Machine Learning for Annotating Semantic Web Services

Andreas Heß, Nicholas Kushmerick

University College Dublin, Ireland

{andreas.hess, nick}@ucd.ie

US Office of Naval Research

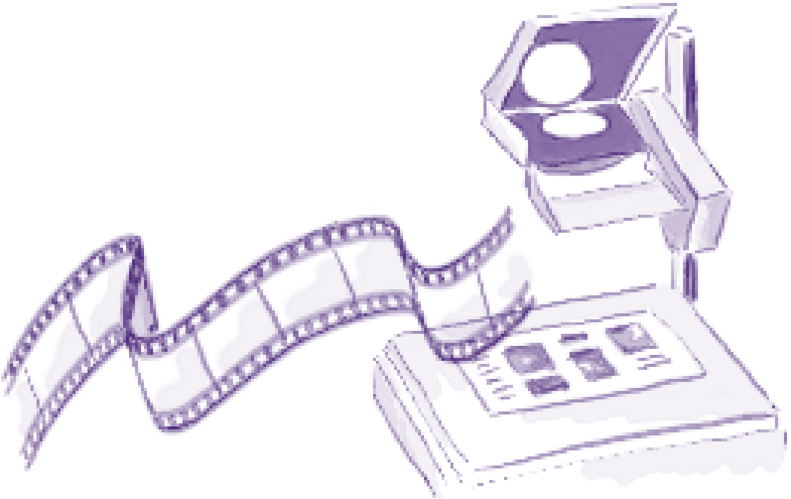
Science Foundation Ireland

1. Introduction

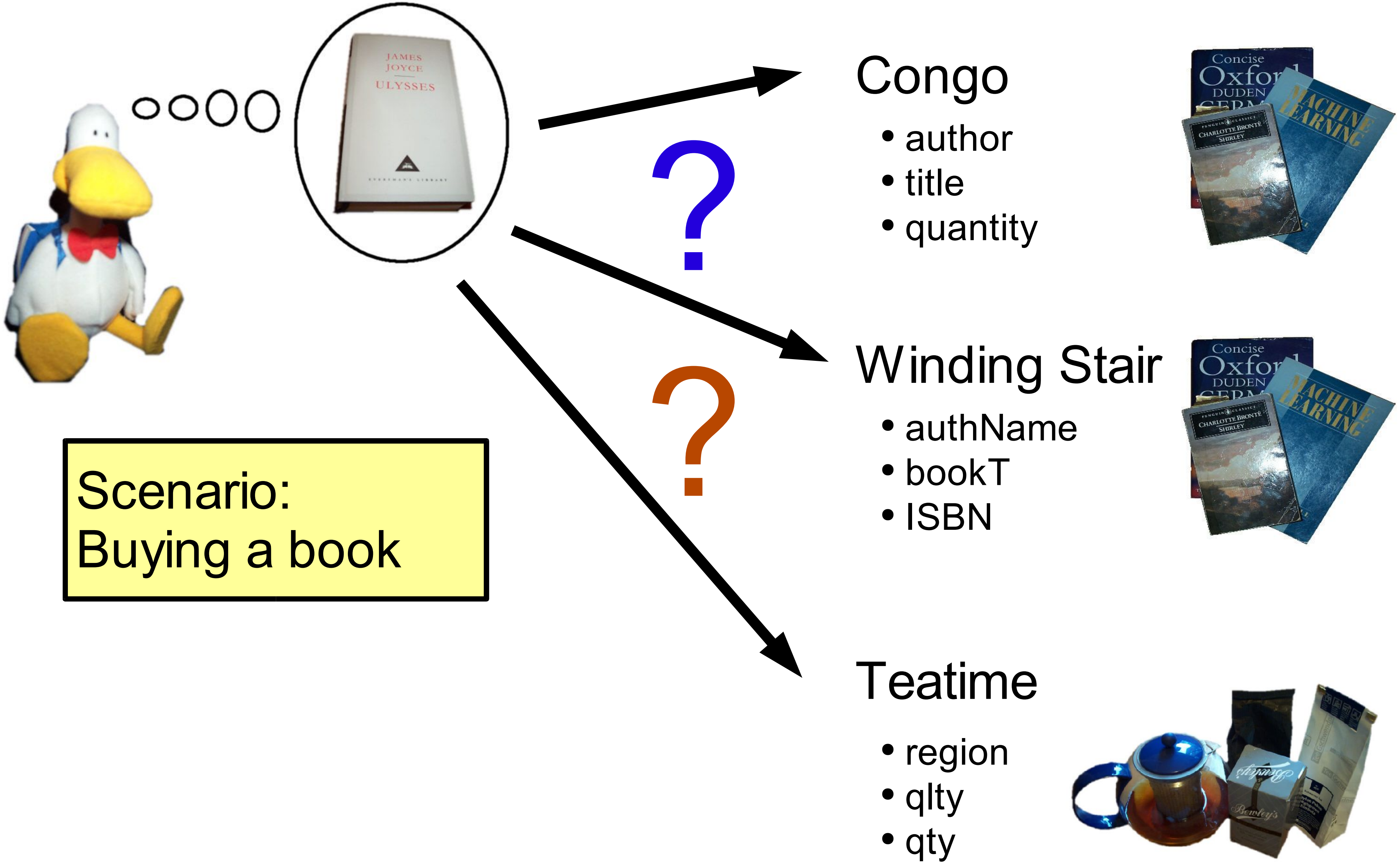
2. Our Machine Learning Approach

3. Machine Learning Assisted Annotation

4. Conclusion & Discussion



Scenario



Congo

- author
- title
- quantity

Winding Stair

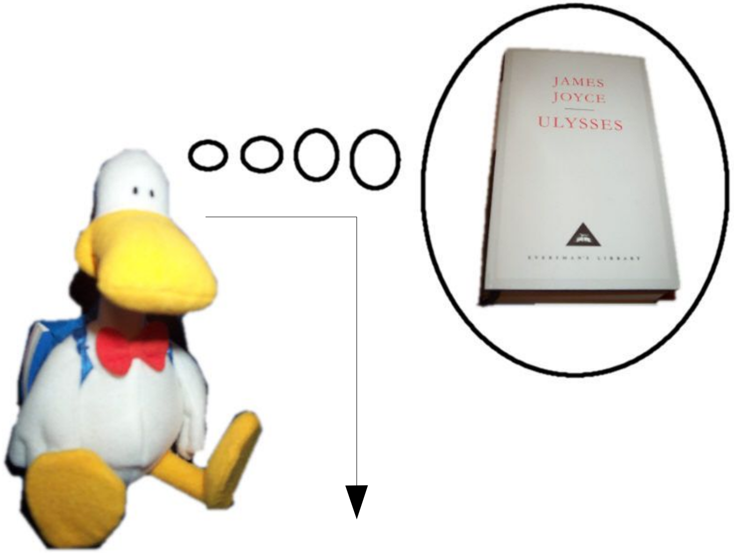
- authName
- bookT
- ISBN

Teatime

- region
- qlty
- qty

Scenario:
Buying a book

Scenario



Global Ontology

- Item
  - Quantity
  - Price
  - Book
    - Author
    - Title
    - ISBN
  - Tea
    - Region
    - Quality

Congo

- author
- title
- quantity

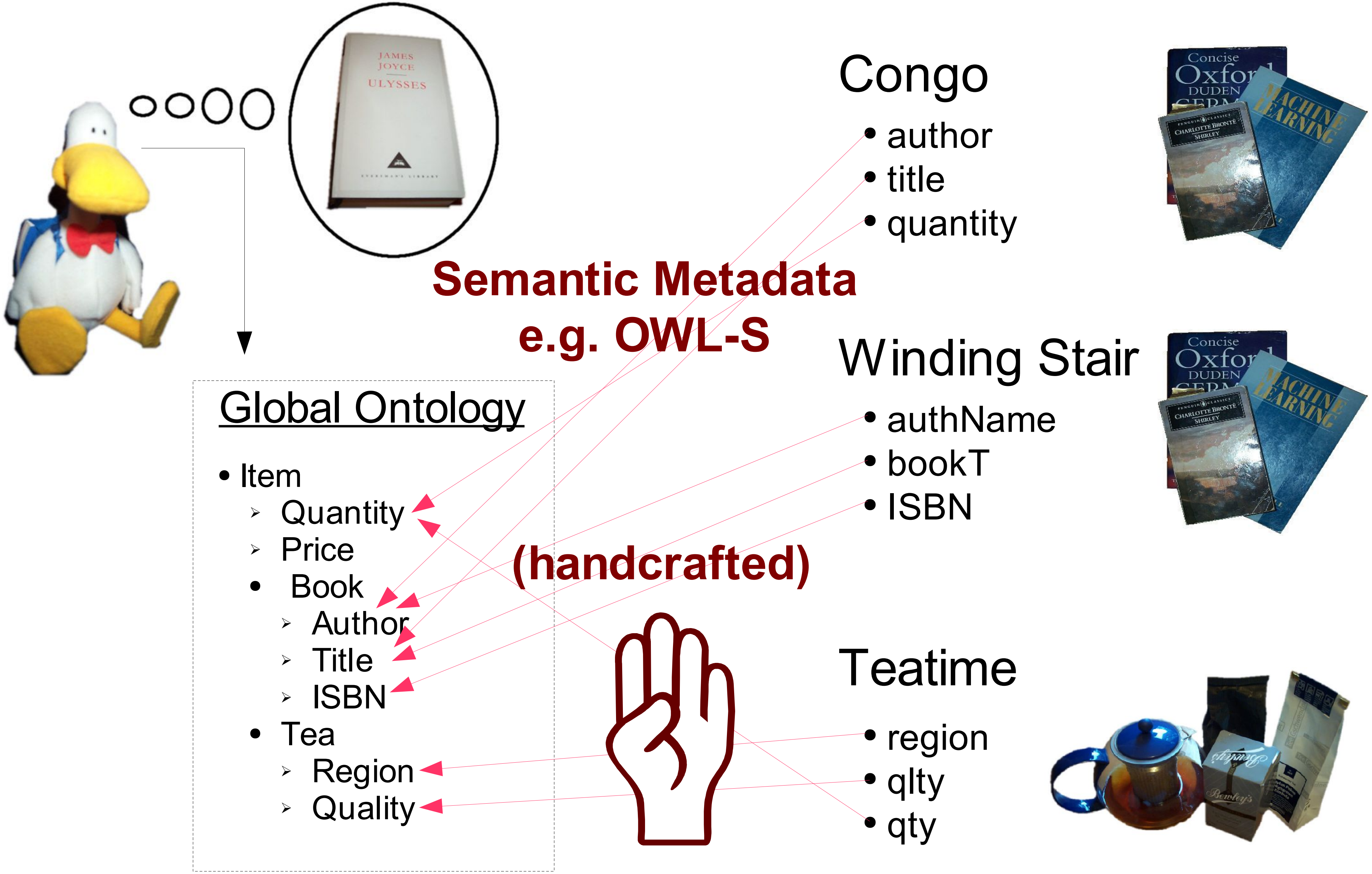
Winding Stair

- authName
- bookT
- ISBN

Teatime

- region
- qlty
- qty

Scenario



Global Ontology

- Item
  - Quantity
  - Price
  - Book
    - Author
    - Title
    - ISBN
  - Tea
    - Region
    - Quality

Semantic Metadata
e.g. OWL-S

(handcrafted)

Congo

- author
- title
- quantity

Winding Stair

- authName
- bookT
- ISBN

Teatime

- region
- qlty
- qty

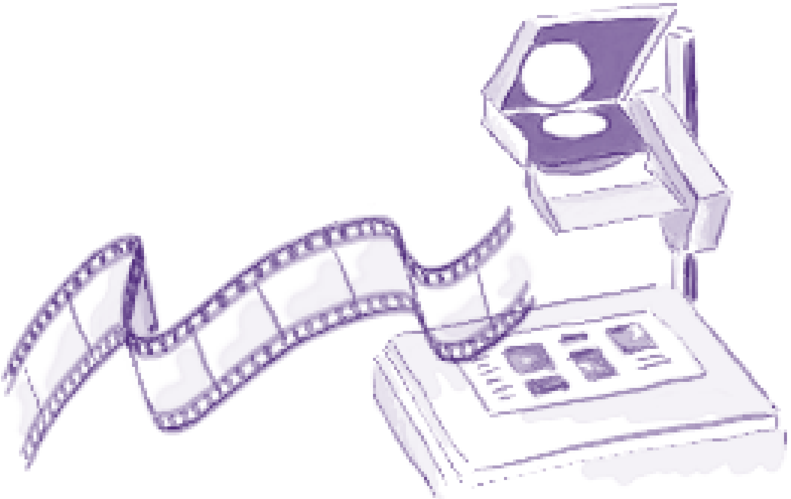
- Assumes:
  - semantic annotation
  - a shared ontology
- Semantic metadata needs to be handcrafted!!
- Our contribution: Use machine learning!

✔ Introduction

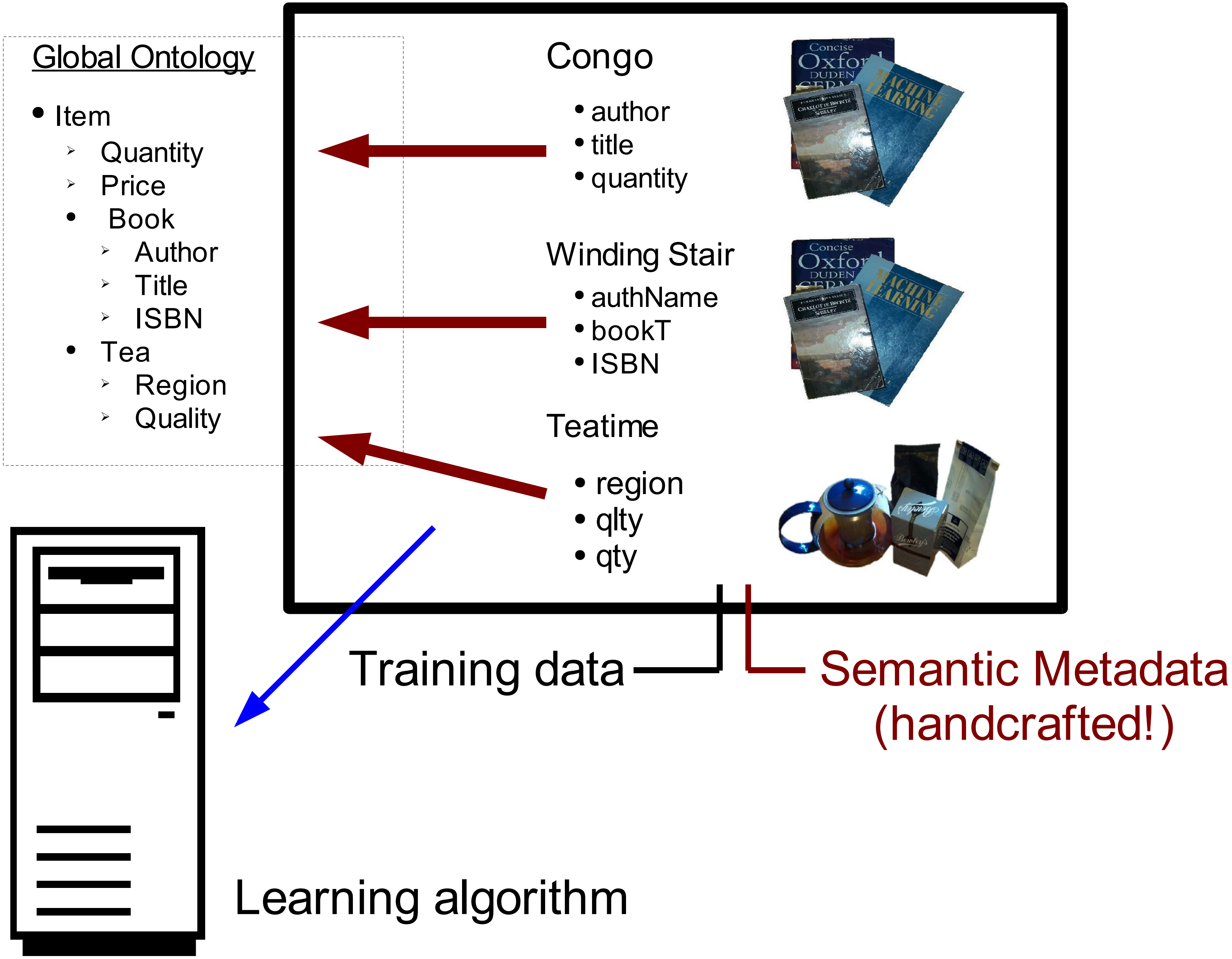
2. <u>Our Machine Learning Approach</u>

3. Machine Learning Assisted Annotation

4. Conclusion & Discussion



Machine Learning



Global Ontology

- Item
  - Quantity
  - Price
  - Book
    - Author
    - Title
    - ISBN
  - Tea
    - Region
    - Quality

Congo

- author
- title
- quantity

Winding Stair

- authName
- bookT
- ISBN

Teatime

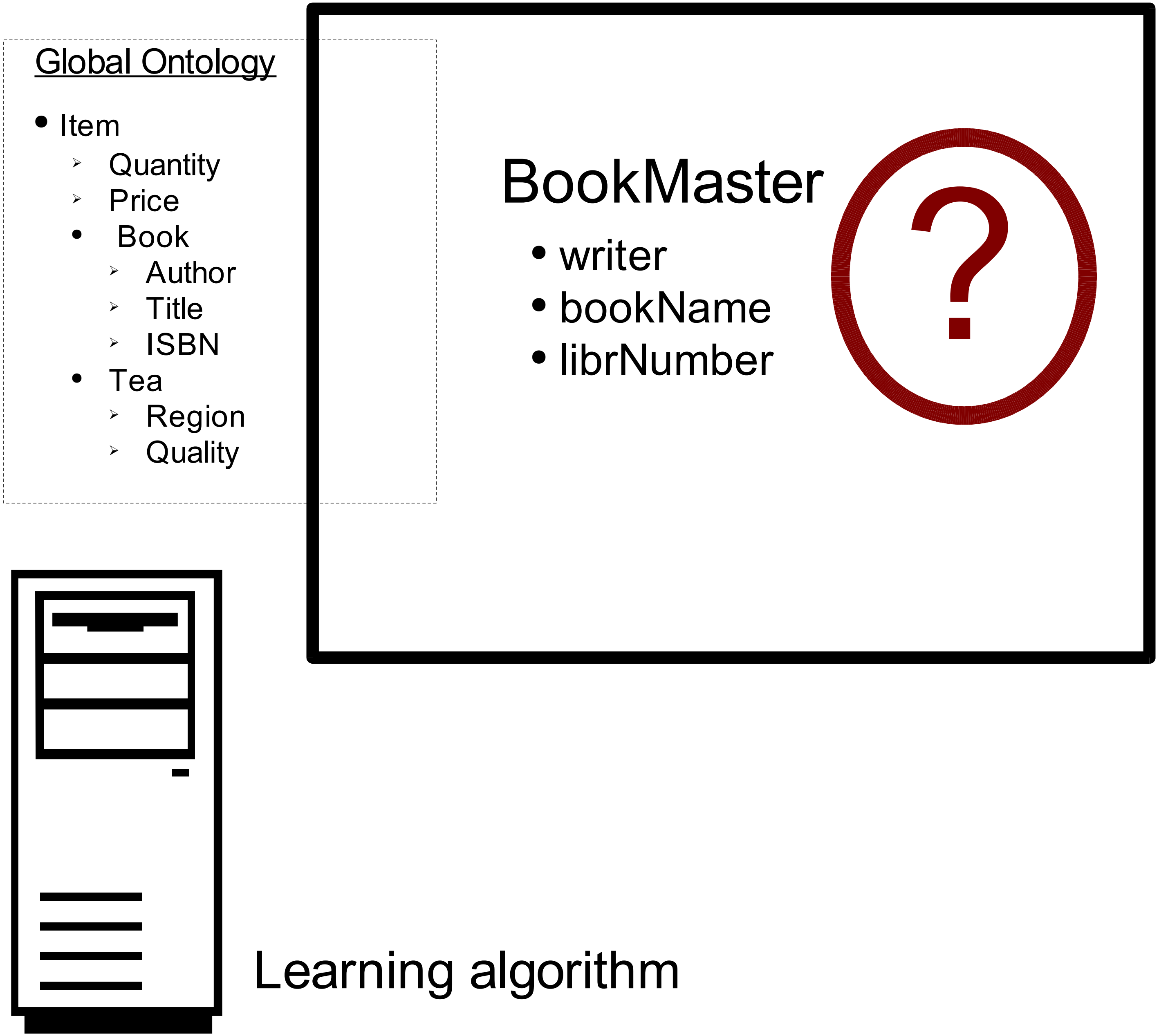
- region
- qlty
- qty

Training data

Semantic Metadata (handcrafted!)

Learning algorithm

Machine Learning



Global Ontology

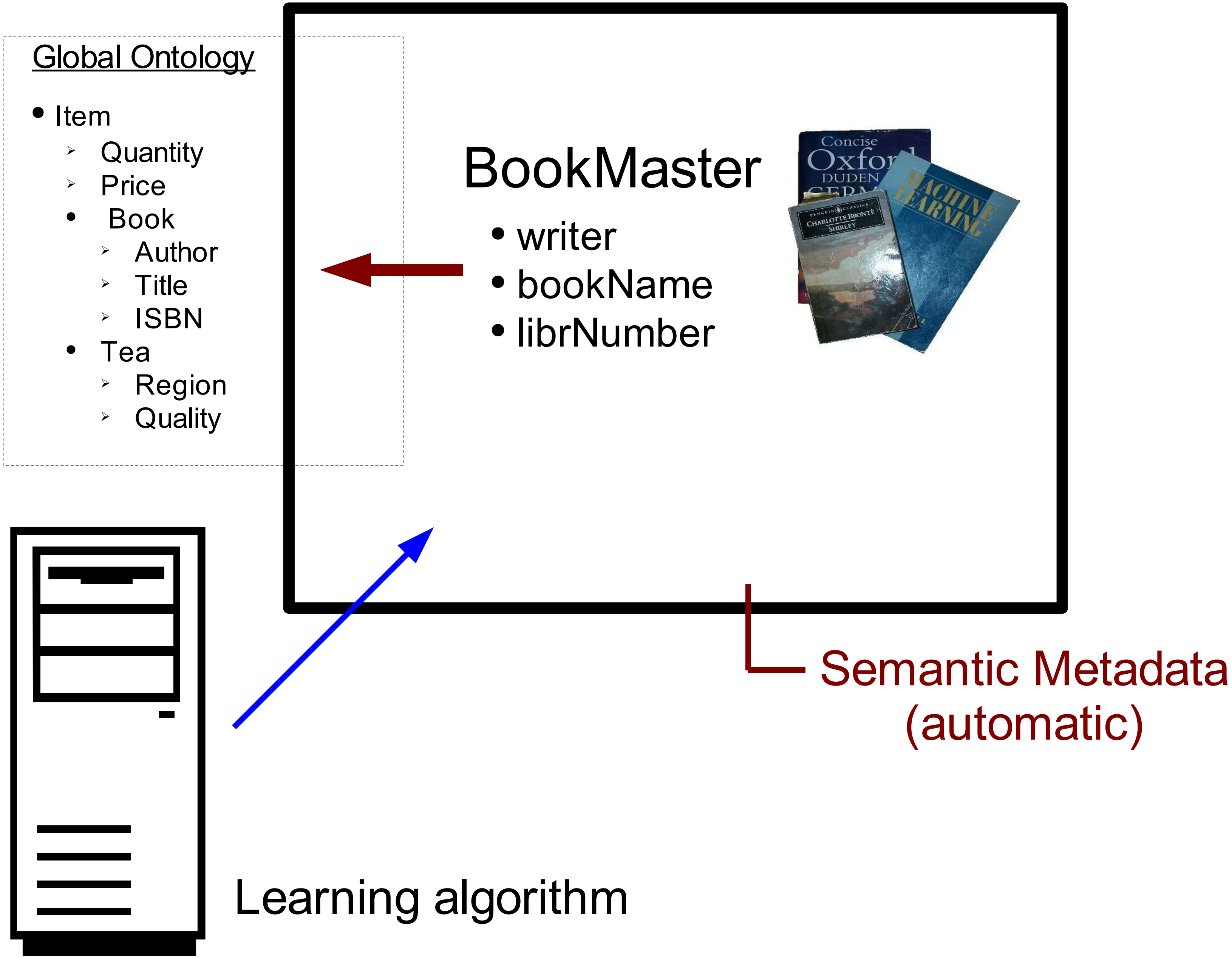
- Item
  - ➢ Quantity
  - ➢ Price
  - Book
    - ➢ Author
    - ➢ Title
    - ➢ ISBN
  - Tea
    - ➢ Region
    - ➢ Quality

BookMaster

- writer
- bookName
- librNumber

Learning algorithm

Machine Learning



Global Ontology

- Item
  - Quantity
  - Price
  - Book
    - Author
    - Title
    - ISBN
  - Tea
    - Region
    - Quality

BookMaster

- writer
- bookName
- librNumber

Semantic Metadata (automatic)

Learning algorithm

Key Assumption

What does this function do?

```
public int nbgfuibhuf(int nvzfdubzuf, int cnuzdc) {

    int vfddf = 0;
    for (int ujz = 0; ujz < nvzfdubzuf; ujz++) {
       vfddf += cnuzdc;
    }
    return vfddf;
}
```

Key Assumption

What does this function do?

```
public int multiply(int factor1, int factor2) {

    int product = 0;
    for (int n = 0; n < factor1; n++) {
       product += factor2;
    }
    return product;
}
```

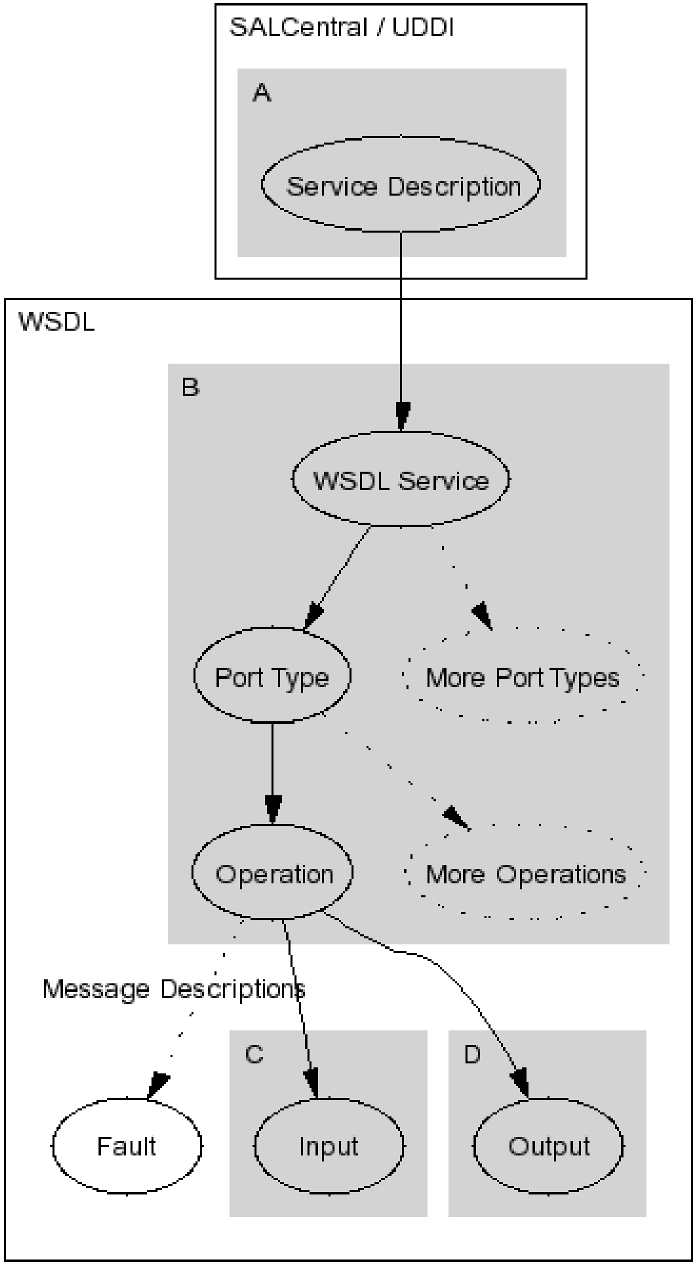
Key Assumption

What does this function do?

```
/**
 * This function multiplies two numbers in a very
 * inefficent way. It serves only as an example.
 */
public int multiply(int factor1, int factor2) {

    int product = 0;
    for (int n = 0; n < factor1; n++) {
       product += factor2;
    }
    return product;
}
```

A Text Classification Problem



Web Service classification

==

Text classification

Definitions

Category

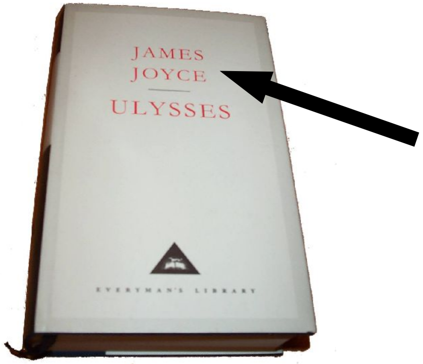
- Broad description of service as a whole
- e.g. e-commerce, weather, finance
- Profile hierarchy

Domain

- Purpose of single operation
- e.g. query price, purchase book
- Atomic process

Datatype

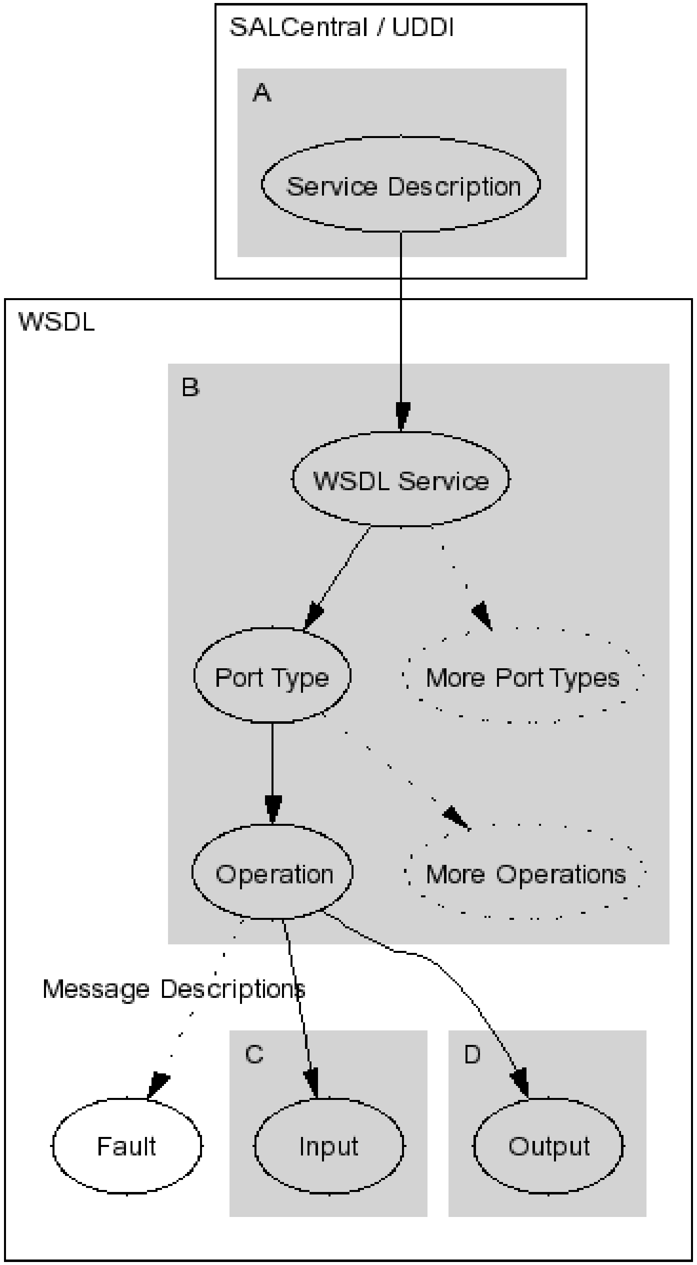
- Meaning of single parameter
- e.g. author name, credit card number
- Property



Definitions

- Category, Domain, Datatype:
  - We do not advocate a new ontology language
  - Machine learning ideas independent of actual syntax

Text Sources



- Text sources:

  A) Service Description

  (plain text, e.g. from UDDI)

  B) WSDL:

  service, portType, operation

  C) WSDL: Input message

  D) WSDL: Output message

Ensemble Learning

- Ensemble Learning
  - Each text source contains different words (e.g. operation “buyBook”, message part “author”)
  - Using seperate learners is more accurate

Evaluation

Dataset 1

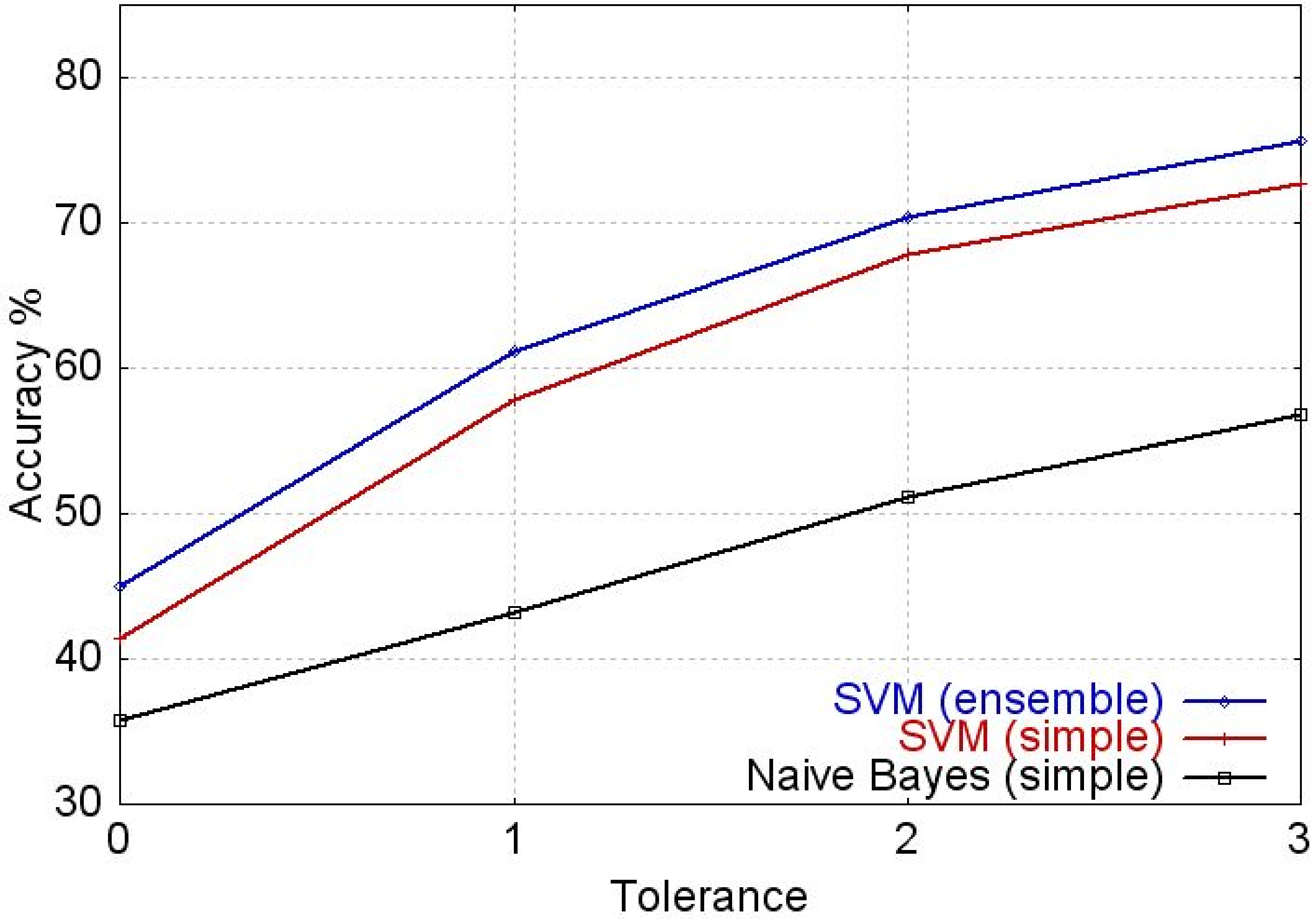
391 categorized Web Services

11 classes

highly skewed, noisy

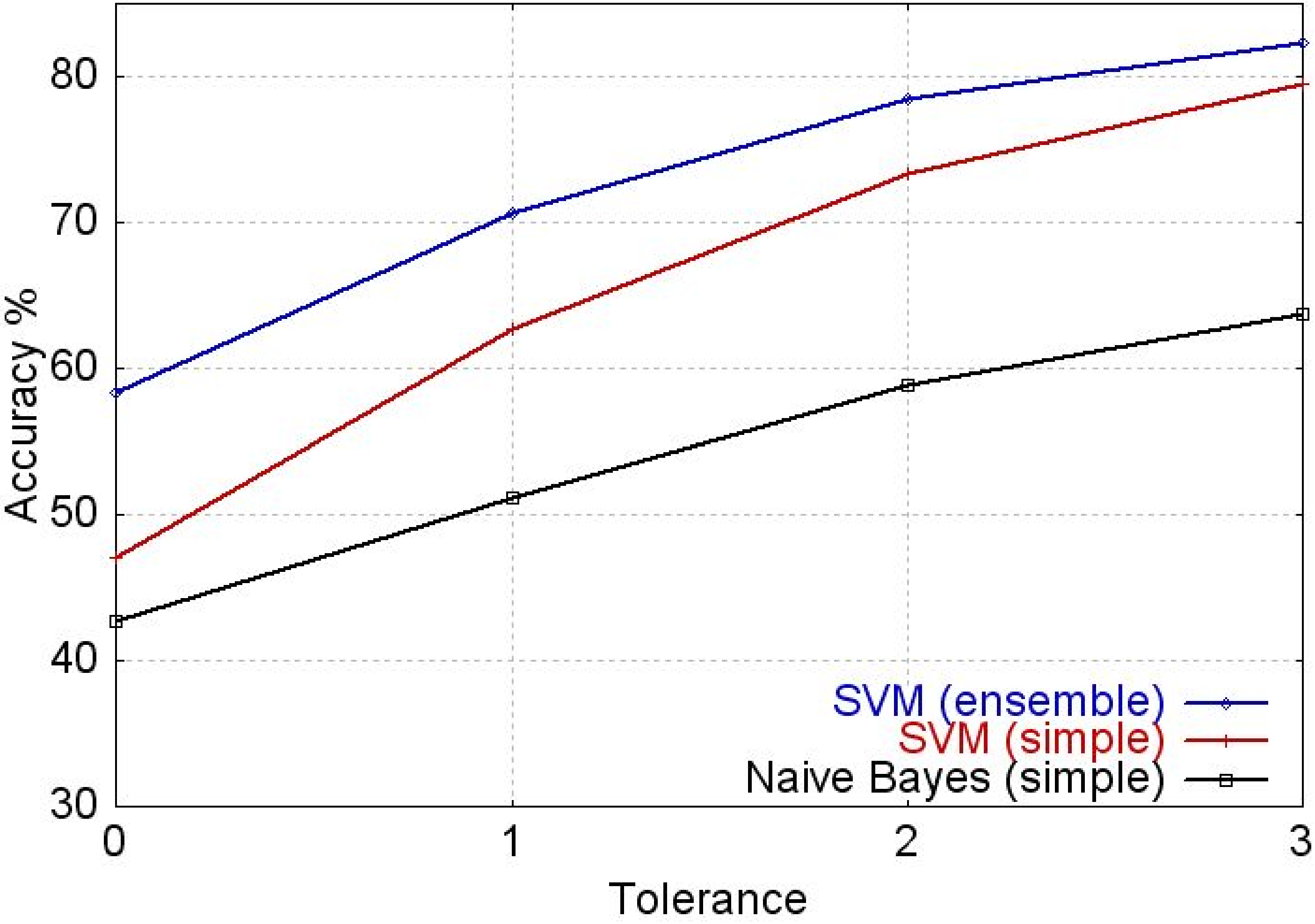
Evaluation

Classifying Category using WSDL only



Evaluation

Classifying category using WSDL plus plain text descriptions (easier)



Improvement

- Improve these results?

➔ Exploit dependencies!

Assumption

Dependencies between Category⇔Domain⇔Datatype

Category

Domain

Datatype

Books

Query book price

Book title

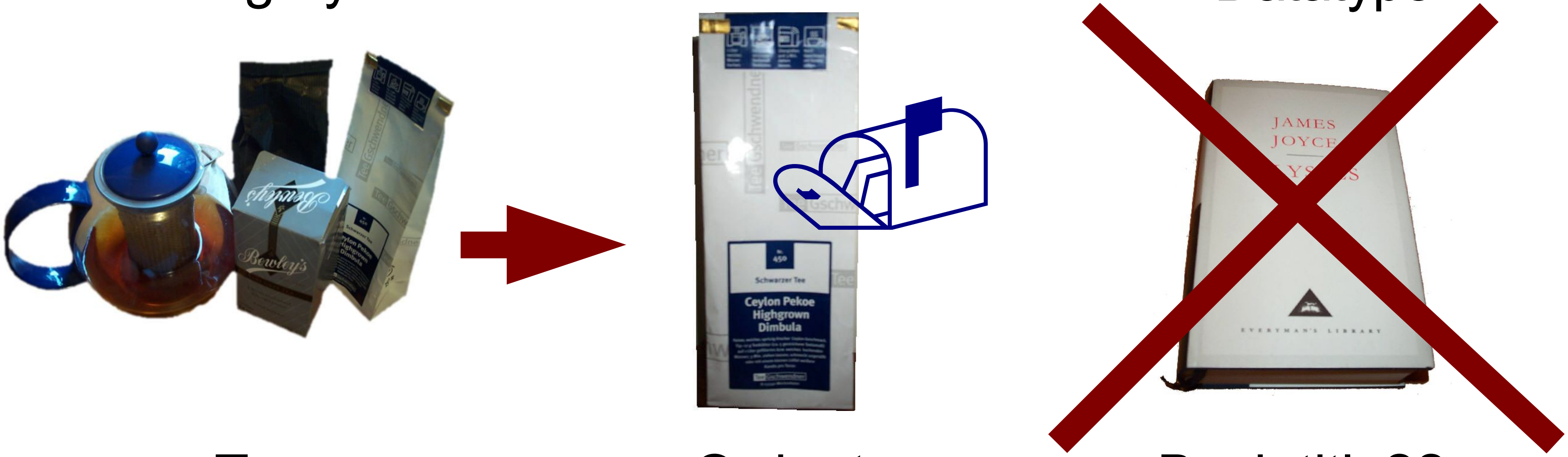
Assumption

Dependencies between Category⇔Domain⇔Datatype

Category

Domain

Datatype



Tea

Order tea

Book title??

Assumption

Dependencies between Category⇔Domain⇔Datatype

Category	Domain	Datatype
Tea	Order tea	Credit card number

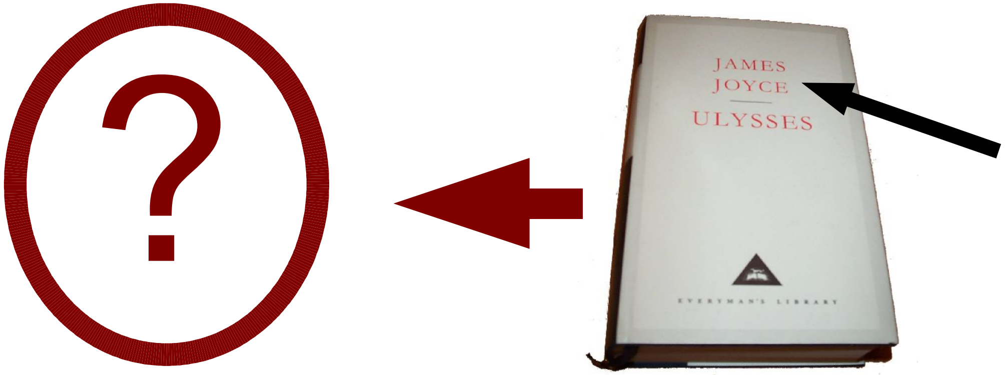
Flow of evidence

Dependencies between Category⇔Domain⇔Datatype

Category

Domain

Datatype



Book title

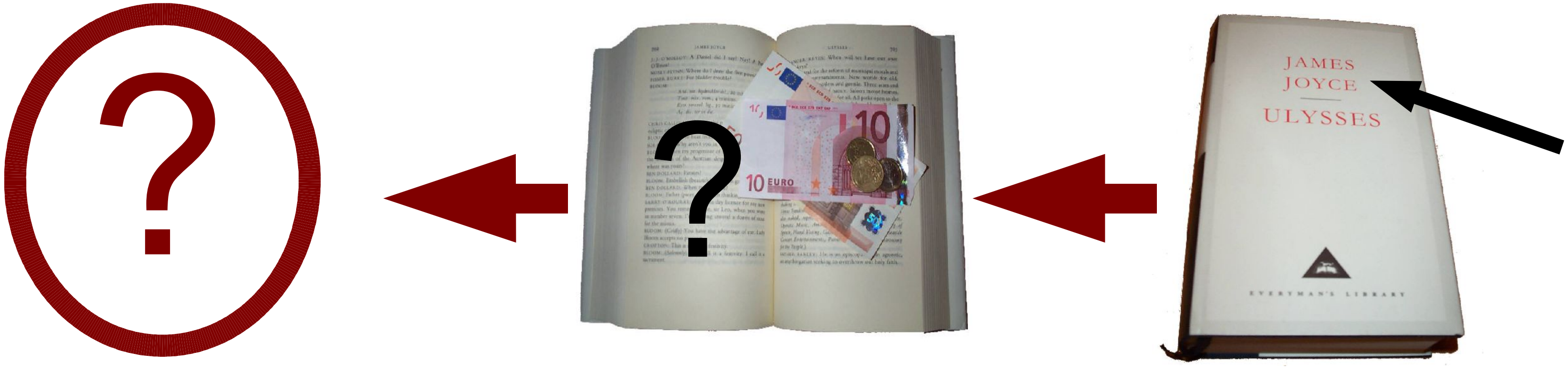
Flow of evidence

Dependencies between Category⬄Domain⬄Datatype

Category

Domain

Datatype

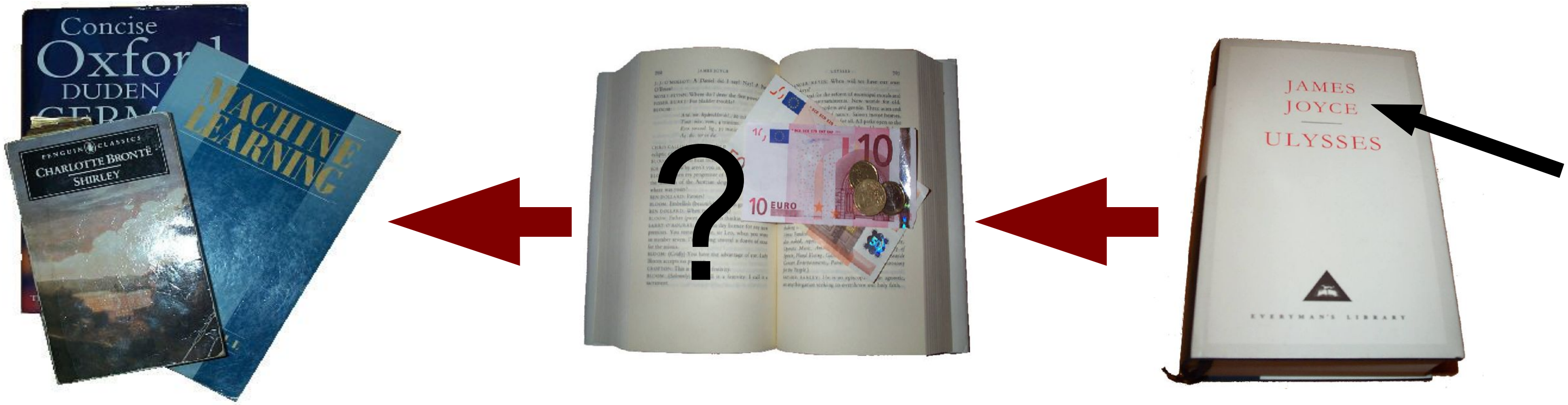


Query book price

Book title

Flow of evidence

Dependencies between Category⇔Domain⇔Datatype



Category

Domain

Datatype

Books

Query book price

Book title

Dependencies between Category, Domain, Datatype

Exploit dependencies:

➔ Iterative classification

Current research

➔ Bayesian Networks

Iterative Classification

- Classification in round N influences classification in round N+1

Category

Domain

Datatype

Round 0

Iterative Classification

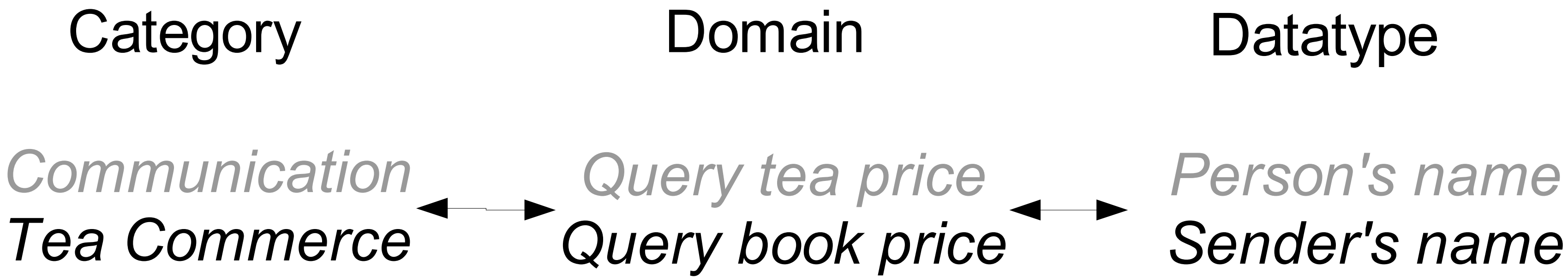
- Classification in round N influences classification in round N+1

Category	Domain	Datatype
Communication	*Query tea price*	*Person's name*

Round 1

Iterative Classification

- Classification in round N influences classification in round N+1



Category

Domain

Datatype

Communication
Tea Commerce

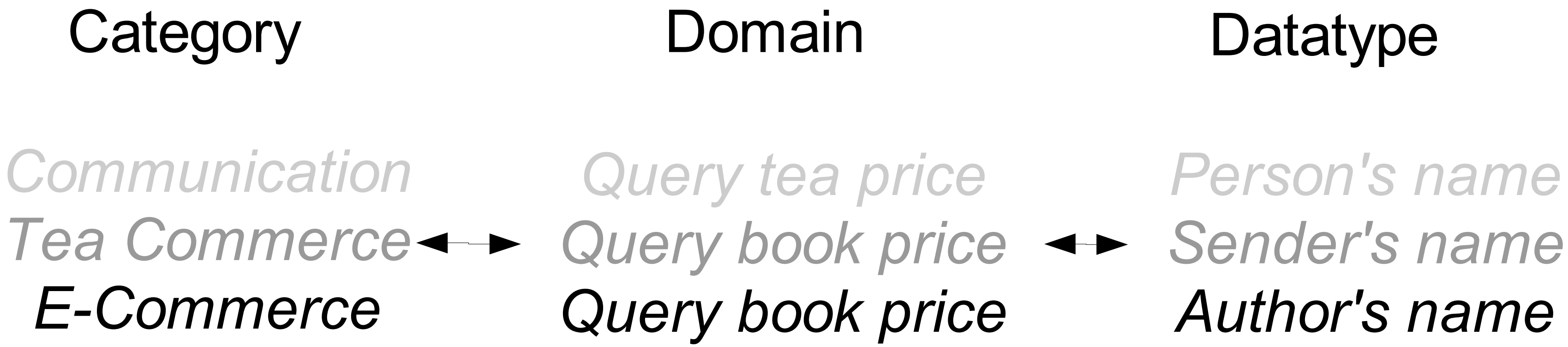
Query tea price
Query book price

Person's name
Sender's name

Round 2

Iterative Classification

- Classification in round N influences classification in round N+1



Category

Communication
Tea Commerce
E-Commerce

Domain

Query tea price
Query book price
Query book price

Datatype

Person's name
Sender's name
Author's name

Round 3

Iterative Classification

- Classification in round N influences classification in round N+1

Category	Domain	Datatype
Communication	*Query tea price*	*Person's name*
Tea Commerce	*Query book price*	*Sender's name*
E-Commerce	*Query book price*	*Author's name*
Books	*Query book price*	*Author's name*

Round 4

Iterative Classification

- Classification in round N influences classification in round N+1

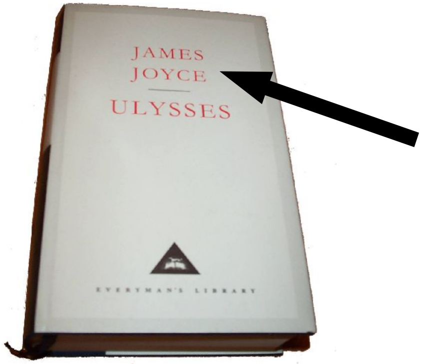
Category

Books

Domain

Query book price

Datatype



Author's name

Round 5

Evaluation

- Dataset 2: Fully annotated, available in OWL-S!
- Improvement over baseline :-)
- Not as good as preliminary results suggested :-(
- Good enough for assisted annotation :-)
- Work in progress, detailed results to come

Dependencies between Category, Domain, Datatype

Exploit dependencies:

✔ Iterative classification

➔ Bayesian Networks

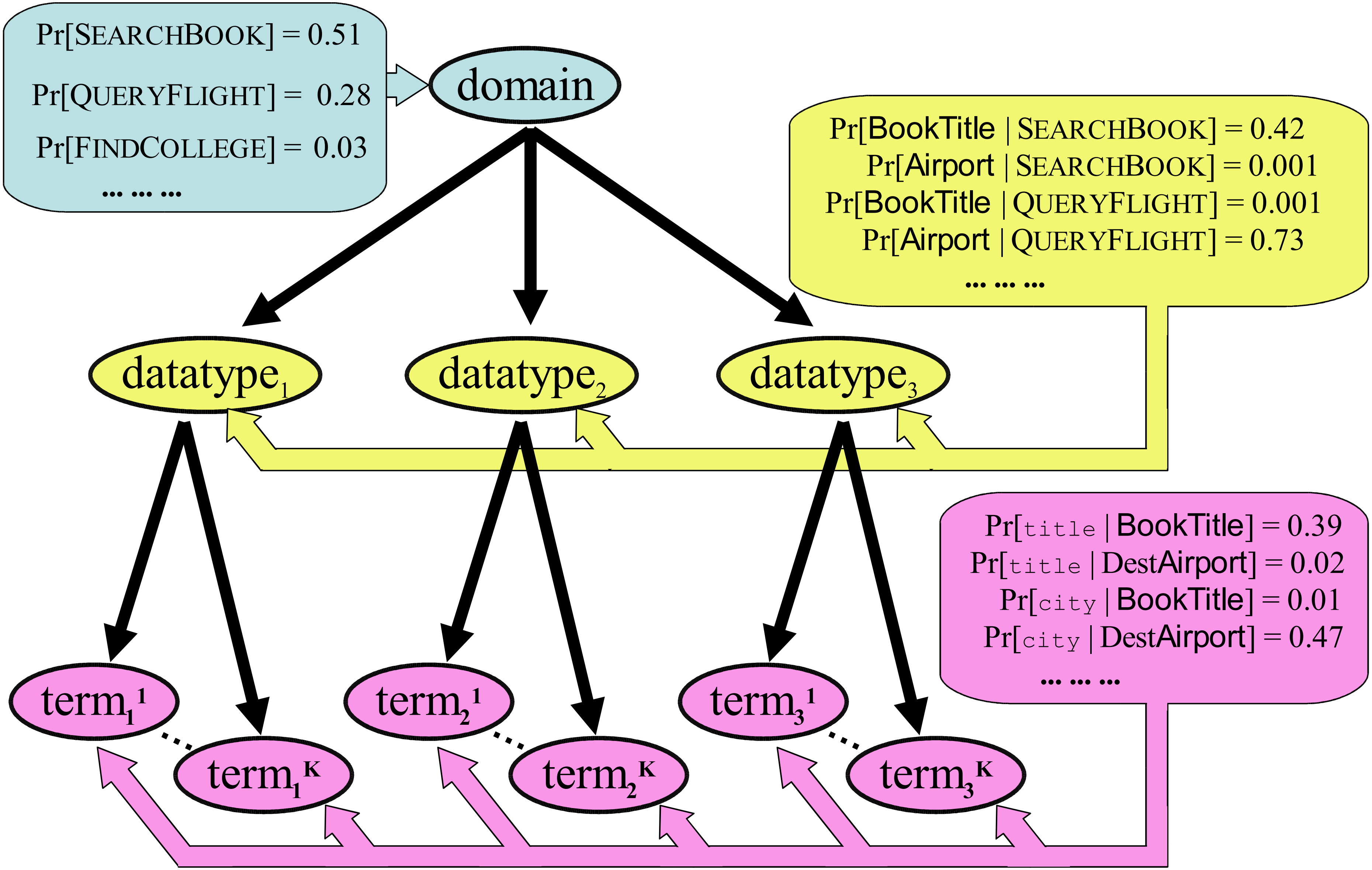
See ODBASE-03

Bayesian Networks

Bayesian Networks

- Nodes are random variables
- Edges indicate conditional probabilites

Bayesian Networks



Pr[SEARCHBOOK] = 0.51

Pr[QUERYFLIGHT] = 0.28

Pr[FINDCOLLEGE] = 0.03

... ...

domain

Pr[BookTitle | SEARCHBOOK] = 0.42

Pr[Airport | SEARCHBOOK] = 0.001

Pr[BookTitle | QUERYFLIGHT] = 0.001

Pr[Airport | QUERYFLIGHT] = 0.73

... ...

datatype_1 datatype_2 datatype_3

Pr[`title` | BookTitle] = 0.39

Pr[`title` | DestAirport] = 0.02

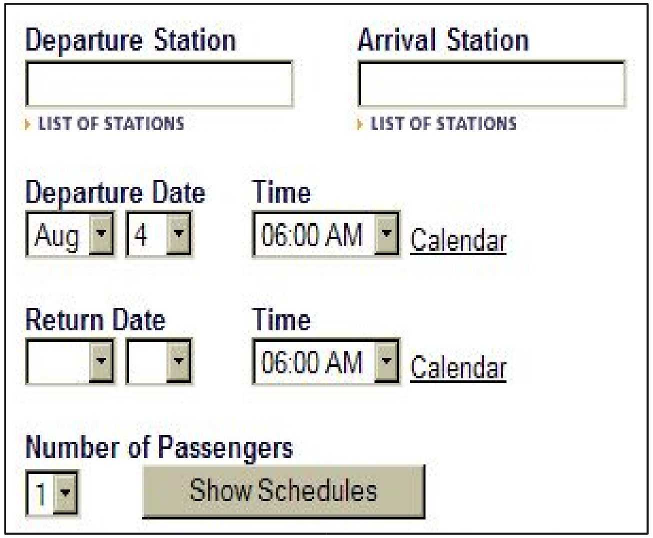
Pr[`city` | BookTitle] = 0.01

Pr[`city` | DestAirport] = 0.47

... ...

term_1^1 term_1^K term_2^1 term_2^K term_3^1 term_3^K

Web Forms



Departure Station

LIST OF STATIONS

Arrival Station

LIST OF STATIONS

Departure Date: Aug, 4

Time: 06:00 AM

Calendar

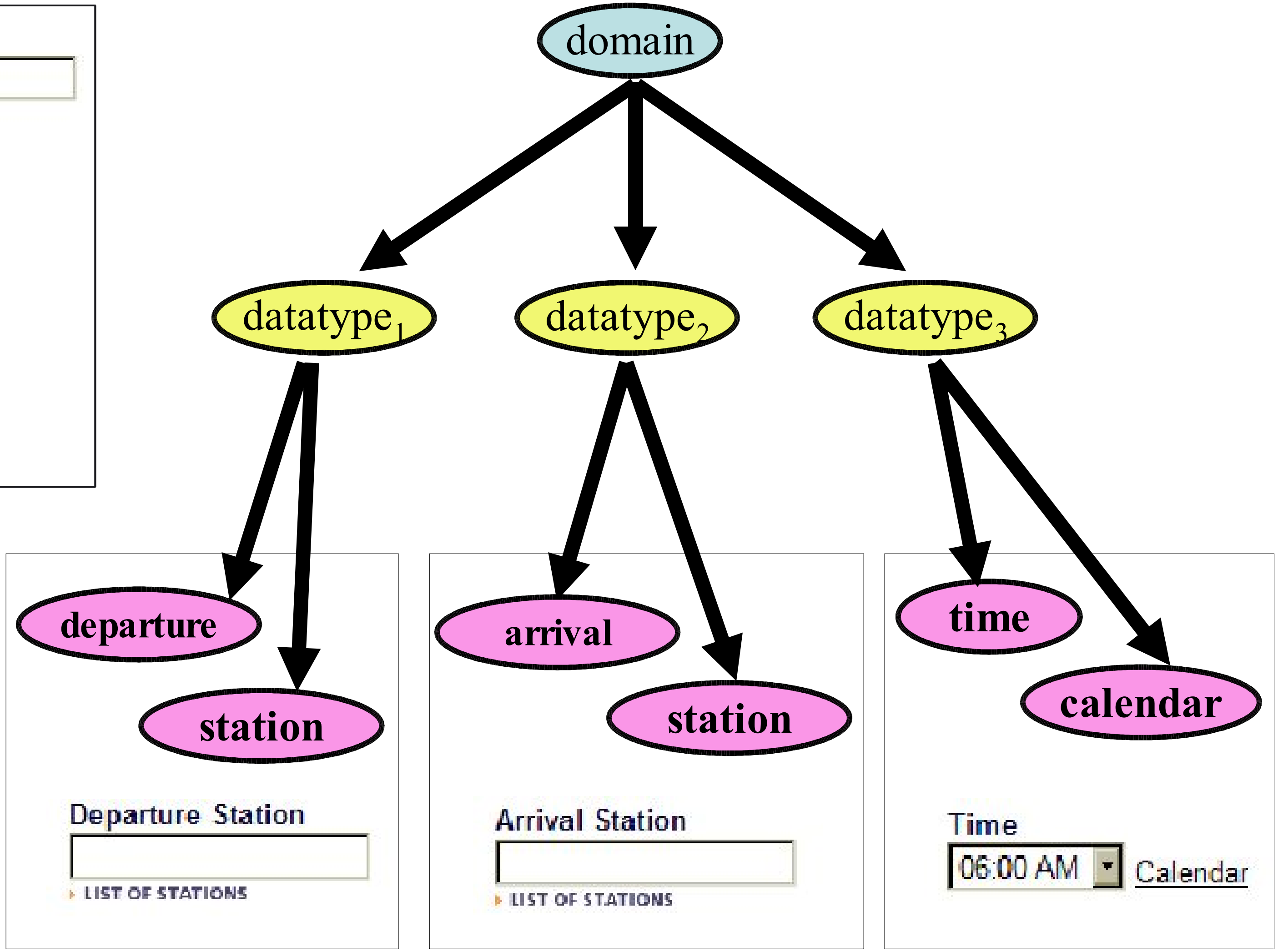
Return Date

Time: 06:00 AM

Calendar

Number of Passengers: 1

Show Schedules



Evaluation

129 real Web forms

656 fields

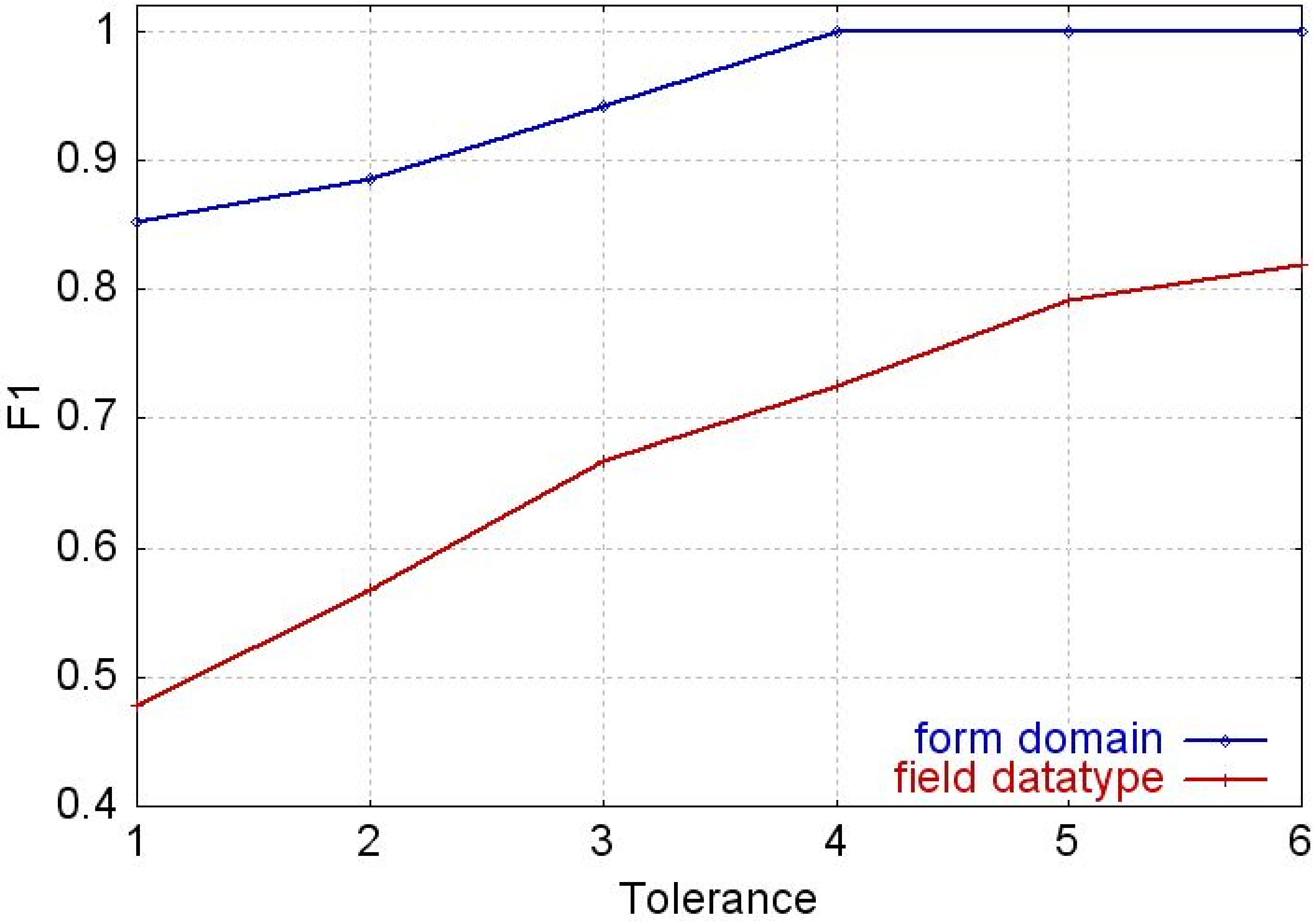
6 domains

Search Book (44)	Find College (2)	Search College Book (17
Query Flight (34)	Find Job (28)	Find Stock Quote (9)

72 datataypes (illustrative sample)

Book Details	Book Edition	Book Format	Book Search Type
Book Title	Number of Children	City	Class
College Subject	Company Name	Country	Currency
...	...	...	...
Ticker Symbol	Time	Time Return	Time Period
Travel Type	US State	ZIP	

Evaluation


1
0.9
0.8
0.7
0.6
0.5
0.4
F1
1
2
3
4
5
6
Tolerance
form domain
field datatype

Web Forms?

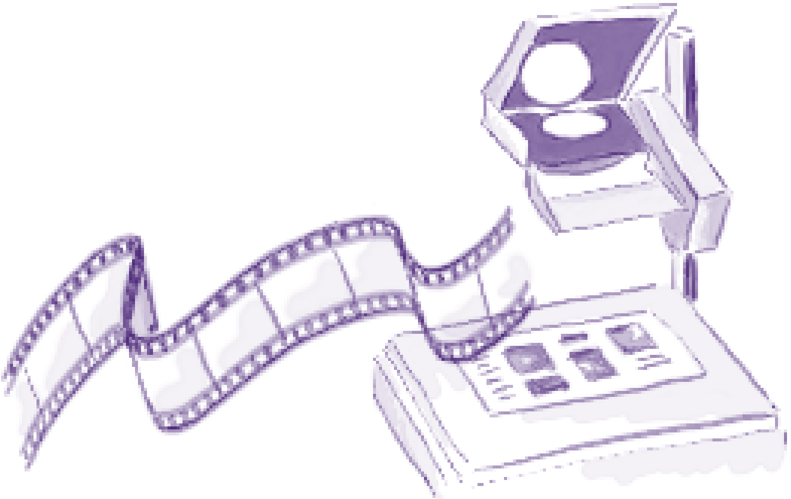
- Why forms? Not talking about services?
  - Fully annotated Web Services not available at time of experiments
  - Web Services actually easier: HTML parsing causes noise

✔ Introduction

✔ Our Machine Learning Approach

3. Machine Learning Assisted Annotation

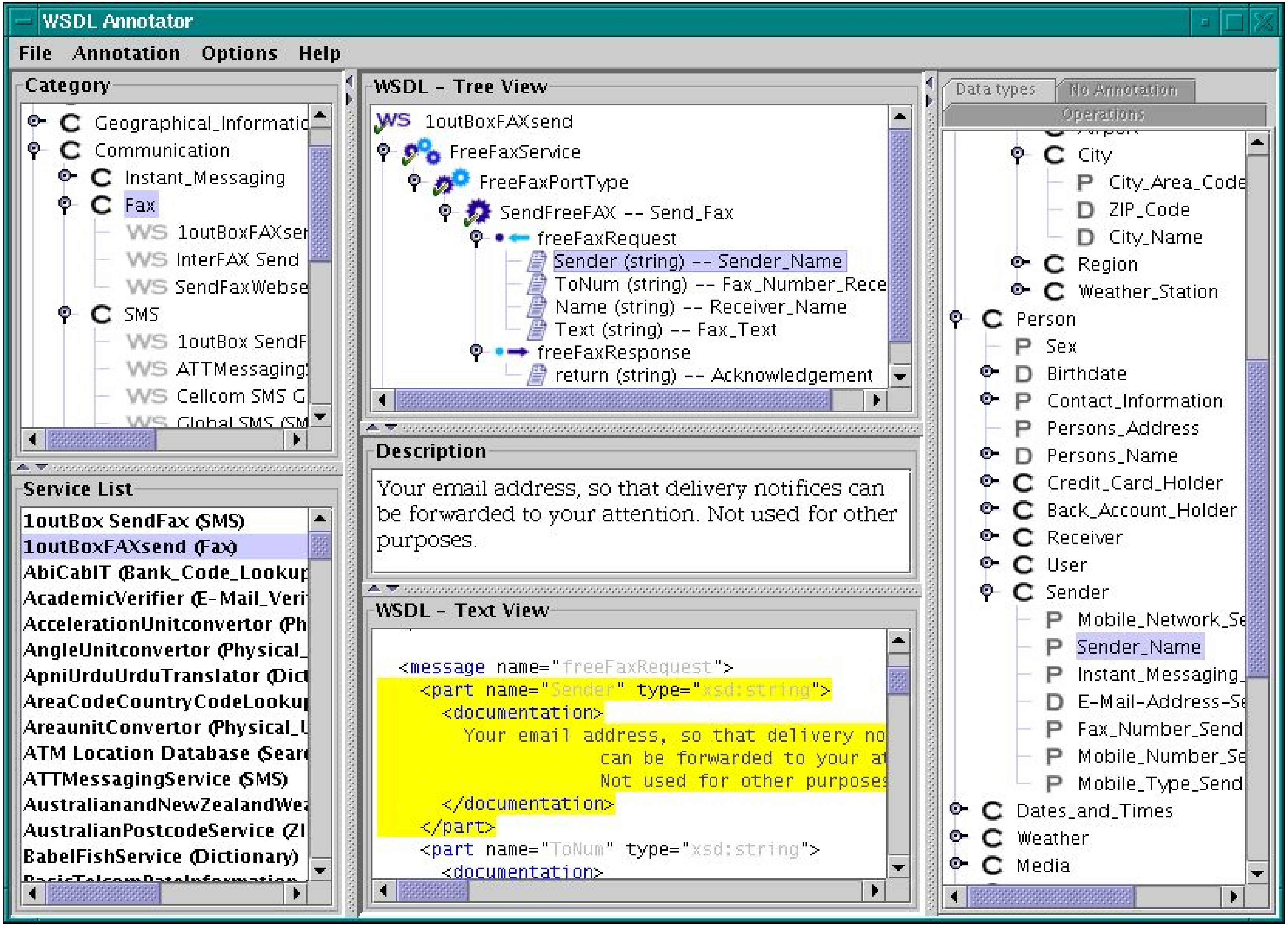
4. Conclusion & Discussion



Use Case

- To annotate several similar Web Services:
  - Hand-crafted annotation for the first few
  - Machine-assisted annotation for the rest
- Intended users:
  - End-users integrating several services
  - At service registries

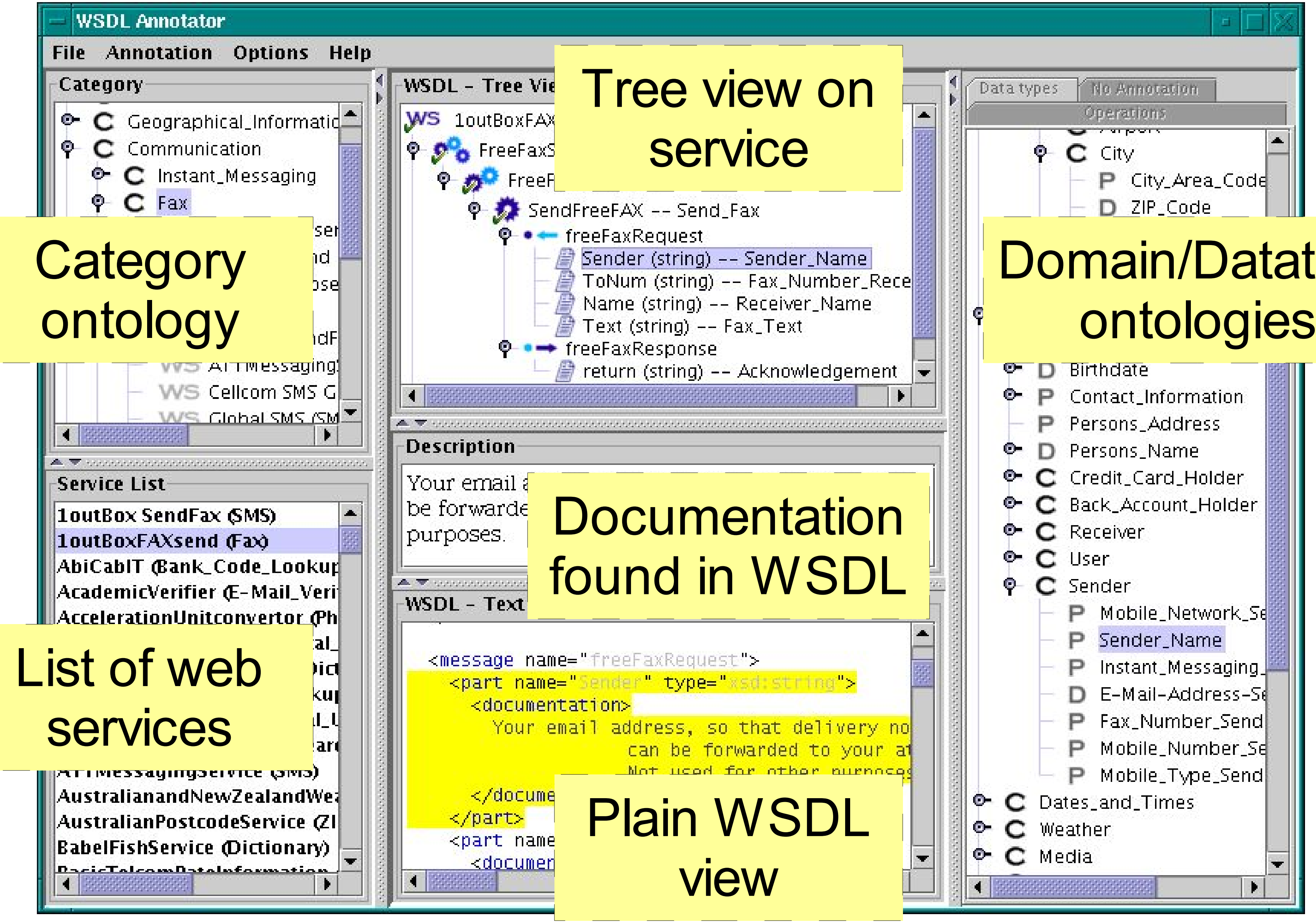
Our Application



Your email address, so that delivery notifices can be forwarded to your attention. Not used for other purposes.

```
<message name="freeFaxRequest">
  <part name="Sender" type="xsd:string">
    <documentation>
      Your email address, so that delivery no
            can be forwarded to your at
            Not used for other purposes
    </documentation>
  </part>
  <part name="ToNum" type="xsd:string">
    <documentation>
```

Our Application



Category

- C Geographical_Informatio
- C Communication
  - C Instant_Messaging
  - C Fax
    - WS AT Messaging
    - WS Cellcom SMS G
    - WS Global SMS (SM

Service List

1outBox SendFax (SMS)
1outBoxFAXsend (Fax)
AbiCabIT (Bank_Code_Lookup
AcademicVerifier (E-Mail_Veri
AccelerationUnitconvertor (Ph
ATMessagingService (SMS)
AustralianandNewZealandWea
AustralianPostcodeService (ZI
BabelFishService (Dictionary)
BasicTelcomDataInformation

WSDL - Tree View

- WS 1outBoxFAX
  - FreeFaxS
    - FreeF
      - SendFreeFAX -- Send_Fax
        - freeFaxRequest
          - Sender (string) -- Sender_Name
          - ToNum (string) -- Fax_Number_Rece
          - Name (string) -- Receiver_Name
          - Text (string) -- Fax_Text
        - freeFaxResponse
          - return (string) -- Acknowledgement

Description

Your email a
be forwarde
purposes.

WSDL - Text

```
  <message name="freeFaxRequest">
    <part name="Sender" type="xsd:string">
      <documentation>
        Your email address, so that delivery no
              can be forwarded to your at
              Not used for other purposes
      </docume
    </part>
    <part name
      <documen
```

Data types

- C Airport
- C City
  - P City_Area_Code
  - D ZIP_Code
- D Birthdate
- P Contact_Information
- P Persons_Address
- D Persons_Name
- C Credit_Card_Holder
- C Back_Account_Holder
- C Receiver
- C User
- C Sender
  - P Mobile_Network_Se
  - P Sender_Name
  - P Instant_Messaging_
  - D E-Mail-Address-Se
  - P Fax_Number_Send
  - P Mobile_Number_Se
  - P Mobile_Type_Send
- C Dates_and_Times
- C Weather
- C Media

Tree view on service

Category ontology

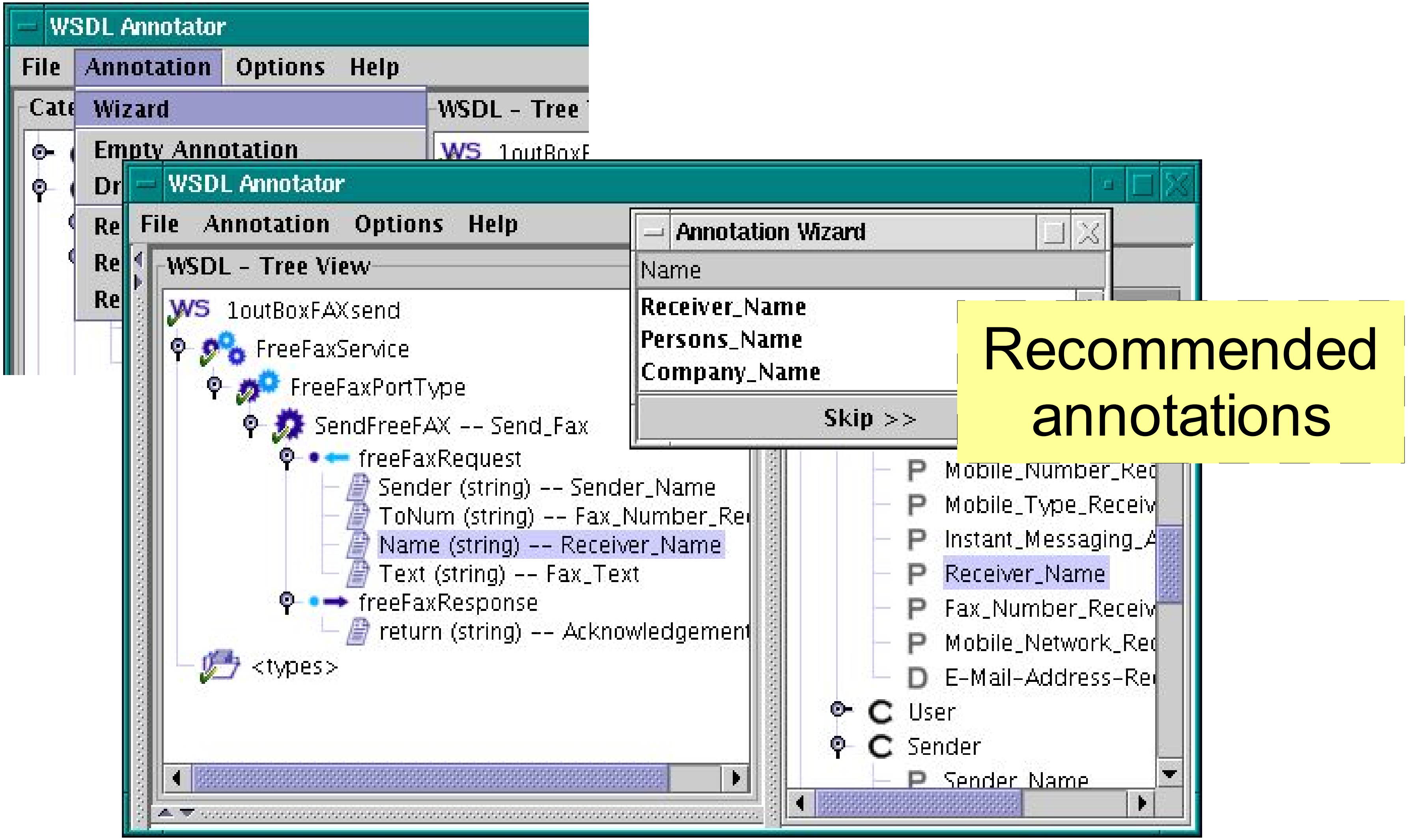
Domain/Datatype ontologies

Documentation found in WSDL

List of web services

Plain WSDL view

Annotation Wizard



WSDL Annotator

File Annotation Options Help

Wizard

Empty Annotation

WSDL - Tree View

WS 1outBoxFAXsend

- FreeFaxService
  - FreeFaxPortType
    - SendFreeFAX -- Send_Fax
      - freeFaxRequest
        - Sender (string) -- Sender_Name
        - ToNum (string) -- Fax_Number_Re
        - Name (string) -- Receiver_Name
        - Text (string) -- Fax_Text
      - freeFaxResponse
        - return (string) -- Acknowledgement
- <types>

Annotation Wizard

Name

Receiver_Name

Persons_Name

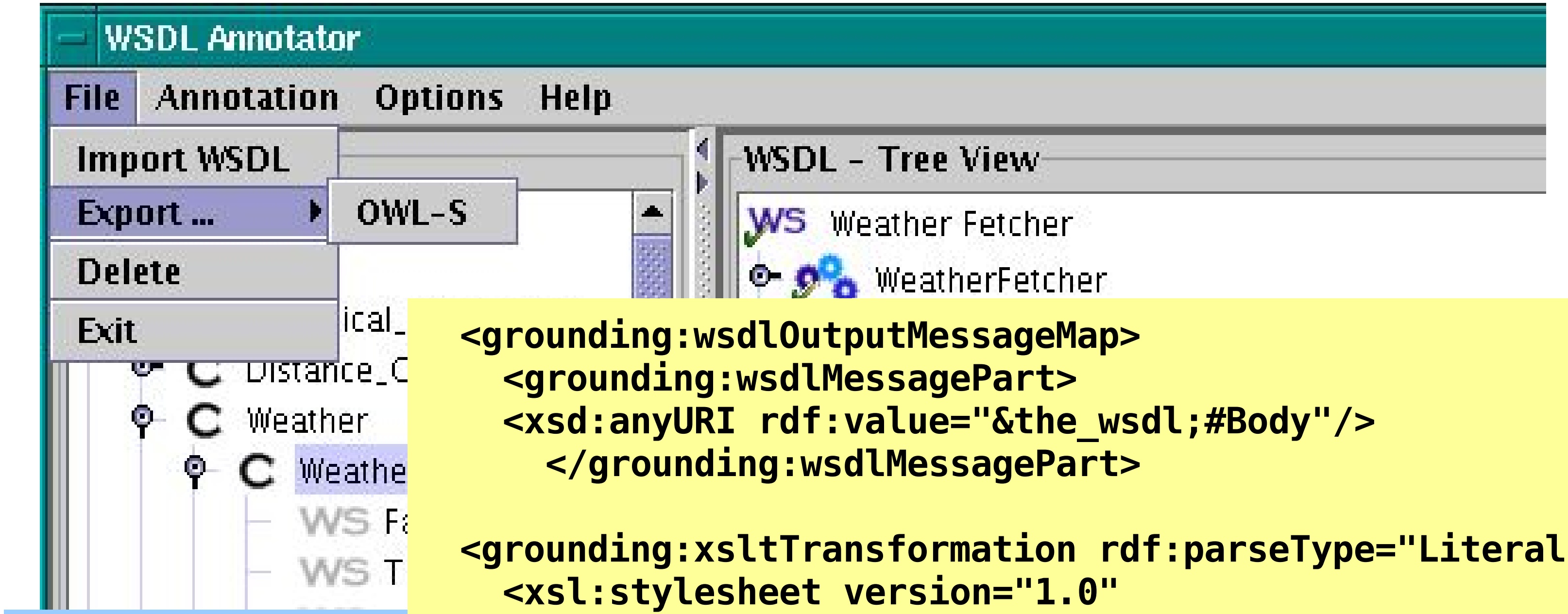
Company_Name

Skip >>

- P Mobile_Number_Rec
- P Mobile_Type_Receiv
- P Instant_Messaging_A
- P Receiver_Name
- P Fax_Number_Receiv
- P Mobile_Network_Rec
- D E-Mail-Address-Re
- C User
- C Sender
  - P Sender_Name

Recommended annotations

OWL-S Export



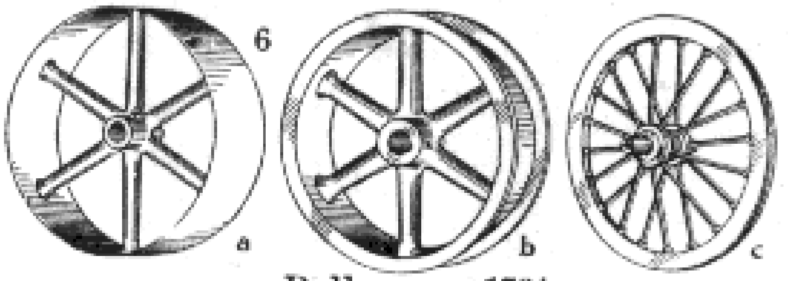
```
<grounding:wsdlOutputMessageMap>
  <grounding:wsdlMessagePart>
  <xsd:anyURI rdf:value="&the_wsdl;#Body"/>
    </grounding:wsdlMessagePart>

<grounding:xsltTransformation rdf:parseType="Literal">
  <xsl:stylesheet version="1.0"
       xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" indent="yes"/>
    <xsl:template match="/">
      <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
               xmlns:the_concepts="&the_concepts;"
               xmlns:the_process="&the_process;">
        <the_concepts:Weather>
          <the_concepts:Time>
            <xsl:value-of select="Body/Time"/>
          </the_concepts:Time>
```

```
<owl:Class rdf:ID="
   <rdfs:subClassO
</owl:Class>
<owl:DatatypePrope
  <rdfs:range rdf:
  <rdfs:domain rdf
  <rdfs:subPropert
</owl:DatatypeProp
<owl:DatatypePrope
  <rdfs:range rdf:
  <rdfs:domain rdf:resource="#Weather"/>
  <rdfs:subPropertyOf rdf:resource="http://moguntia.ucd.ie/owl/Datatypes.owl#Temperature"/>
</owl:DatatypeProperty>
```

OWL-S Export

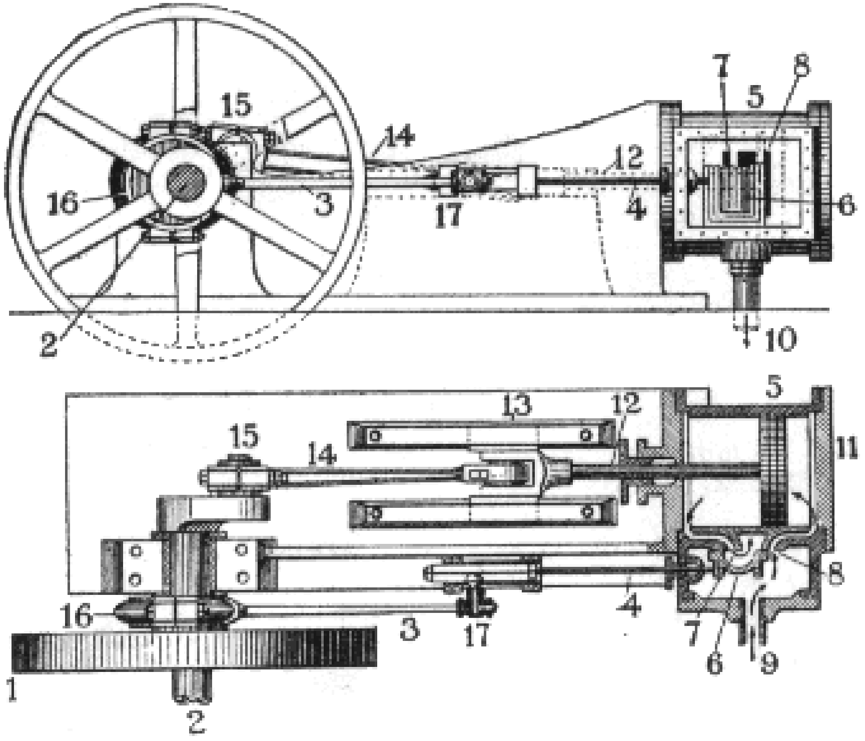
- The OWL-S Export generates not only:
  - Grounding, Profile, Process Model, Concepts (like WSDL2DAML-S by Paolucci, Sycara & al.)



Pulleys, p. 1734.

OWL-S Export

- But also the harder stuff:
  - XSLT Transformations in Grounding
  - Use shared ontologies for Concepts, Profile

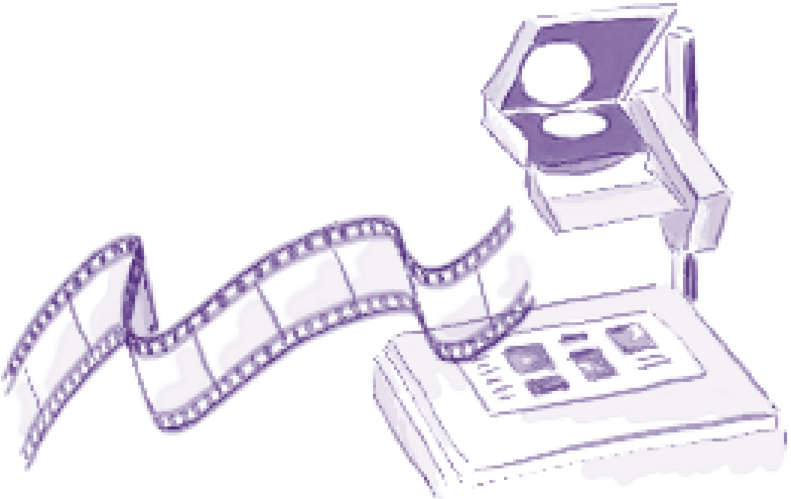


Steam Engine, p. 2038.

Machine Learning for Annotating Semantic Web Services

- ✔ Introduction
- ✔ Our Machine Learning Approach
- ✔ Machine Learning Assisted Annotation

4. Conclusion & Discussion



Contributions

- Summary
  - Machine Learning
    a) Ensemble learning for classifying Web Services categories
    b) Iterative classification for classifying complete Web Services
    c) Bayesian inference algorithm for classifying forms
  - Tool for Assisted Annotation

Open Issues & Limitations

- Open issues
  - Predictions can never be good enough!
  - Upper ontologies used for annotation
- Limitations: We do not handle...
  - Composite Processes
  - Composition & Workflow

Repository of Semantic Web Services

- We have annotated Web Services!
- Visit our Repository of Semantic Web Services:

smi.ucd.ie/RSWS

Discussion & Questions

- Questions?