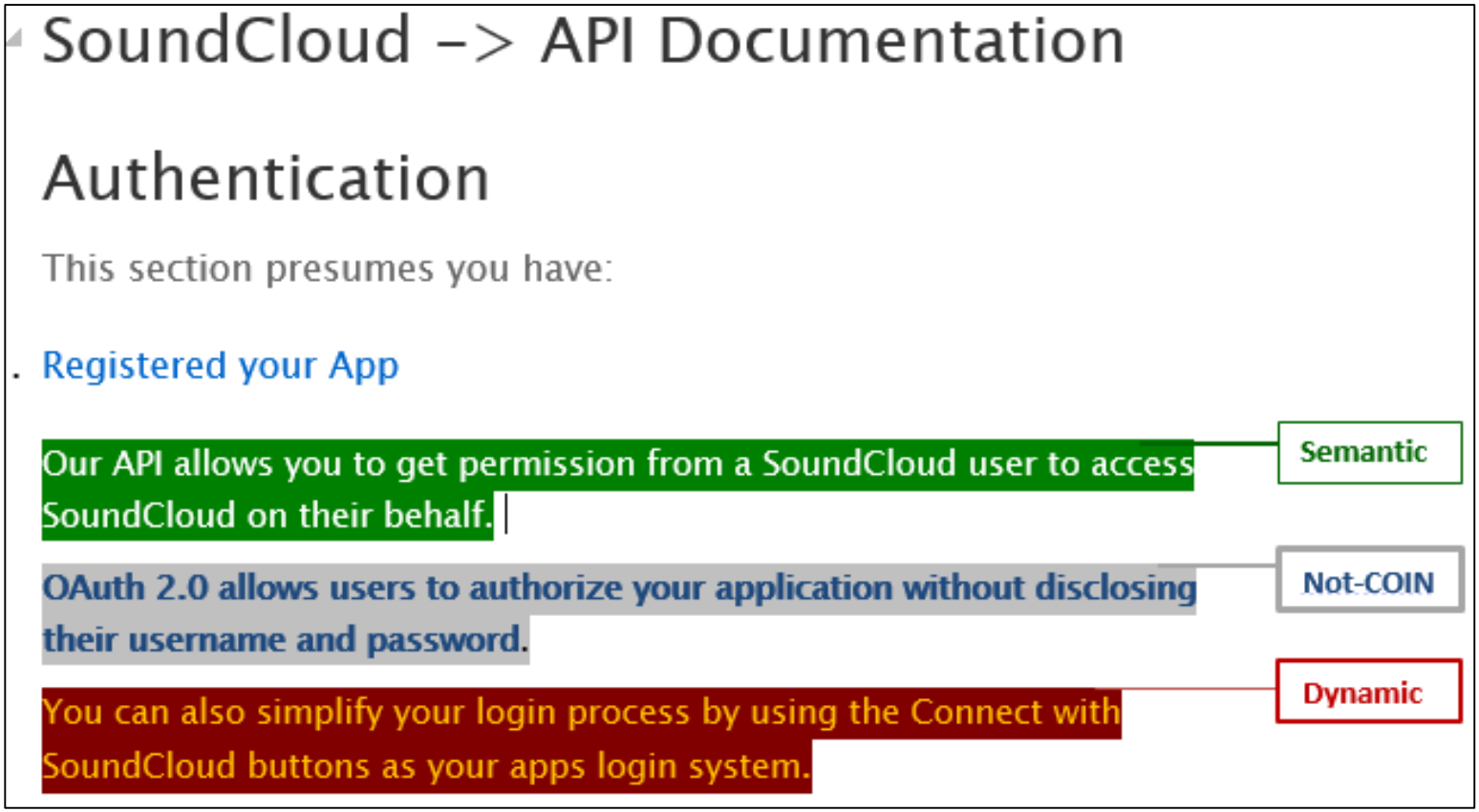
COINs

1- Not-COIN
2- Dynamic
3- Semantic
4- Syntax
5- Structure
6- Context
7- Quality

COIN class
<Output>

Motivation scenario

Example:



https://developers.soundcloud.com/docs/api

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Problem

- **Time**
  - For example, it took one of the authors more than **10 hours** to only browse (reading) the documentation of the Ebay web service operation [7].
- **Mental Effort**
  - Linguistic and analytical skills
  - API reading experiences
- **Accuracy**
  - Human analysis can be error prone
    - Missing COINs
    - Wrong COINs

Goal & Research Questions

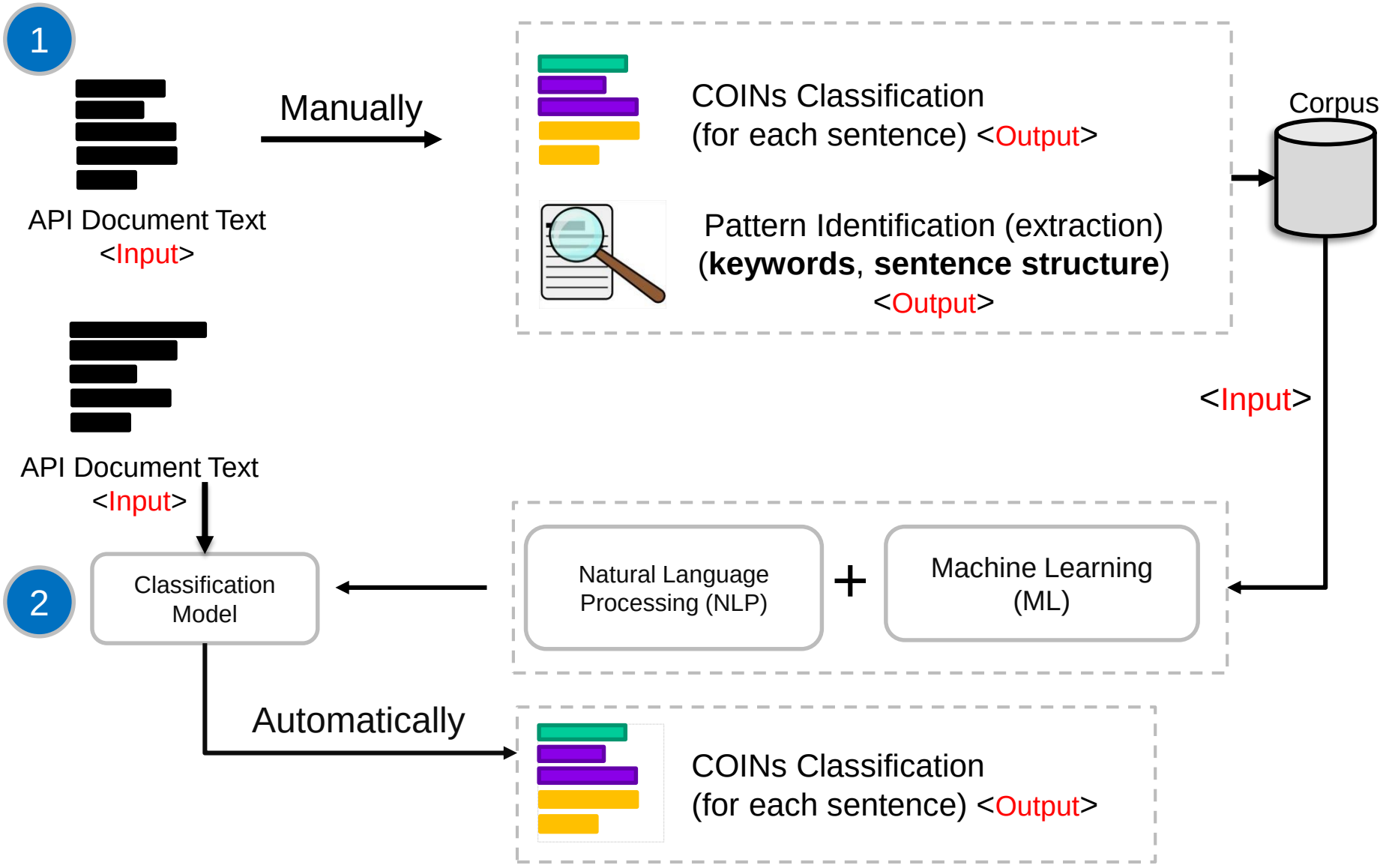
Goal

- **To**: support the conceptual interoperability analysis task.
- **For the purpose** of: improvement.
- **With respect to**: effectiveness and efficiency of detecting COINs.
- **From the viewpoint of**: software architects and analysts.
- **In the context of**: analyzing text in API documentation within integration projects.

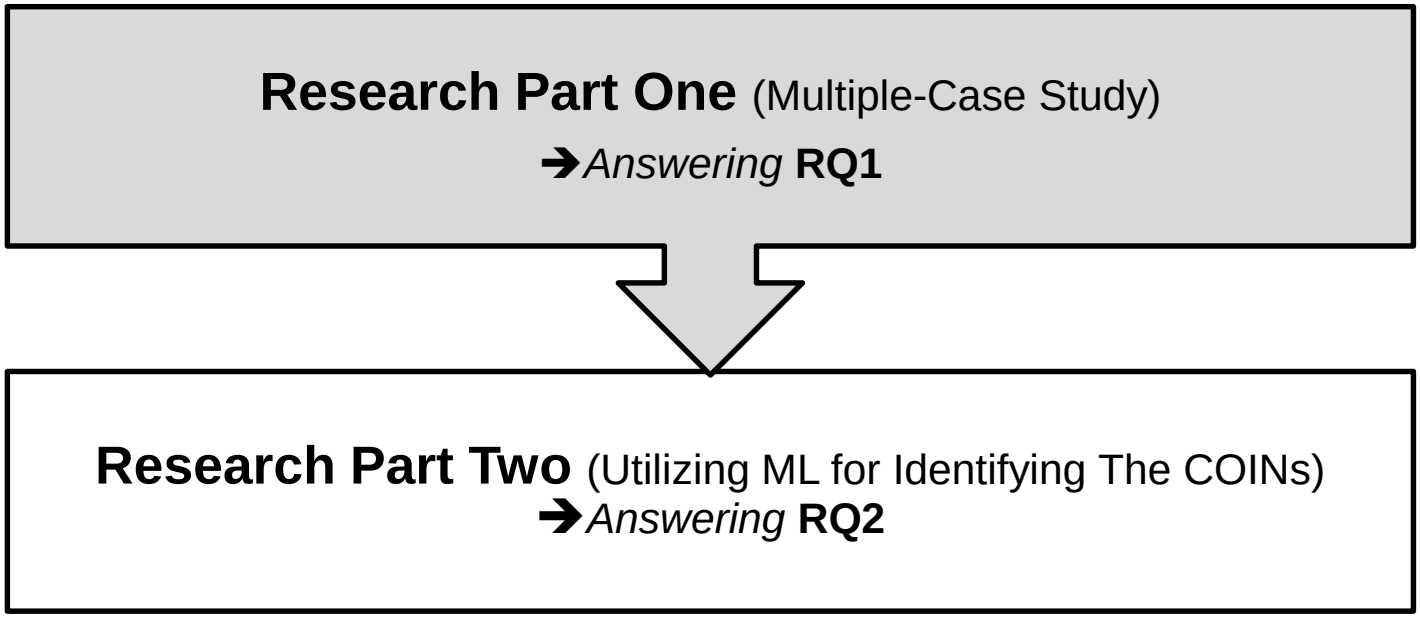
- **RQ1:** What are the observed **patterns** in specifying the conceptual interoperability constraints COINs in the NL text of API documentation?

- **RQ2:** How effective and efficient would it be to use Natural Language Processing (**NLP**) along with Machine Learning (**ML**) technologies to **automate** the extraction of COINs from the text in API documentations?

Idea (overview)



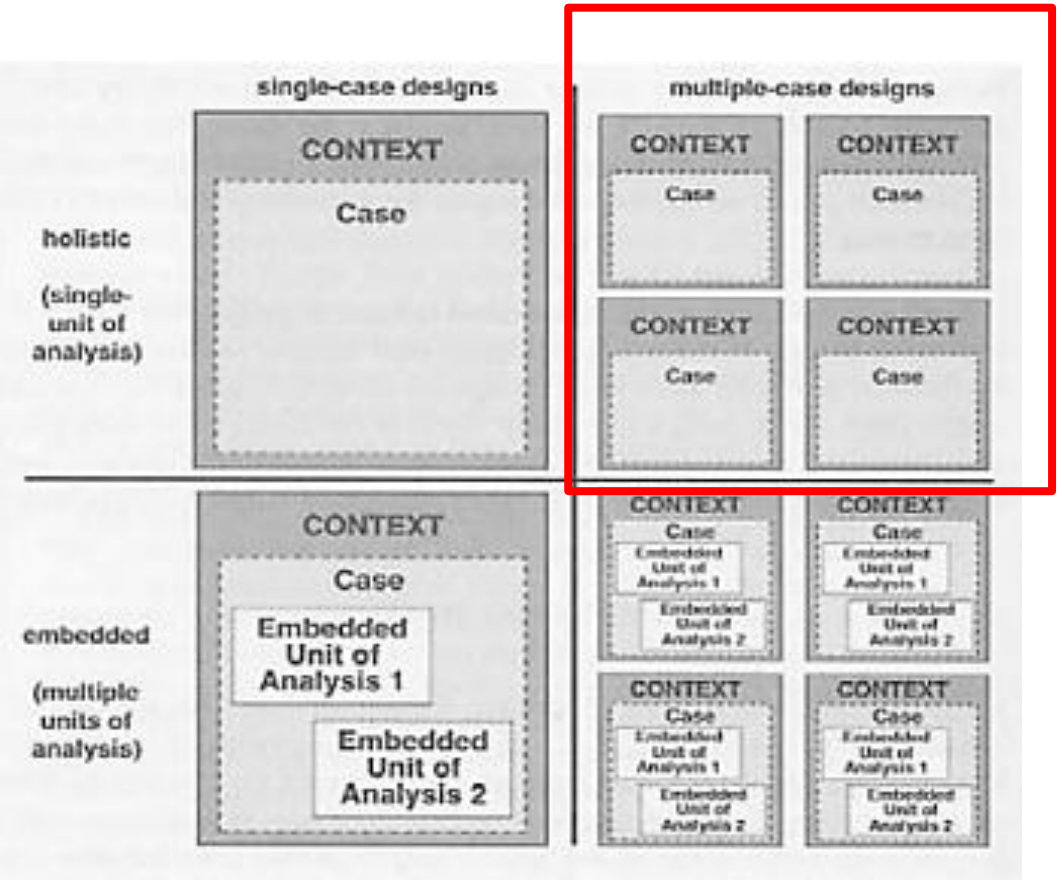
Research methodology



Research part one (Study design)

Holistic multiple-case study (Action Research)

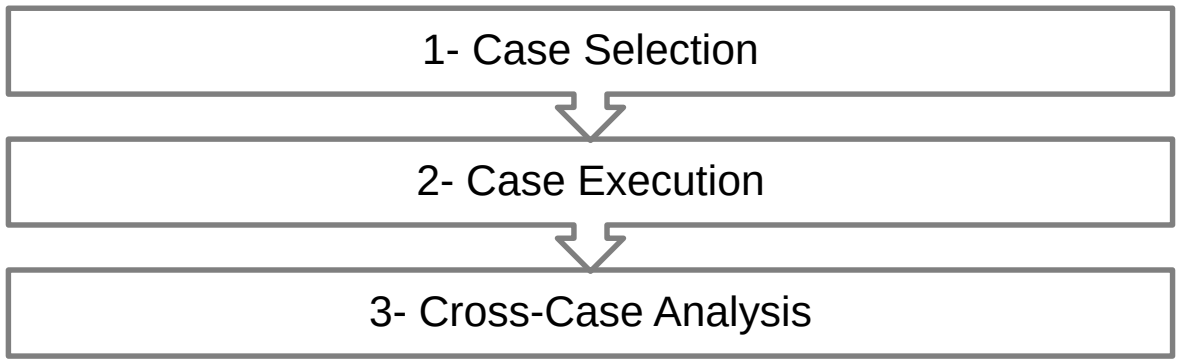
- with *literal replication* of cases from different domains



Holistic multiple-case study [3]

Research part one (Study Execution)

Study protocol. three main activities:



Research part one (Study Execution)

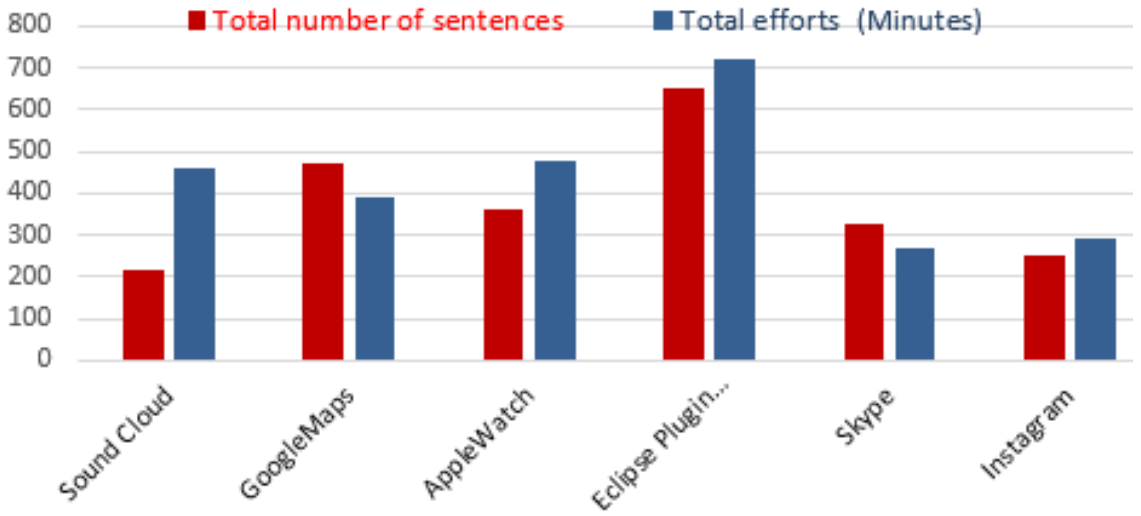
1- Case Selection

2- Case Execution.

3- Cross-Case Analysis.

- Case selection criteria (six cases)
  - API type: Platform API, Web-Service
  - API popularity
  - API domain: music, maps, development

API Document	Total number of sentences	Document manual filtering (Minutes)	Sentence Classification (Hours)	Total efforts (Hours)	Total efforts (Minutes)
Sound Cloud	219	40	7	7.7	460
GoogleMaps	473	60	5.5	6.5	390
AppleWatch	360	60	7	8.0	480
Eclipse Plugin Dev	651	60	11	12.0	720
Skype	325	30	4	4.5	270
Instagram	253	20	4.5	4.8	290
Total	**2281**	**270**	**39**	**43.5**	**2610**



Research part one (Study Execution)

1- Case Selection

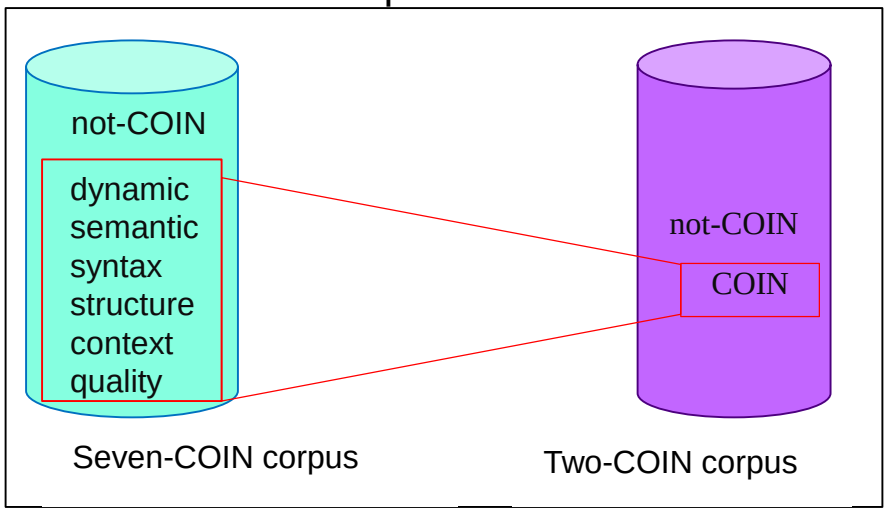
2- Case Execution

2.1 Manual Classification

2.2 Pattern Identification

3- Cross-Case Analysis.

- **Manual Classification (Building the corpus)**
- **Input: API document**
- **Output: COIN Corpus**
  - Seven-COIN corpus
  - Two-COIN corpus structure



Example: from SoundCloud API:

Our API gives you the ability to upload, manage and share sounds on the web. — **Semantic**

Research part one (Study Execution)

1- Case Selection

2- Case Execution
2.1 Manual Classification
2.2 Pattern Identification

3- Cross-Case Analysis.

- **Pattern identification**

Snapshot from GoogleMaps API documentation

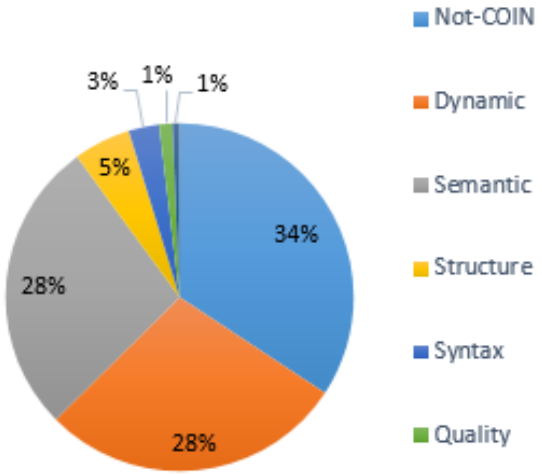
Sentence	detected patterns					
	1	2	3	4	5	6
	Input/Output	explanation	Conditional statement	Technical terms	Structure Terms	Method call
these web services use **HTTP requests** to specific URLs, passing URL parameters as arguments to the services.	requests			HTTP		
for example,? is used within URLs to indicate the beginning of the **query** string.		for example			query	
when processing **XML responses**, you should use an appropriate query language for selecting **nodes** within the **XML document**, rather than assume the **elements** reside at absolute positions within the **XML** markup.	response		when	XML	nodes, elements, document	
by default, XPath expressions match all elements.				XPath	elements	
this object can then process passed **XML** and **XPath** expressions using the **evaluate()** method.				XML, XPath		evaluate()

Research part one (Data analysis and findings)

1- Case Selection

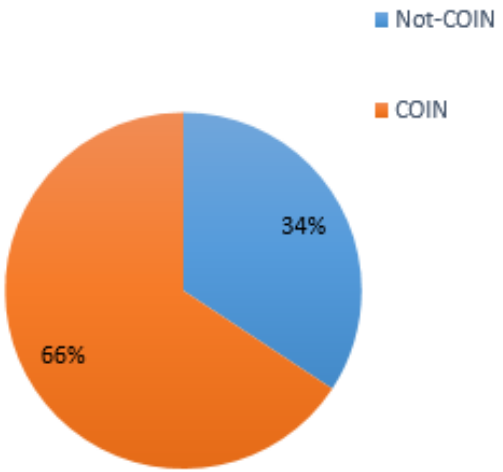
2- Case Execution.

3- Cross-Case Analysis.



COIN category	# No. of instances
Not-COIN	690
Dynamic	570
Semantic	551
Structure	107
Syntax	57
Quality	25
Context	13

Seven-COIN instances distribution



COIN category	# No. of instances
Not-COIN	690
COIN	1323

Two-COIN instances distribution

COINs distribution

Research part one (Data analysis and findings)

1- Case Selection

2- Case Execution.

3- Cross-Case Analysis.

- **RQ1:**
  - What are the observed **patterns** in specifying the conceptual interoperability constraints COINs in the NL text of API documentation?
- **Answer:**
  - Pattern Table

COIN	Pattern	Example	COINs %
Not-COIN	Technical keywords	XML, iOS, XPath, JSON, OSGi, SDK, HTTP, GET, POST, etc.	**30.7%**
Dynamic	Action Verbs	create, use, request, access, plug, lock, include, set-up, run, start ,call-up ,redirect.	**35.8%**
	Conditional statement	if , when, once, while, as long as ,unless	**24.0%**
Semantic	Output/Input verbs	return, receive, display, response, send,	**18.8%**
	Supporting verbs	support, provide, Suggest, give, propose.	**16.4%**
	Admission verbs	allow, enable, admit, grant, permit, facilitate, authorize, prevent	**13.5%**

Research part one (Thread to validity)

Generalizability

- We decided to include multiple cases (*six cases*)

Completeness

- We have selected inclusive parts of the large API documentations (e.g. in the API document of Eclipse (**651** sentences)

Researcher bias

- It was replicated by another researcher.

Research part two (Utilizing ML for Identifying The COINs)

Research Part One (Multiple-Case Study)

➔*Answering* **RQ1**

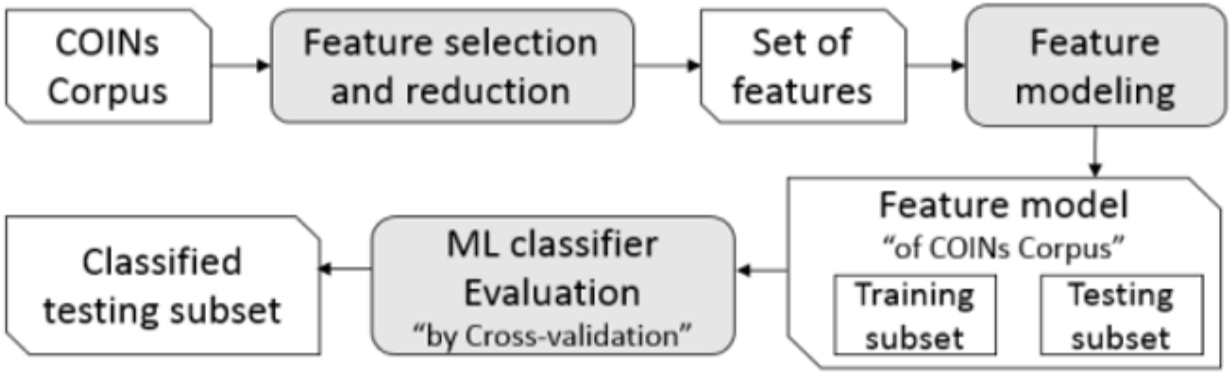
Research Part Two (Utilizing ML for Identifying The COINs) ➔*Answering* **RQ2**

Research part two (Utilizing ML for Identifying The COINs)

Feature selection (Alternatives):

1- Rule Based: using manually identified patterns

2- Bag-of-Words (BOWs) [5]: automatically



. 'Process Flow' of the classification model

BOWs [5]: is a simple technique for text classification, in this approach, each word in a sentence is considered as a feature and a document is represented as a matrix of weighted values using some kind of a weighting method such as TF-IDF (Term frequency –Inverse Document Frequency)

Research part two (Utilizing ML for Identifying The COINs)

Explored ML Classification Algorithms:

Classification Algorithm
Logistic Regression
Naïve Bayes
Complement Naive Bayes
Decision Tree (J48)
Neural Network
Random Forest Tree
KNN, k=18
Support Vector Machine

Research part two (Utilizing ML for Identifying The COINs)

- **Configuring and running tests** for the ML classification algorithms (using **Weka 3.7.13**)

- **K-fold Cross-Validation** [4] for training and testing:

  k=10, 9 for training and 1 for testing for 10 rounds.

  Take average of the 10 rounds

- Evaluate the experimental results in terms of:

  Precision

  Recall

  F-Measure

Weka 3.7.13: Weka is a collection of machine learning algorithms for data mining tasks.
URL: http://www.cs.waikato.ac.nz/ml/weka

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Research part two (Evaluation)

Answering RQ2: Effectiveness using ML for automated COINs Identification.

1- Evaluation of the first approach "**Rule-based**"

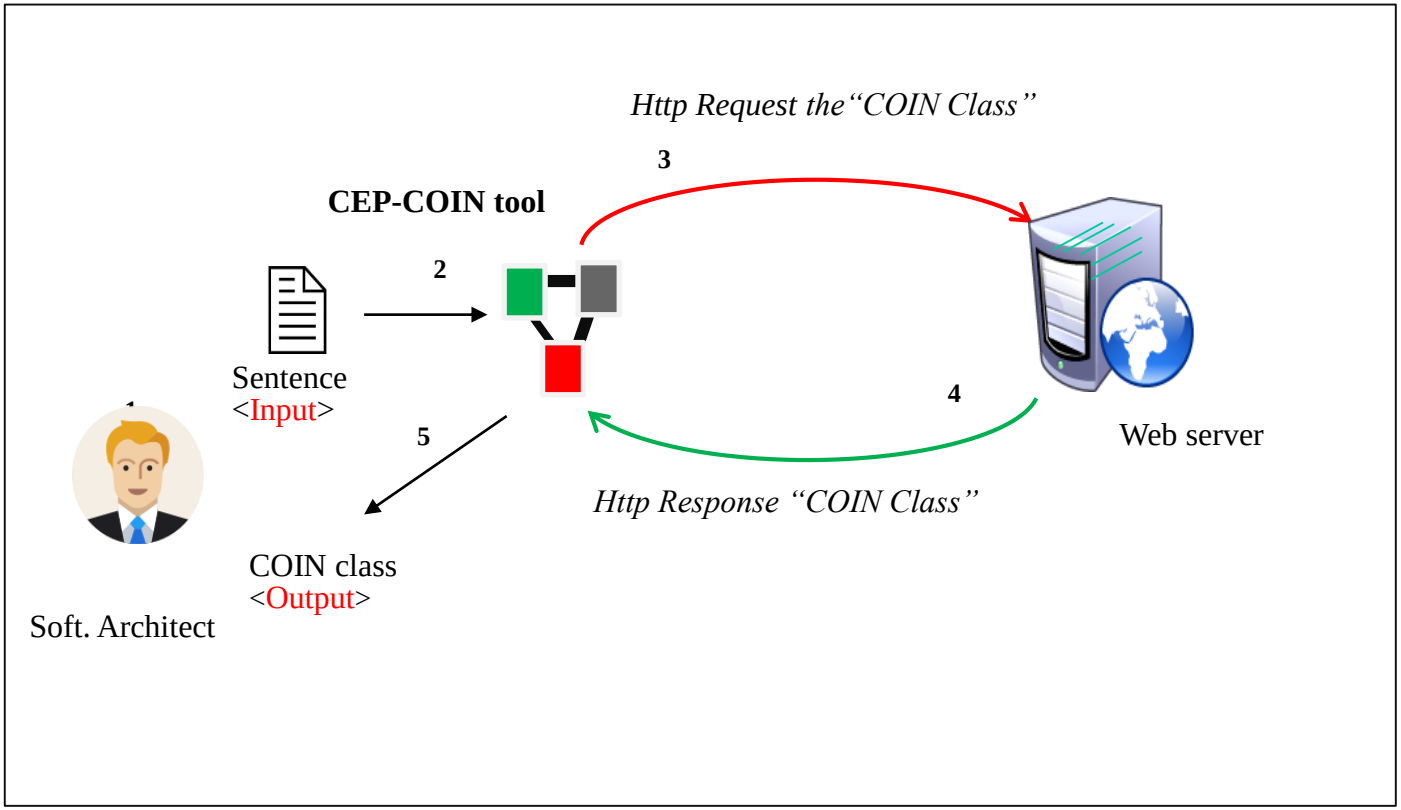
Corpus	Classification Algorithm	Recall	Precision	F-Measure
Seven-COIN	Logistic Regression	47.0%	51.7%	**47.6%**
Two-COIN	Logistic Regression	66.5%	66.1%	**65.7%**

2- Evaluation of the first approach "**BOWs"**

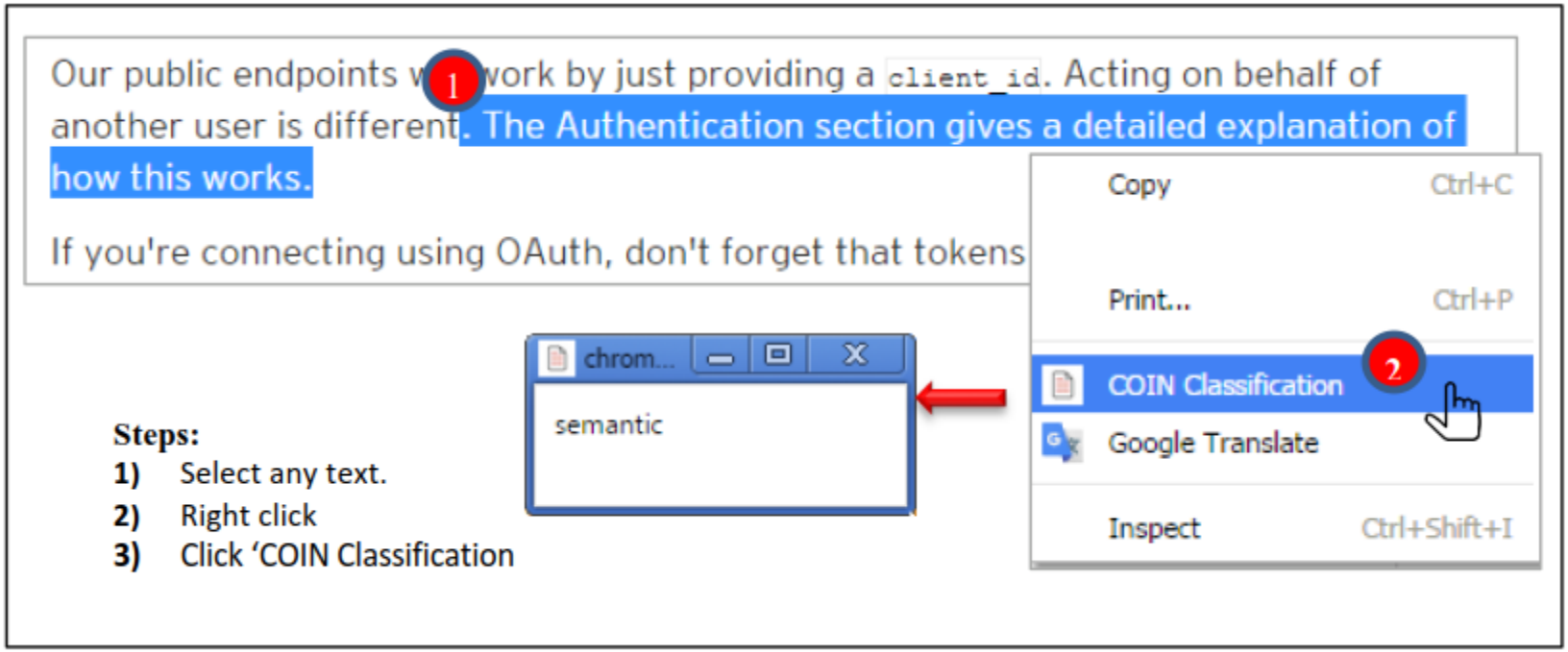
Corpus	Classification Algorithm	Recall	Precision	F-Measure
Seven-COIN	ComplementNaïveBayes	70.4%	70.2%	**70.0%**
Two-COIN	ComplementNaïveBayes	81.9%	82.0%	**81.9%**

Technical support

Classifier Ensemble Plugin–COIN (CEP-COIN)



Technical support



Practical using of the tool

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Conclusion

- Utilizing ML in COINs Identification has potentials in improving efficiency (Efforts & Time) and effectiveness (Accuracy) of Conceptual Interoperability Analysis Task

- In our findings, **BOWs** achieved the best results for building the COINs classification model
  - prototype tool built upon BOWs model

- Automatic Two-COIN Classification achieves higher accuracy compared to Seven-COIN classification

Research Paper: ***H. Abukwaik, M. Abujayyab, S. R. Humayoun and D. Rombach, "Extracting Conceptual Interoperability Constraints from API Documentation using Machine Learning," in The 38th International Conference on Software Engineering (ICSE 2016) Companion, TX,USA, 2016***

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Future work

- Analyze and classify more sentences manually for better training of the ML classification model and for better results
- Extending our tool with more features, so that it generates a report for all existing COINs in an API document
- Evaluating the Tool within industrial case studies

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

References

[1] H. Abukwaik , M. Naab and D. Rombach, "A Proactive Support for Conceptual Interoperability Analysis in Software Systems," in Software Architecture (WICSA), 2015 12th Working IEEE/IFIP Conference on, Montreal, 2015.

[2] X. Guo, Y. Yin, C. Dong, G. Yang and G. Zhou, "On the Class Imbalance Problem," in *Fourth International Conference on Natural Computation*, Jinan, 2008.

[3] R. K. Yin, Case Study Research: Design and Methods (Applied Social Research Methods), p. 46.

[4] C. Silva and B. Ribeiro, Inductive Inference for Large Scale Text Classification (Kernel Approaches and Techniques), vol. 225, Berlin: Springer-Verlag, pp. 21-24

[5] W. Chu and T. Y. Lin, Foundations and Advances in Data Mining (Studies in Fuzziness and Soft Computing), Springer; 2005 edition (October 26, 2005), pp. 225-226.

[6] H. Abukwaik, M. Abujayyab, S. R. Humayoun and D. Rombach, "Extracting Conceptual Interoperability Constraints from API Documentation using Machine Learning," in *The 38th International Conference on Software Engineering (ICSE 2016) Companion*, TX,USA, 2016.

[7] Q. Wu, L. Wu, G. Liang, Q. Wang, T. Xie and H. Mei, "**Inferring Dependency** Constraints on Parameters for Web Services," in *Proceedings of the 22Nd International Conference on World Wide Web*, Rio de Janeiro, Brazil, International World Wide Web Conferences Steering Committee, 2013, pp. 1421-1432.

[8] R. Pandita, X. Xiao, H. Zhong, T. Xie, S. Oney and A. Paradkar, "Inferring Method Specifications from Natural Language API Descriptions," in *Proceedings of the 34th International Conference on Software Engineering*, Zurich, Switzerland, IEEE Press, 2012, pp. 815-825.

[9] H. Zhong, L. Zhang, T. Xie and H. Mei, "Inferring Resource Specifications from Natural Language API Documentation," in *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering*, IEEE Computer Society, 2009, pp. 307-318.

[10] U. Dekel and J. D. Herbsleb, "Improving API Documentation Usability with Knowledge Pushing," in *Proceedings of the 31st International Conference on Software Engineering*, IEEE Computer Society, 2009, pp. 320-330.

[11] S. Forrest , S. Janice and S. I. K. Dag, Guide to Advanced Empirical Software Engineering, Springer; 2008 edition (October 26, 2007), 2007.

Thank you

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Research part one (Study Execution)

1- Case Selection

2- Case Execution.

3- Cross-Case Analysis.

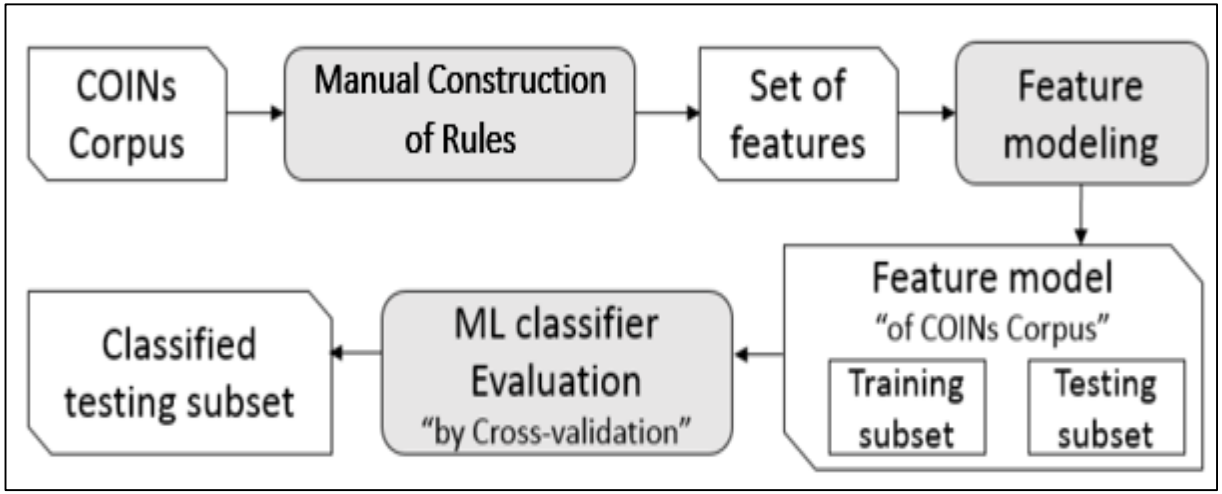
1- Data Preparation → 2- Data Collection → 3- Data Analysis

Cleaning data

- Extraction sheet
- Fields ,,,,

- Manual Classification
- Pattern Identification

1- First Approach: Rule-based Machine Learning Classification



'Process Flow' of the first machine learning classification approach

Backup

1- First Approach

2- Second Approach

#	Rule name	Example of sentences satisfying the rule
1	Definition	oEmbed **is an** open standard to easily embed content from oEmbed providers into your site
2	Goal	Background actions launch the containing iOS app in the background **so that** it can process the action
3	Conditional	**if** a command name is specified, the help message for this command is displayed.
4	Explanation/Example	**for example,** you can use this to protect against CSRF issues.
5	Method Call	in order to embed a player widget using JavaScript SDK, you can call **SC.oEmbed()** function
6	Modal Verb	**you can** also get a list of comments for a specified sound
7	Resource	artifacts for each tool, such as **files** data, are coordinated by a common platform resource model.
8	Structure	Fundamentally, a bundle is just a **collection** of files (resources code) installed in platform
9	Technical Term	instead, create a complementary experience to **iOS** app.
10	Variable	Some API only require use of a **client_id.**
11	Warning	do not assume **access_token** is valid forever.
12	Output/Input	on success, function **returns** true.
13	Action Verb	to perform a task, a plug-in **creates** a job then **schedules** it.

NLP + Text Processing

Backup

Exploratory Experiment

1- First Approach

2- Second Approach

Phase 1: Preparing the training data set

Sentence	Definition	Goal	Conditional Statement	Explanation/ Example	Method Call	Modal Verb	Resource	Structure
a user's availability can be free and their activity can	0	1	0	0	0	0	0	0
Because this authorization flow depends on passing a set of user credentials, it	0	1	0	0	1	0	2	0
Depending on your needs, you can embed a player widget, use the JavaScript	0	1	0	0	0	0	2	2
for a full list of properties that can be set on a sound	0	0	0	1	0	0	3	1
Given a sound or set URL, you can retrieve all of the	0	1	0	0	0	1	3	1
if you are writing an application that runs in the browser, you should	0	0	1	0	2	1	1	1
if you donot want to use the SoundCloud widget, our API gives you the	0	0	1	0	0	0	5	0

Snapshot of an excerpt of the rule matrix

Research part two (Evaluation)

1- First Approach

2- Second Approach

$$Recall\ (R) = \frac{\#\ of\ correct\ prediction}{\#\ of\ relevant\ records} = \frac{\#TP}{\#TP + \#\ FN}$$

$$Precision\ (P) = \frac{\#\ of\ correct\ prediction}{\#\ of\ (relevant + irrelative)\ records} = \frac{\#TP}{\#TP + \#FP}$$

$$F = \frac{2 \times precision\ \times recall}{precision + recall}$$

F-Measure (F) is a combination of recall and precision. F-Measure is a **popular evaluation metric** for imbalance problem, in which the data set are not classified equally (e.g., some of the classes are more than the others)

Contribution

Research Contributions 1/2:

- **Building the COINs corpus** (i.e. ground truth): Manually classifying the conceptual interoperability constraints (COINs) of the collected data with the help of “Constraints of COIN Model”. [1]

- **Defining representation patterns of COINs**: Manually mining and analyzing of the textual content of the API documents in order to identify the **frequently used** terms and sentences structures from the collected API documents.

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Contribution

- **Research Contributions 2/2:**

- **Building the text classification model (classifier):** Utilizing the obtained corpus, we designed two different classifiers. These classifiers are used for automatically classifying the COINs.

- **Exploratory experiment**: Evaluating the efficiency of the created classifiers in terms of accuracy by conducting experiments that utilize **different text classification algorithms**.

- **Developing a plugin prototype**, which is available to be used as web service.

- *Research Paper:* *H. Abukwaik, M. Abujayyab, S. R. Humayoun and D. Rombach, "Extracting Conceptual Interoperability Constraints from API Documentation using Machine Learning," in The 38th International Conference on Software Engineering (ICSE 2016) Companion, TX,USA, 2016*

Limitation

- Not tested using an industrial experiments
- Restricted only to Six-Cases
- Used to find the COIN of single sentence (instead of the whole document)

Related works

Wu et al. [7] identified parameters' dependency constraints from multiple library resources. (NLP)

Pandita et al. [8] proposed an approach to automatically infer the formal **method specifications** from natural text of API documents. (NLP)

Zhong et al. [9] proposed an approach called Doc2Spec to recognize and **infer resource specifications**. (ML)

Dekel and Herbsleb [10] introduced an approach for improving API documentation usability by extracting and **highlighting the important part of documentation**, which includes the sensitive information, instructions, and guidelines to push them into a programming IDE editor. [**eMoose tool**]

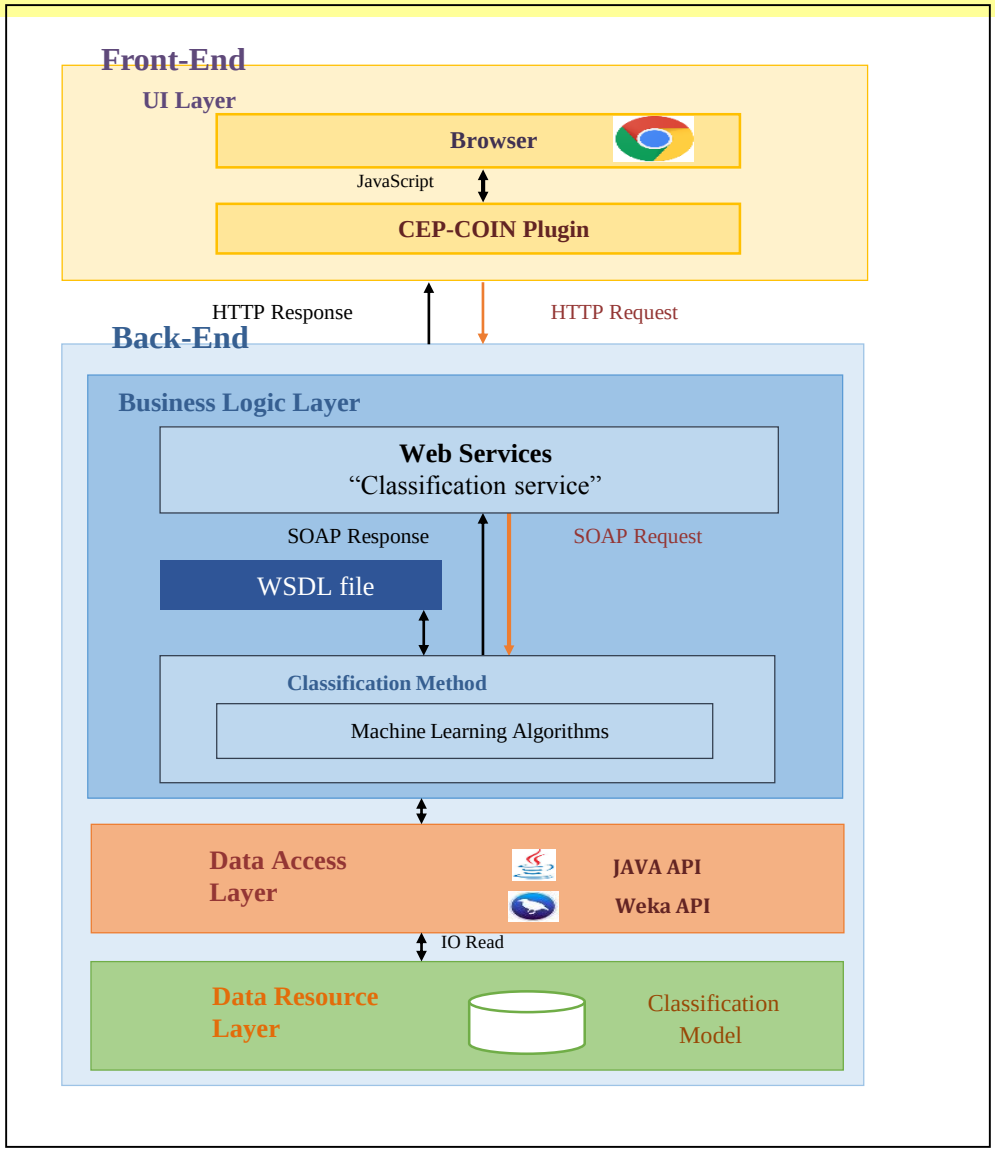
TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

Related works

In Our research:

- We elaborate on Abukwaik et al [6] idea of extracting different type of conceptual constraints utilizing (NLP and ML).
- We extracted different types of constraints as mentioned before (**Not-COIN, Dynamic, Semantic, Syntax, Structure, Context and Quality**) constraints.
- We followed the Empirical Methodology [47].

Contribution



Architecture of the CEP-COIN

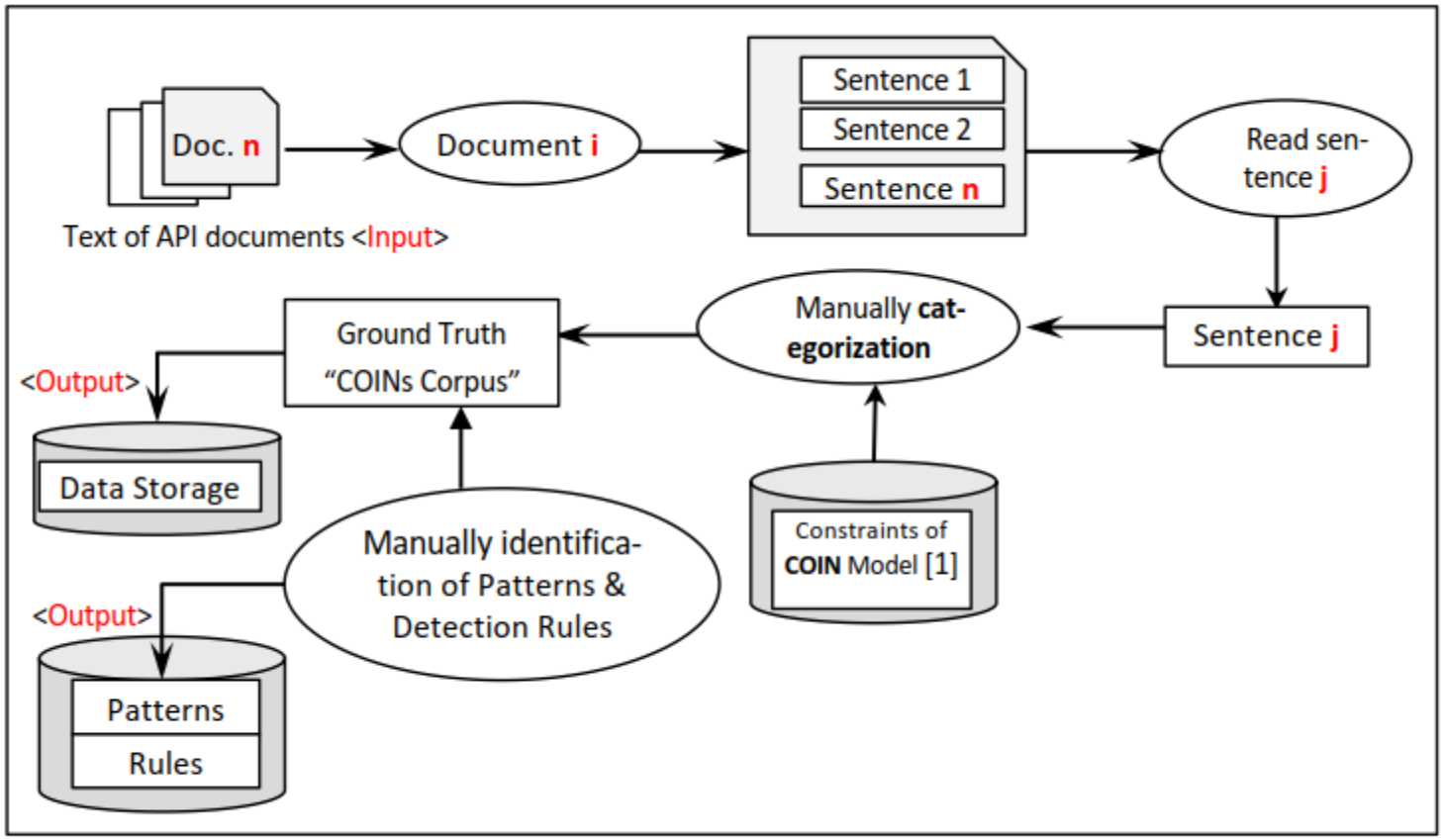
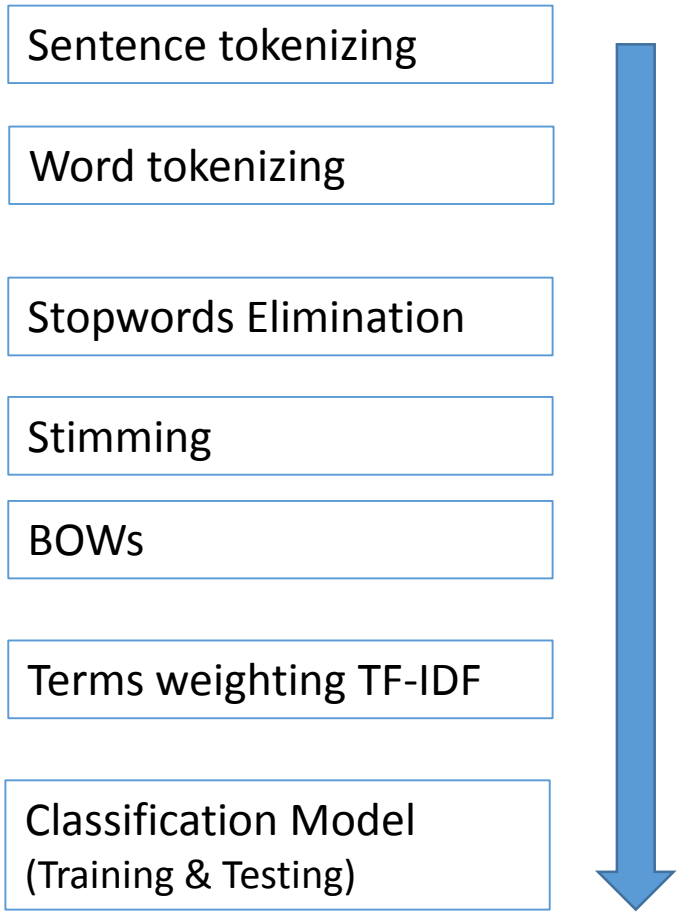


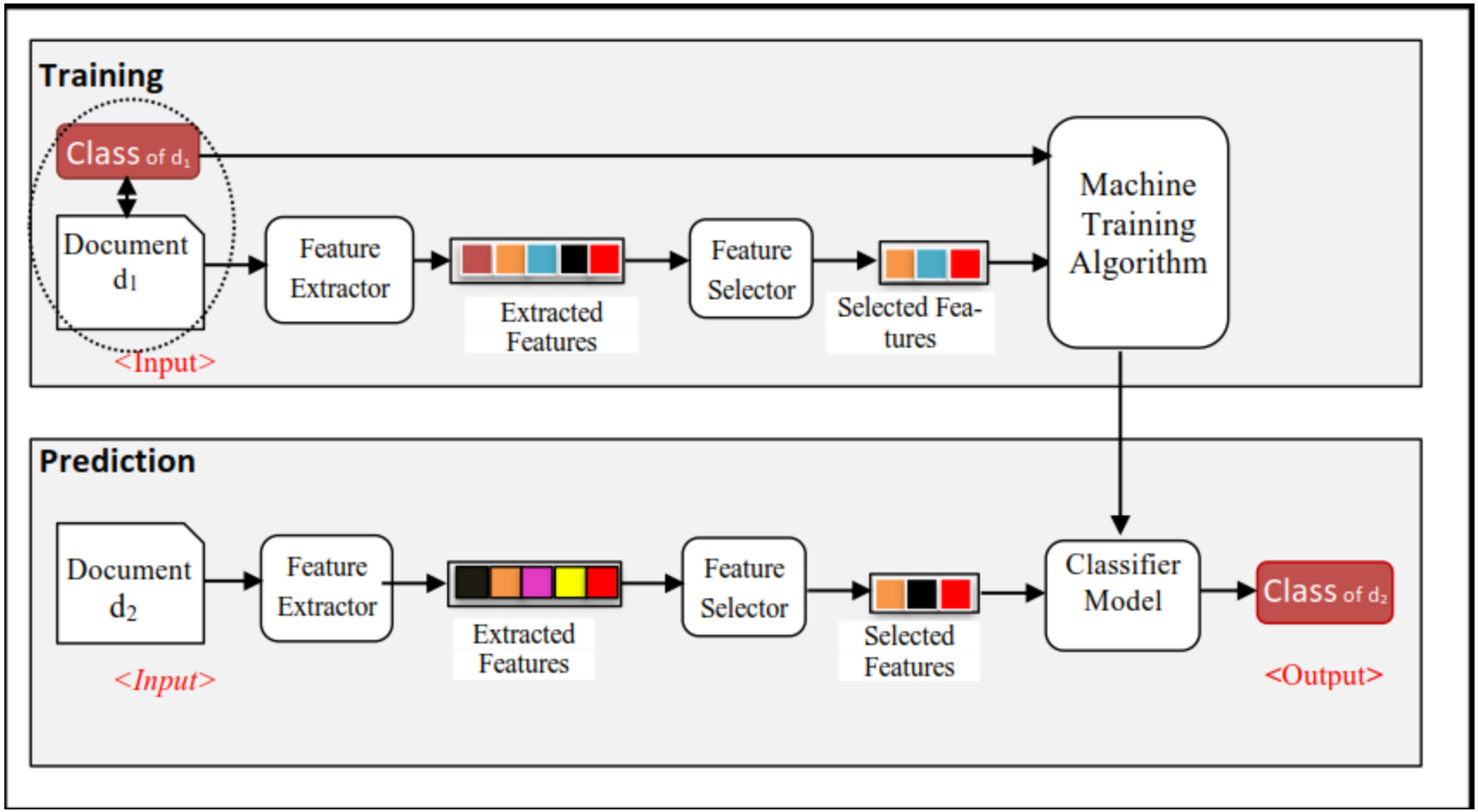
Fig. 9. Content analysis method 'process flow'

Algorithm 1: Create TwoCOINs Corpus

Data: Seven-COINs corpus
Result: Two-COINs corpus

```
1  SixClasses [ ] ={'semantic','dynamic','structure','syntax','context','quality'}
2  Corpus ← ReadCorpus()
3  TwoCOINsCorpus ← empty
4  foreach sentence ∈ Corpus do
5      temp=sentence.class
6      if ( temp ∈ SixClasses ) then
7          sentence.class='COIN'
8      //update the class of the current COIN
9      TwoCOINsCorpus.add(sentence)
10 return TwoCOINsCorpus
```


Sentence tokenizing
Word tokenizing
Stopwords Elimination
Stimming
BOWs
Terms weighting TF-IDF
Classification Model
(Training & Testing)



Text Classification Workflow