ASSAM: A Tool for Semi-Automatically Annotating Semantic Web Services

Andreas Heß, Eddie Johnston and Nicholas Kushmerick
University College Dublin, Ireland

November 10, 2004

Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

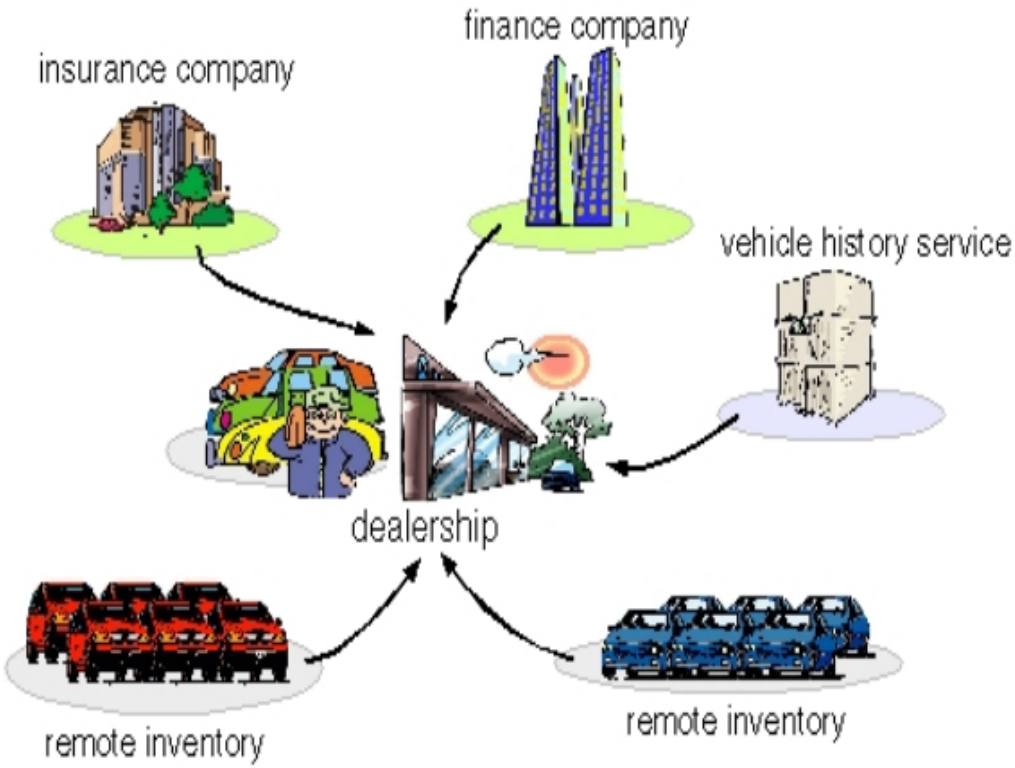
Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

Semantic Web Services

Desired Features

- Automatic discovery
- Automatic composition
- Automatic invocation



Semantic Metadata

Assumptions

- Semantic annotation, e.g. OWL-S
- Shared ontology

Problems

- Hand-crafting annotations can be tedious
- Integrate legacy web services

Our Solution

- Learn mappings to ontology!

Semantic Metadata

Assumptions

- Semantic annotation, e.g. OWL-S
- Shared ontology

Problems

- Hand-crafting annotations can be tedious
- Integrate legacy web services

Our Solution

- Learn mappings to ontology!

Semantic Metadata

Assumptions

- Semantic annotation, e.g. OWL-S
- Shared ontology

Problems

- Hand-crafting annotations can be tedious
- Integrate legacy web services

Our Solution

- Learn mappings to ontology!

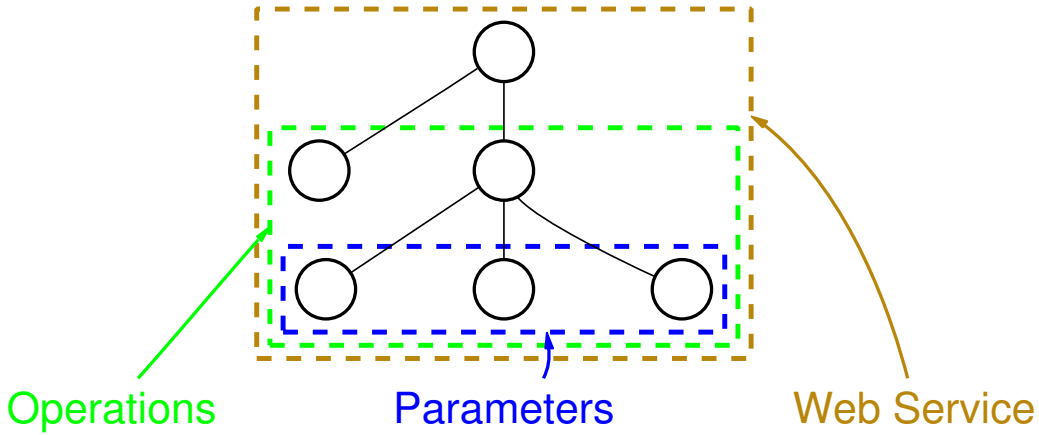
What is learned?

Learn from:

- Web Service as a whole
- Operations
- Parameters

Features used:

- Names
- Comments (if available)



Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

Progress

Earlier

- Classifying Services
- Ensemble Learning

Now

- Classifying Services, Operations, Datatypes
- *Relational* Learning
- Application that can export OWL-S
- New OATS algorithm for output aggregation

Progress

Earlier

- Classifying Services
- Ensemble Learning

Now

- Classifying Services, Operations, Datatypes
- *Relational* Learning
- Application that can export OWL-S
- New OATS algorithm for output aggregation

Progress

Earlier

- Classifying Services
- Ensemble Learning

Now

- Classifying Services, Operations, Datatypes
- *Relational* Learning
- Application that can export OWL-S
- New OATS algorithm for output aggregation

Progress

Earlier

- Classifying Services
- Ensemble Learning

Now

- Classifying Services, Operations, Datatypes
- *Relational* Learning
- Application that can export OWL-S
- New OATS algorithm for output aggregation

Progress

Earlier

- Classifying Services
- Ensemble Learning

Now

- Classifying Services, Operations, Datatypes
- *Relational* Learning
- Application that can export OWL-S
- New OATS algorithm for output aggregation

Progress

Earlier

- Classifying Services
- Ensemble Learning

Now

- Classifying Services, Operations, Datatypes
- *Relational* Learning
- Application that can export OWL-S
- New OATS algorithm for output aggregation

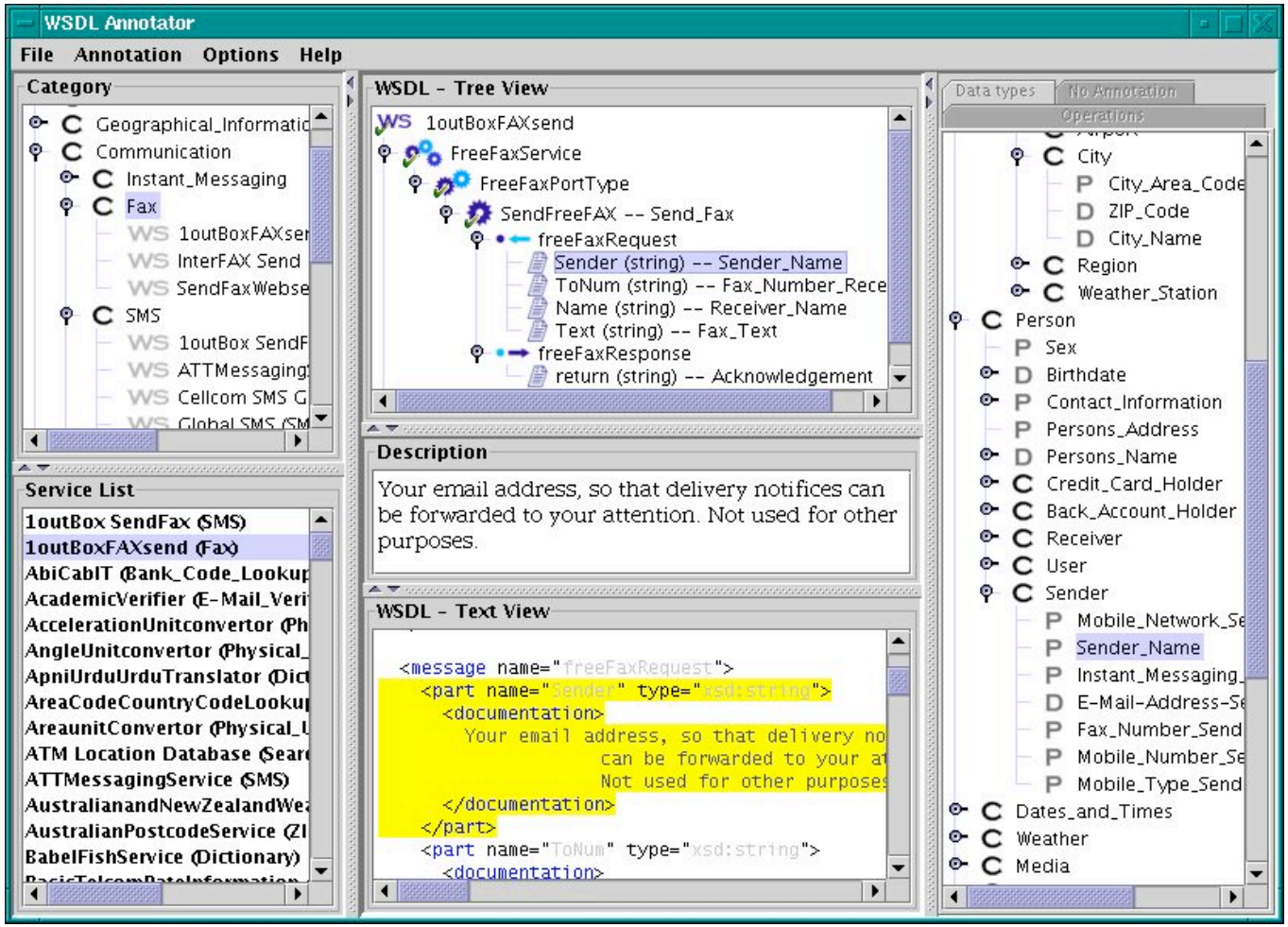
Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

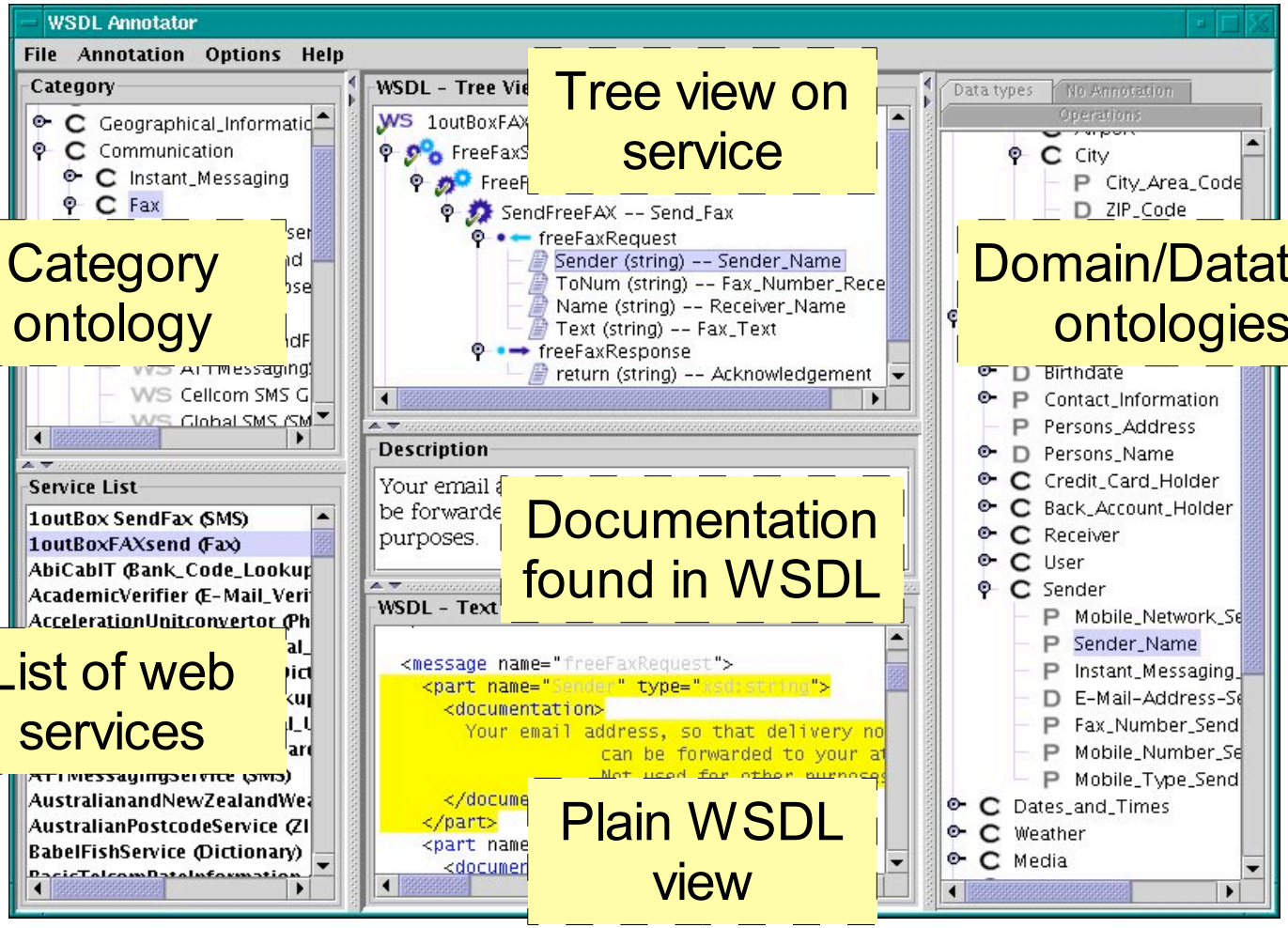
Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

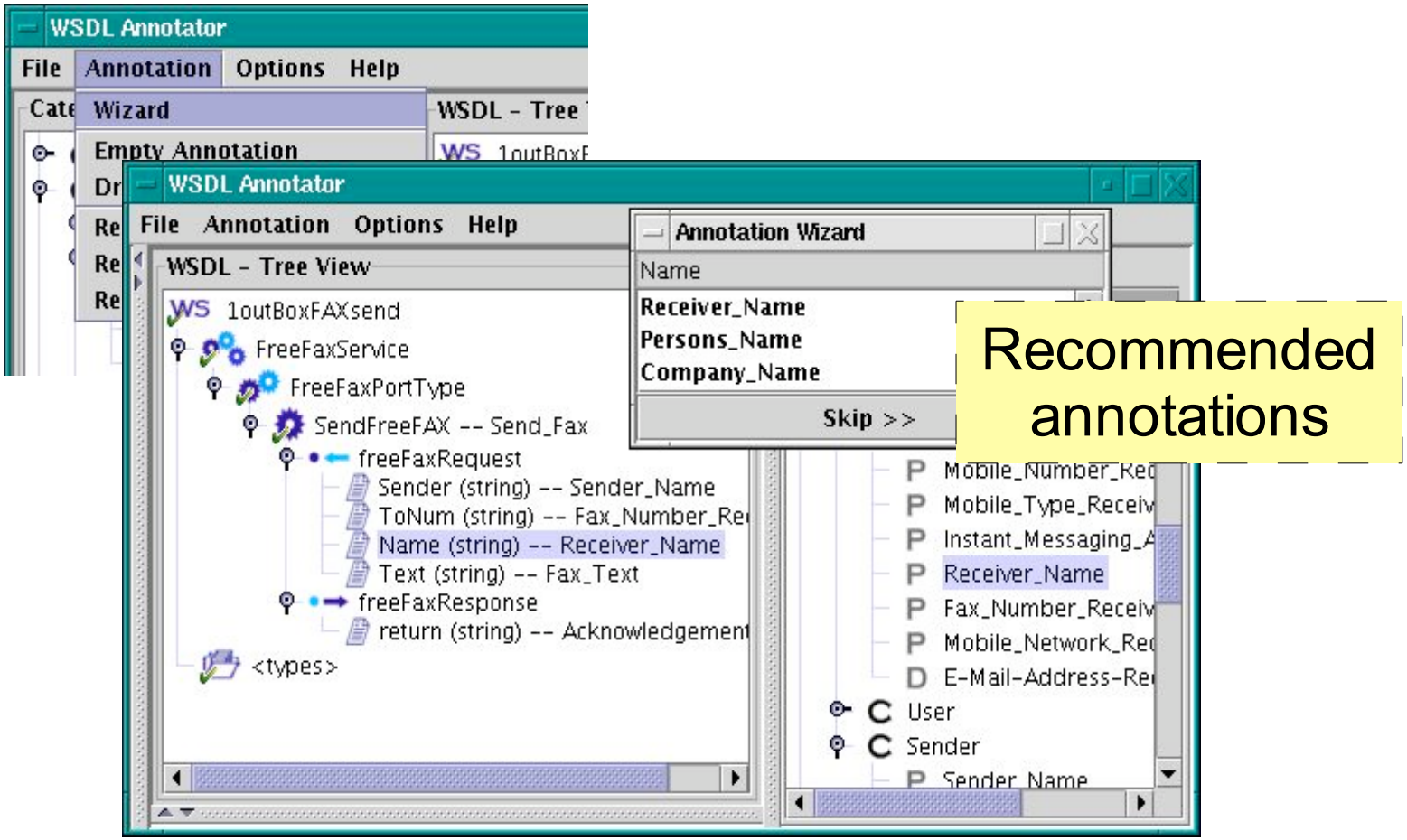
Our Application



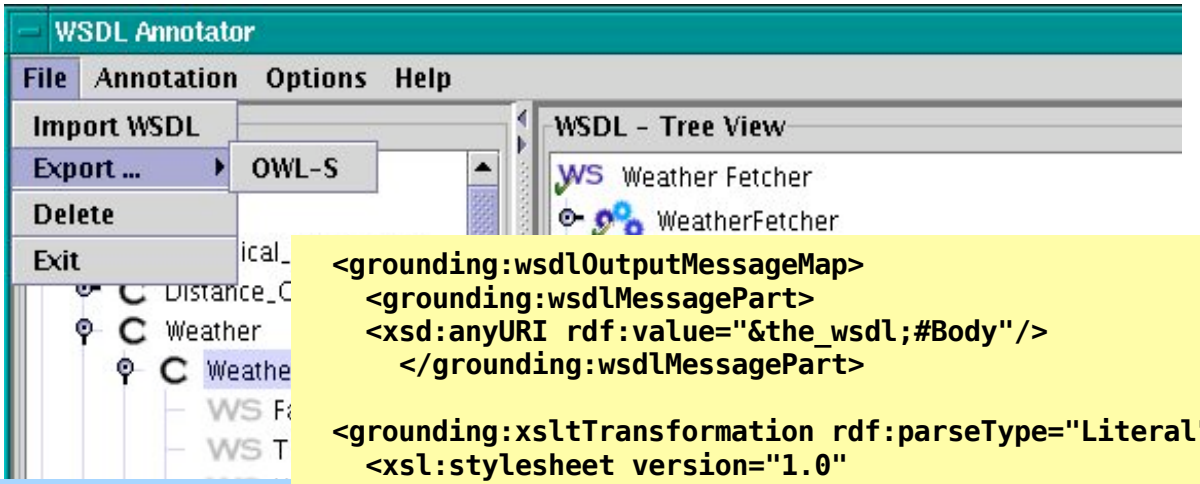
Our Application



ASSAM's Key Feature



OWL-S Export




```
<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>
```

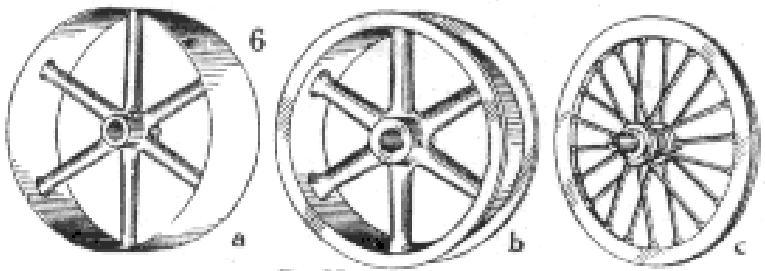
```
<grounding:wsdlOutputMessageMap>
  <grounding:wsdlMessagePart>
  <xsd:anyURI rdf:value="&the_wsdl;#Body"/>
    </grounding:wsdlMessagePart>

<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-S Export

ASSAM generates...

- Grounding
- Profile
- Process Model
- Concepts



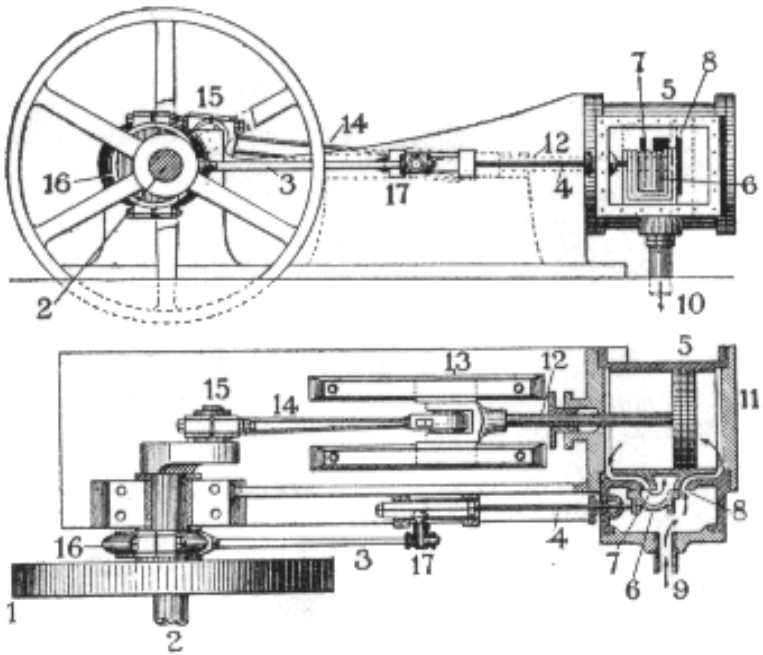
Pulleys, p. 1734.

(like WSDL2DAML-S by Paolucci, Sycara et al.)

OWL-S Export – Additional Features

But ASSAM also...

- generates XSLT transformation for Grounding
- uses shared ontologies for Concepts and Profile



Steam Engine, p. 2038.

Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

Definitions

Category

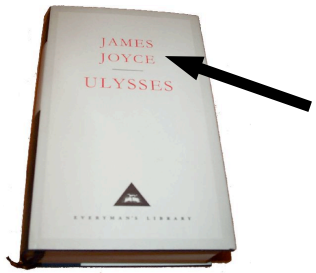
- Description of service as a whole
- e.g. Weather, Finance, Books
- Profile

Domain

- Purpose of single operation
- e.g. Query Price, Purchase Book
- PEs

Datatype

- Meaning of single parameter
- e.g. Title, Author's Name
- IOs

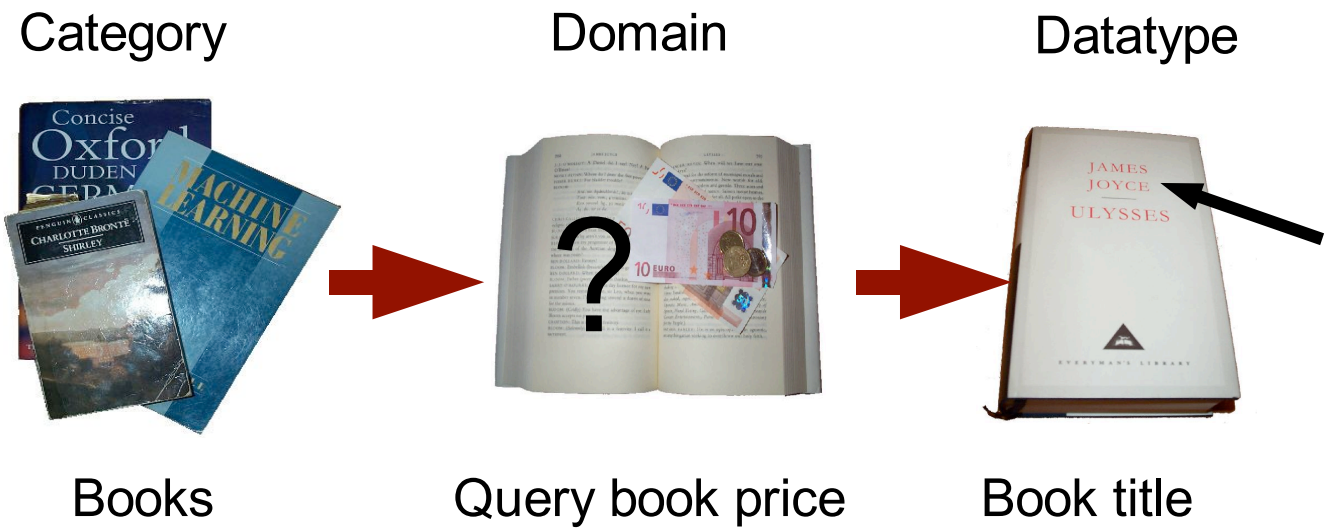


Definitions: What? Why?

No new standard!

- We do not propose a new standard or ontology language
- Concepts independent of language

Relations: Example



Relations: Example

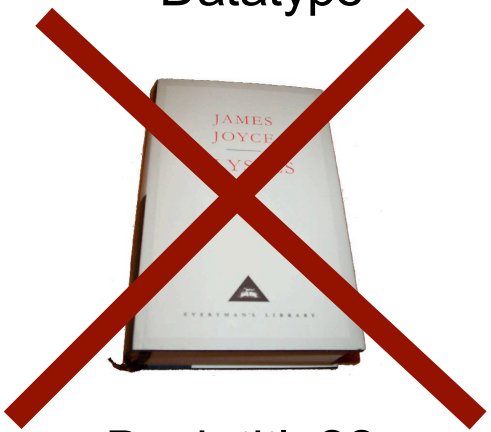
Category

Tea

Domain

Order tea

Datatype



Book title??

Relations: Example

Category

Tea

Domain

Order tea

Datatype

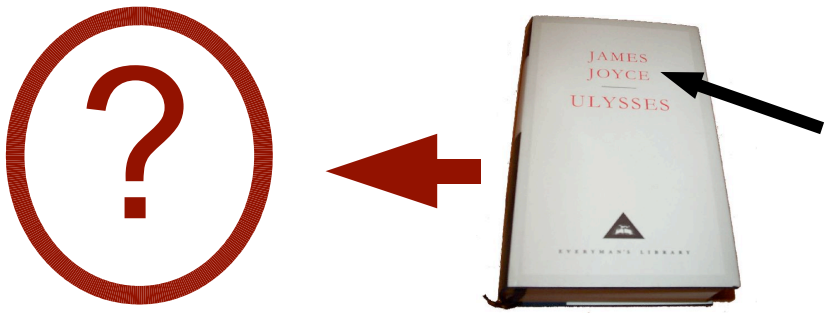
Credit card number

Relations: Example

Category

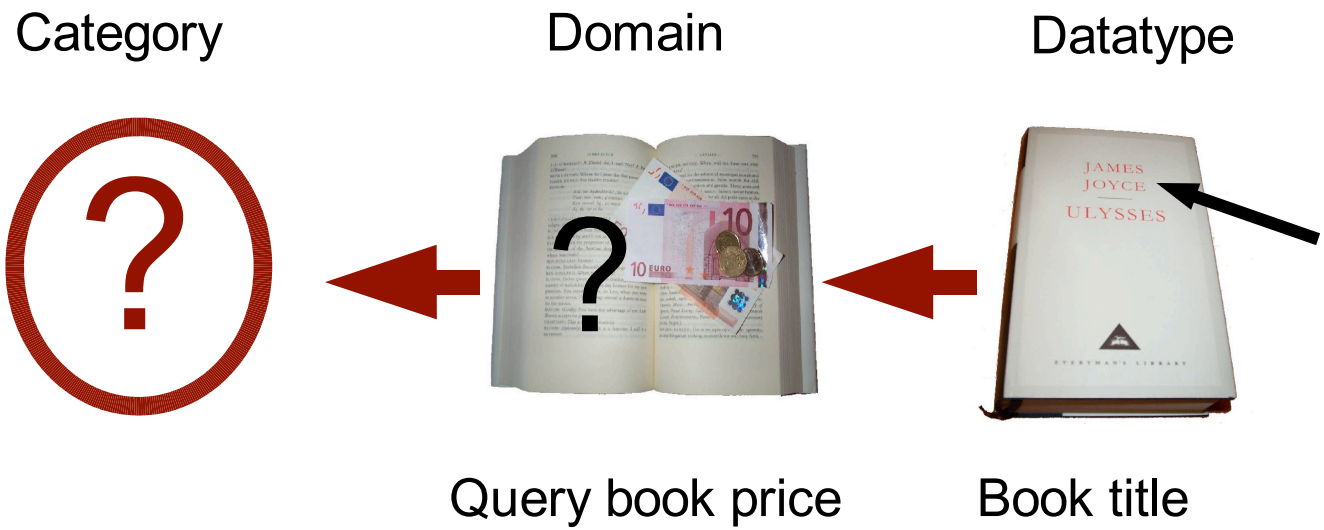
Domain

Datatype

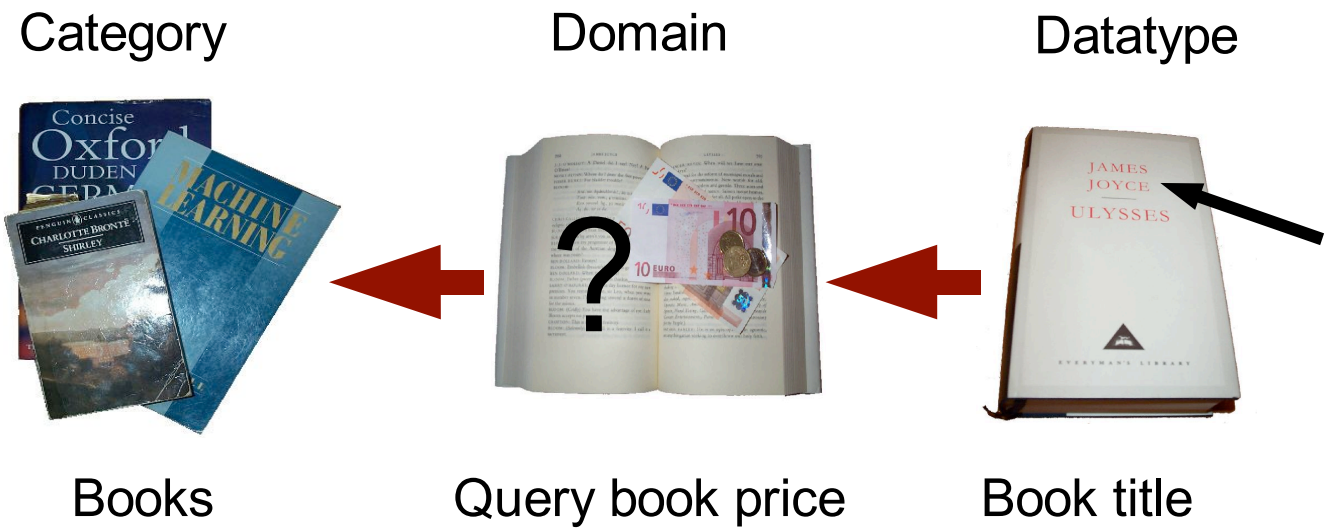


Book title

Relations: Example



Relations: Example



Relational Learning

Ways to exploit relations

- Bayesian Networks
  (see our earlier paper)
- Iterative Classification

Iterative Classification – Example

Classification in round n based on classification from round $n-1$

Category	Domain	Datatype
initially unknown	*initially unknown*	*initially unknown*

Iterative Classification – Example

Classification in round 1 is made in standard way

Category
- Communication

Domain
- Query tea price

Datatype
- Person's name

Iterative Classification – Example

Classification in round 2 is based on classifcation from round 1

Category

- Communication
- Tea

Domain

- Query tea price
- Query book price

Datatype

- Person's name
- Sender's name

Iterative Classification – Example

Classification in round 3 is based on classifcation from round 2

Category

- Communication
- Tea
- Books

Domain

- Query tea price
- Query book price
- Query book price

Datatype

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

Iterative Classification – Example

Observation

Iterative classification can correct initial errors!

Category

- Communication
- Tea
- Books

Domain

- Query tea price
- Query book price
- Query book price

Datatype

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

Two Views

Intrinsic View

- Features inherent to instance
- e.g. name of operation

Extrinsic View

- Relations between instances
- e.g. annotation of parameters

(Following Neville and Jensen)

Two Views

Intrinsic View

- Features inherent to instance
- e.g. name of operation

Extrinsic View

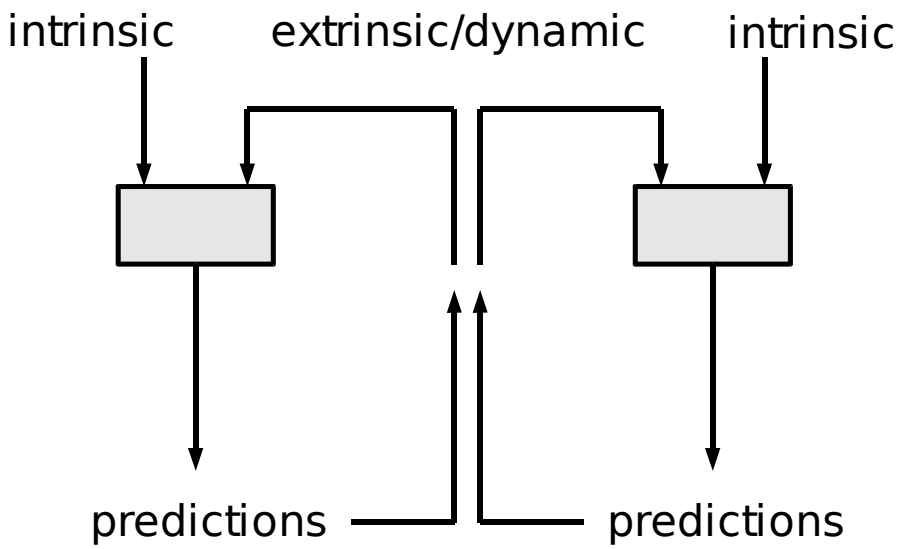
- Relations between instances
- e.g. annotation of parameters

(Following Neville and Jensen)

Iterative Algorithm for Relational Classification

Iterative Algorithm

- Features are combination of intrinsic and extrinsic
- Extrinsic view changes with new results
- Predictions are fed back
- Extrinsic view is *dynamic*

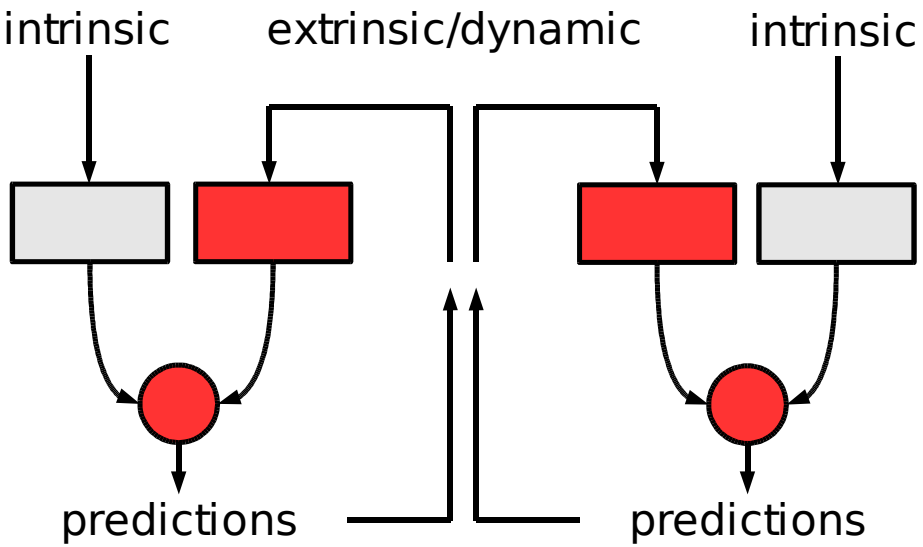


e.g. Neville/Jensen, Chakrabarti, Lu/Getoor

Iterative *Ensemble* Algorithm for Relational Classification

Iterative Ensemble Algorithm

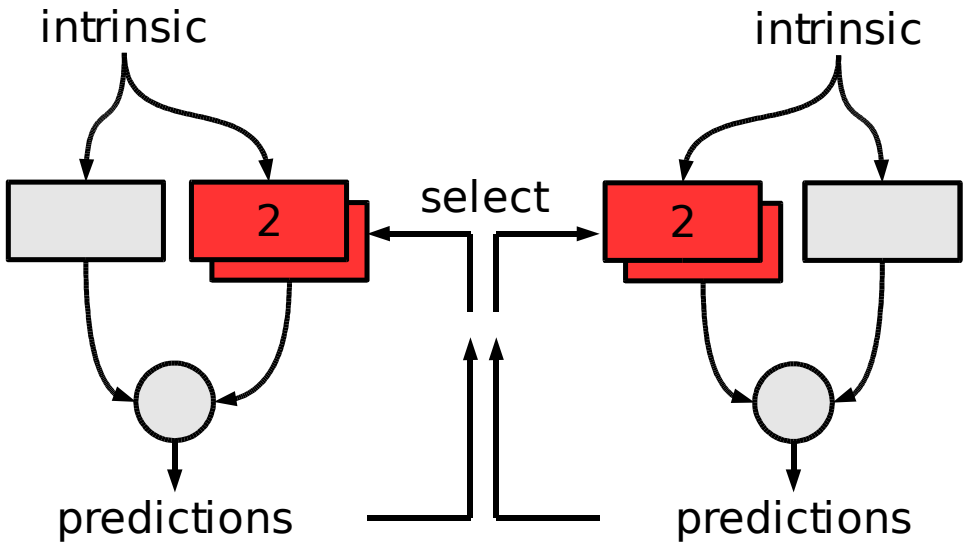
- Separate intrinsic/extrinsic classifiers
- Better performance with high-dimensional features



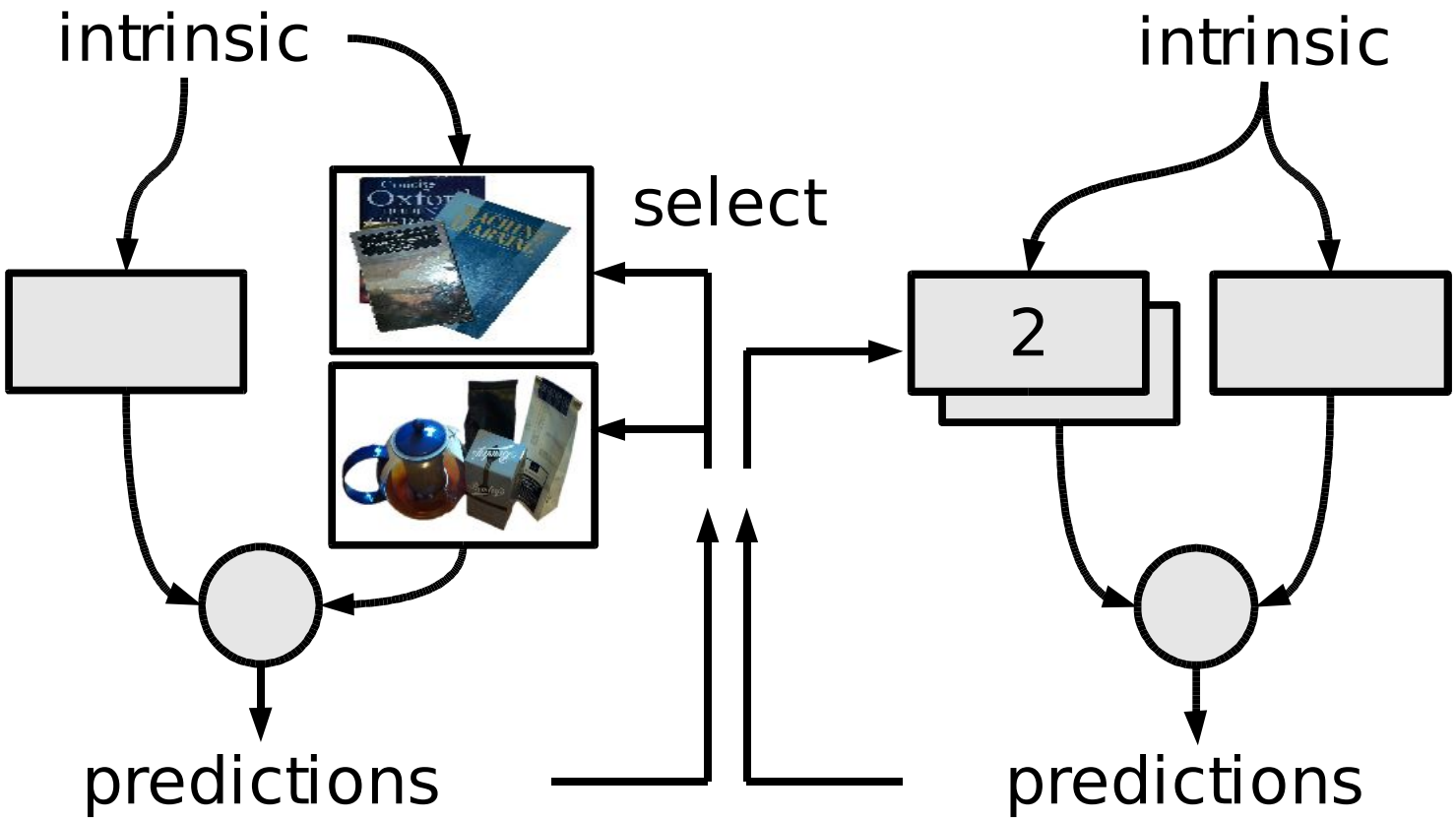
Iterative Ensemble Algorithm *with Specialised Classifiers*

Specialised Classifiers

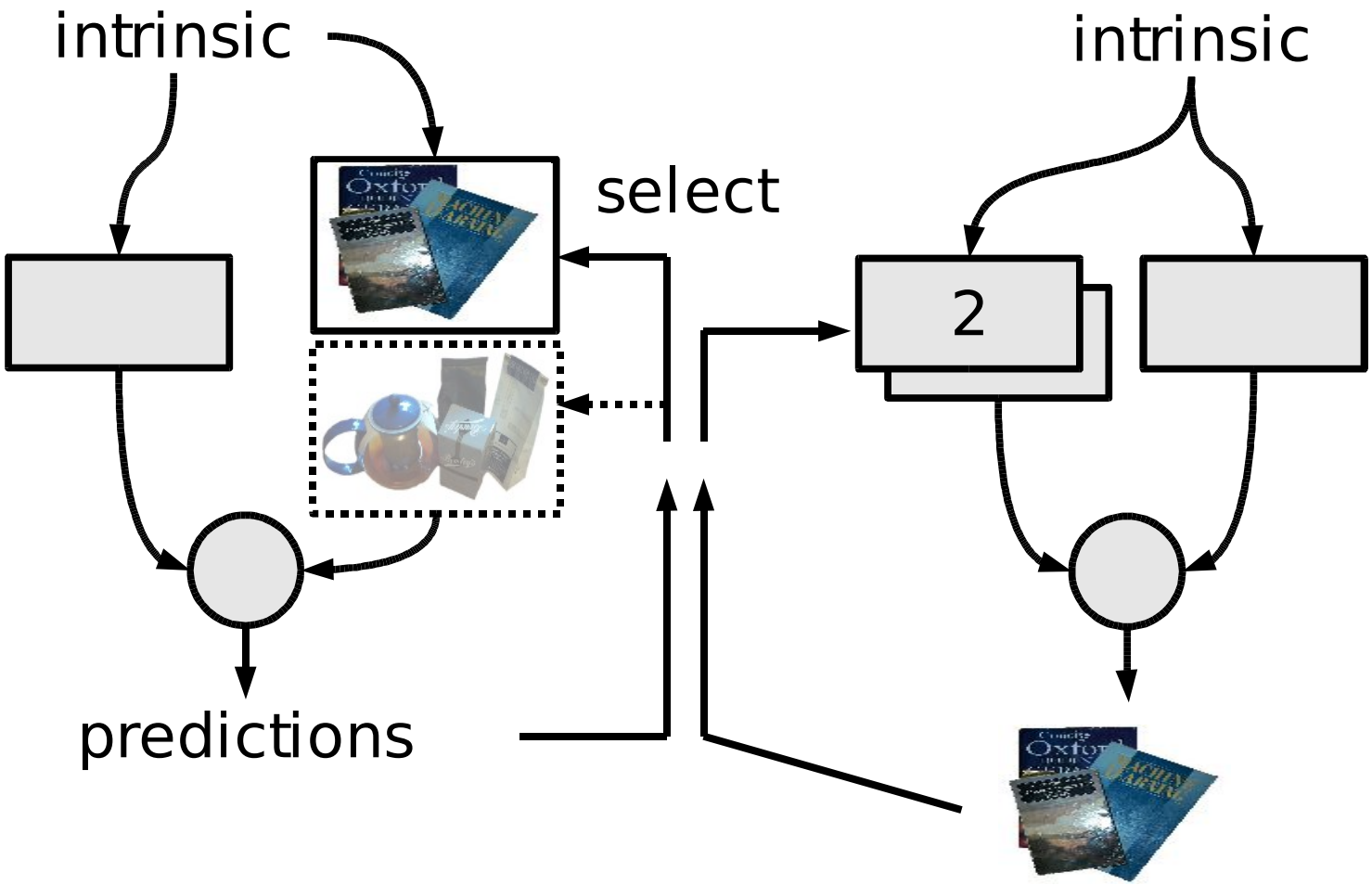
- Train several classifiers on intrinsic features
- Each one trained on subset of instances
- Classifier selected based on extrinsic features



Specialised Classifiers: Example



Specialised Classifiers: Example



Specialised Classifiers

Specialised Classifiers

- Useful if extrinsic view not sufficient on its own
- Extrinsic features serve as selector
- Idea can be applied in various ways

Specialised Classifiers

Specialised Classifiers

- Useful if extrinsic view not sufficient on its own
- Extrinsic features serve as selector
- Idea can be applied in various ways

Specialised Classifiers

Specialised Classifiers

- Useful if extrinsic view not sufficient on its own
- Extrinsic features serve as selector
- Idea can be applied in various ways

Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

Dataset

Our Web Services Dataset

- 164 Web Services in 22 Categories
- 1138 Operations in 136 Domains
- 5452 Parameters with 312 Datatypes

Download

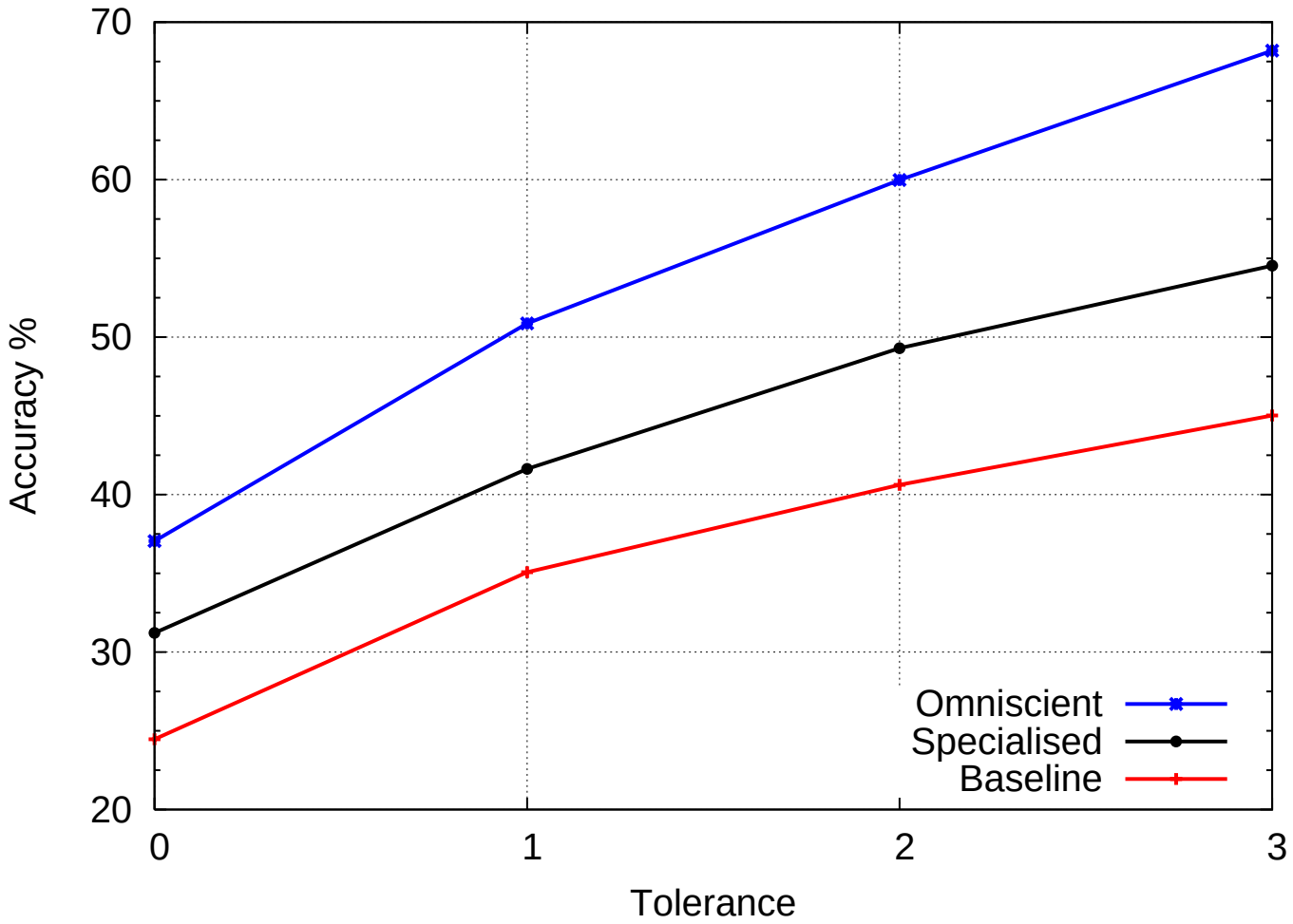
This collection is available for download from the Repository of Semantic Web Services: http://smi.ucd.ie/RSWS

Setup

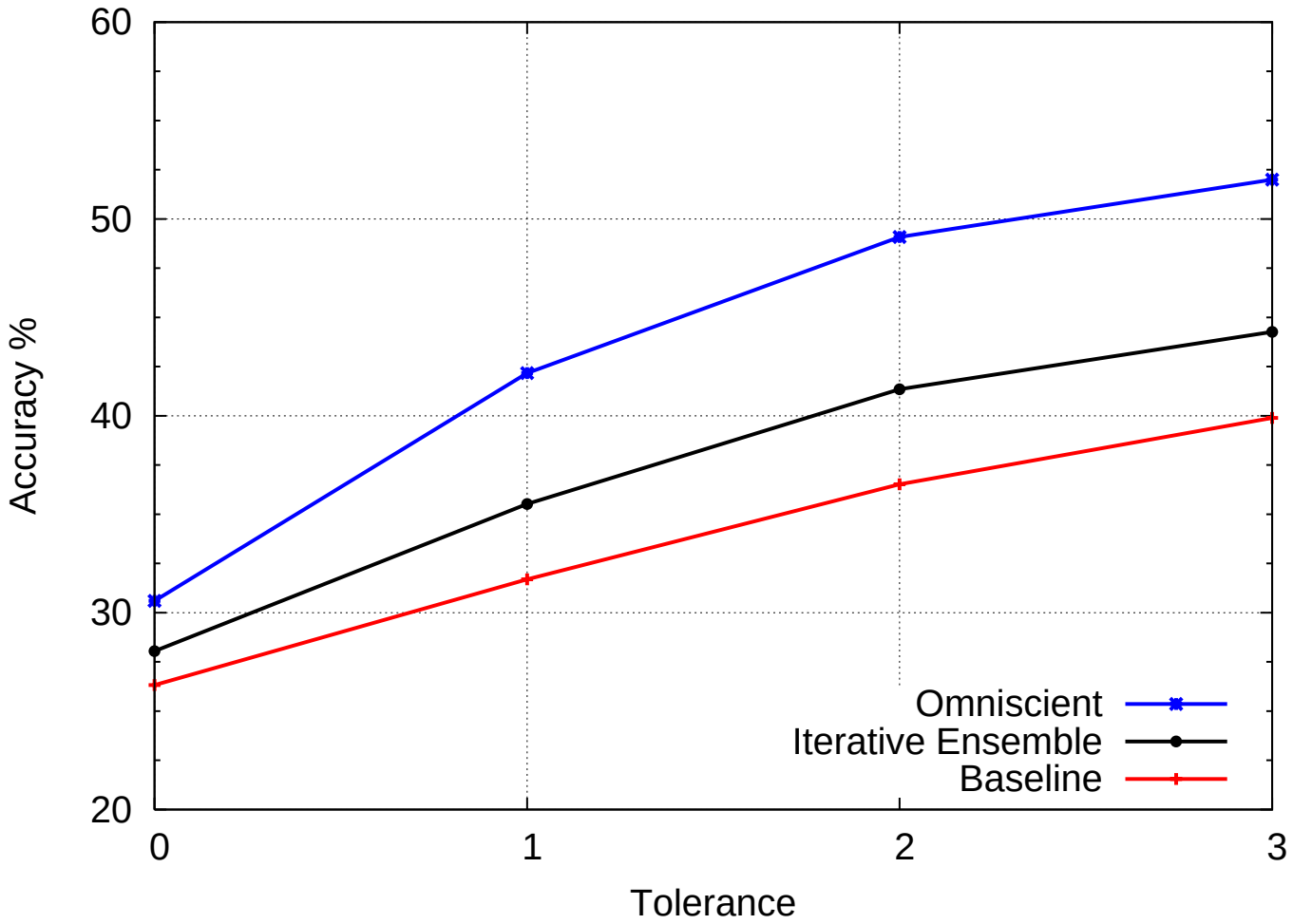
Experimental Setup

- Using iterative ensemble for service, operations
- Using specialised classifiers for datatypes
- Leave-one-service-out
- Compared to omniscient setup
  with extrinsic view always correct
- Compared to non-relational baseline
- Non-ensemble setting ommitted: always worse than baseline

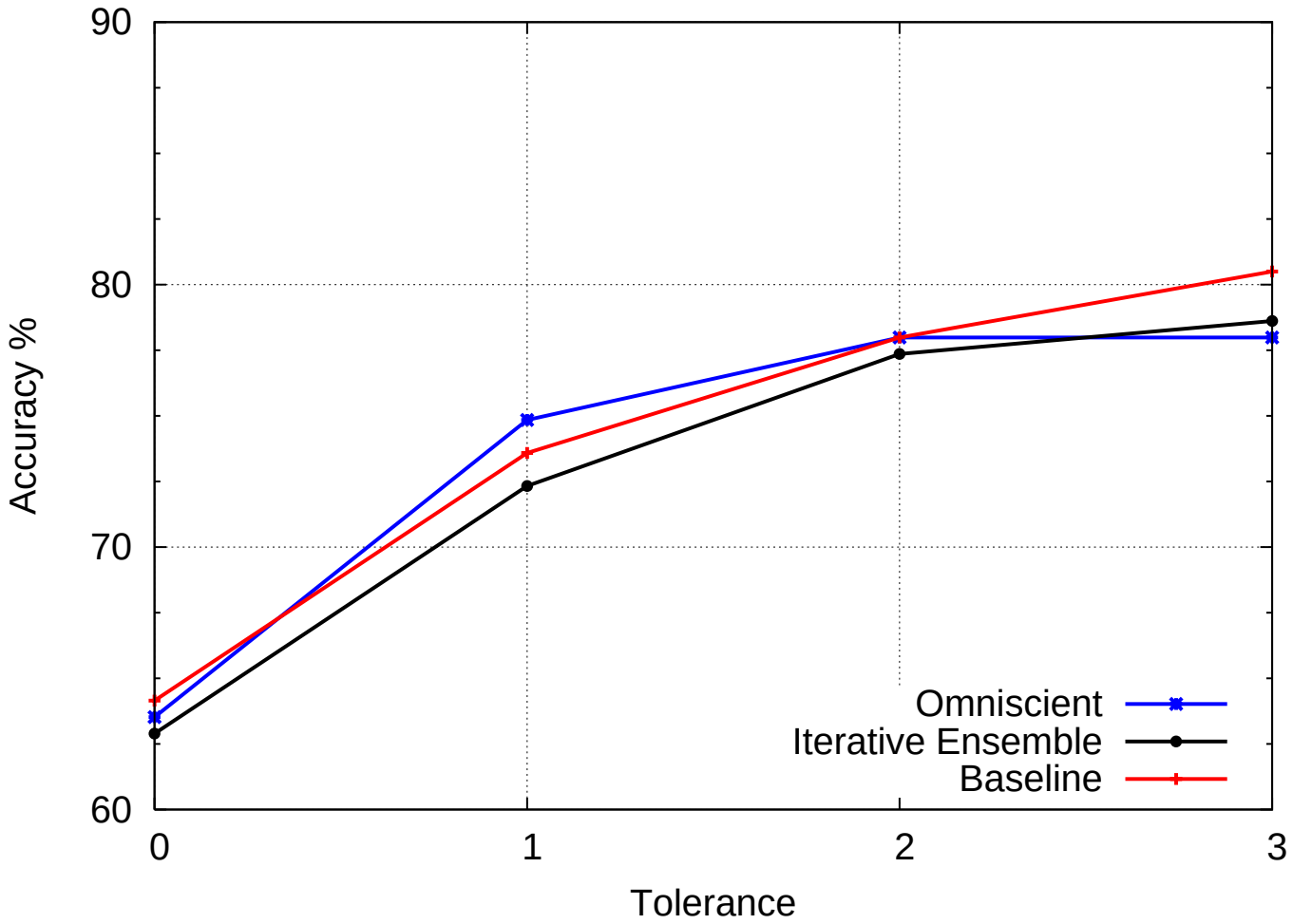
Evaluation: Datatype of Parameters



Evaluation: Domain of Operations



Evaluation: Category of Service



Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

Operation Aggregation Tool for Web Services

Key Features

- Aggregates output from (legacy) Web Services
- Considers instances data
- Uses probes to influence output
- Use string distance metrics to match outputs

ASSAM and OATS

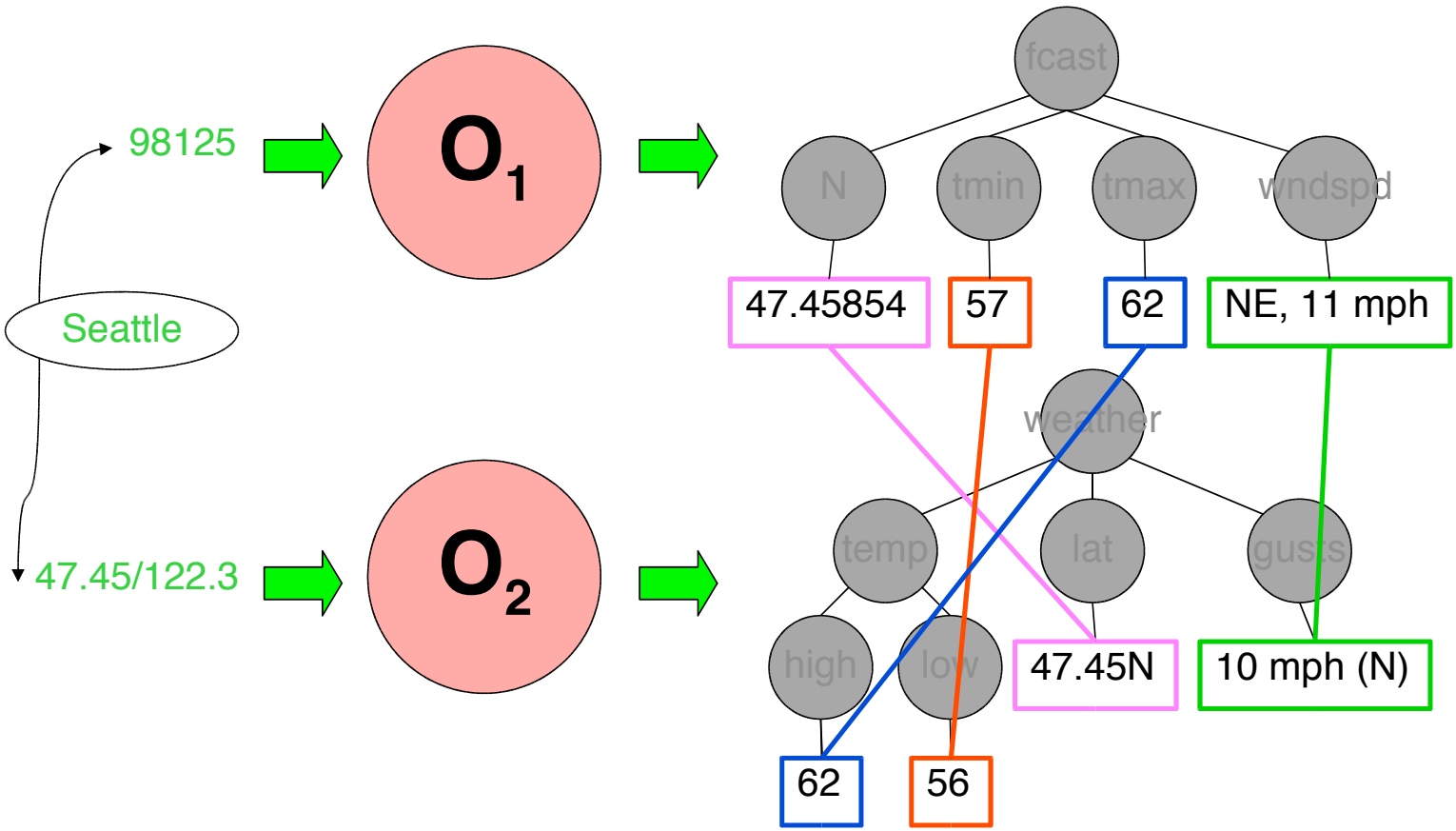
ASSAM

- Complete Service
- Ignores instance data
- Annotation

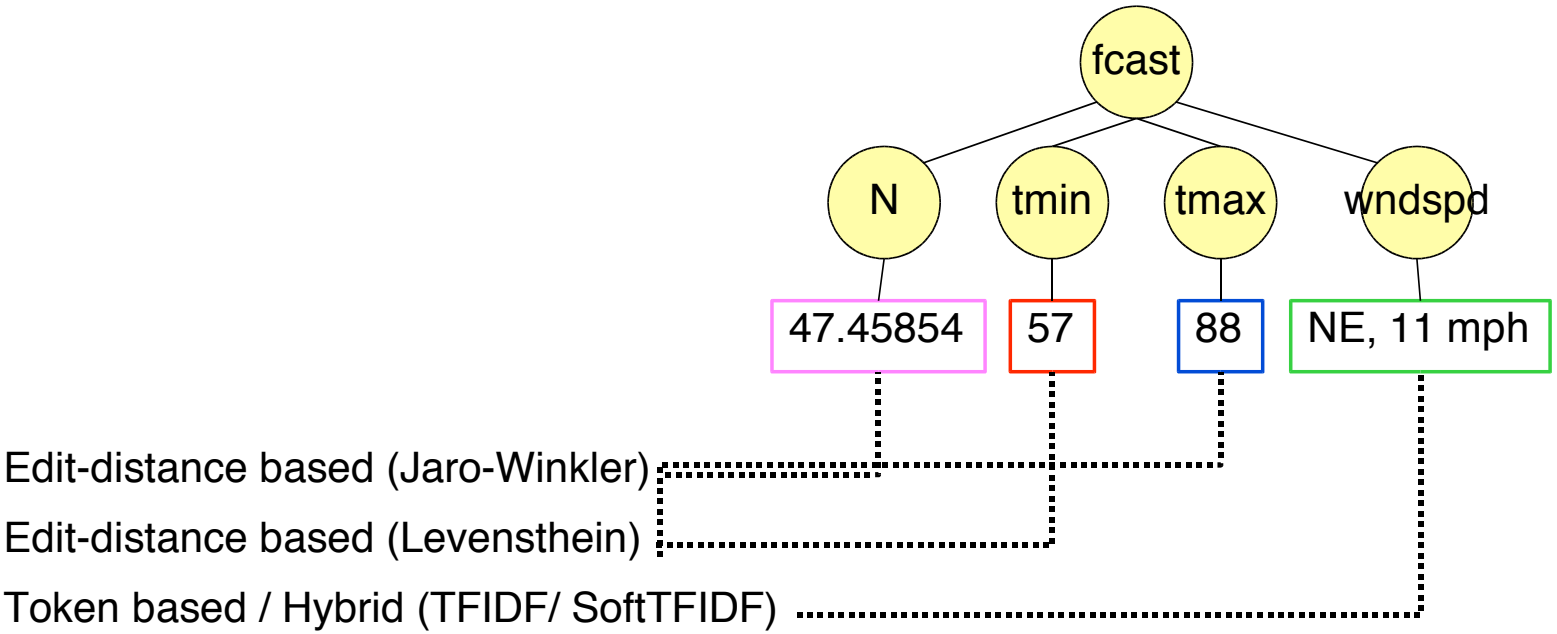
OATS

- Output only
- Uses instance data
- Aggregation

Active Probing & Matching Outputs

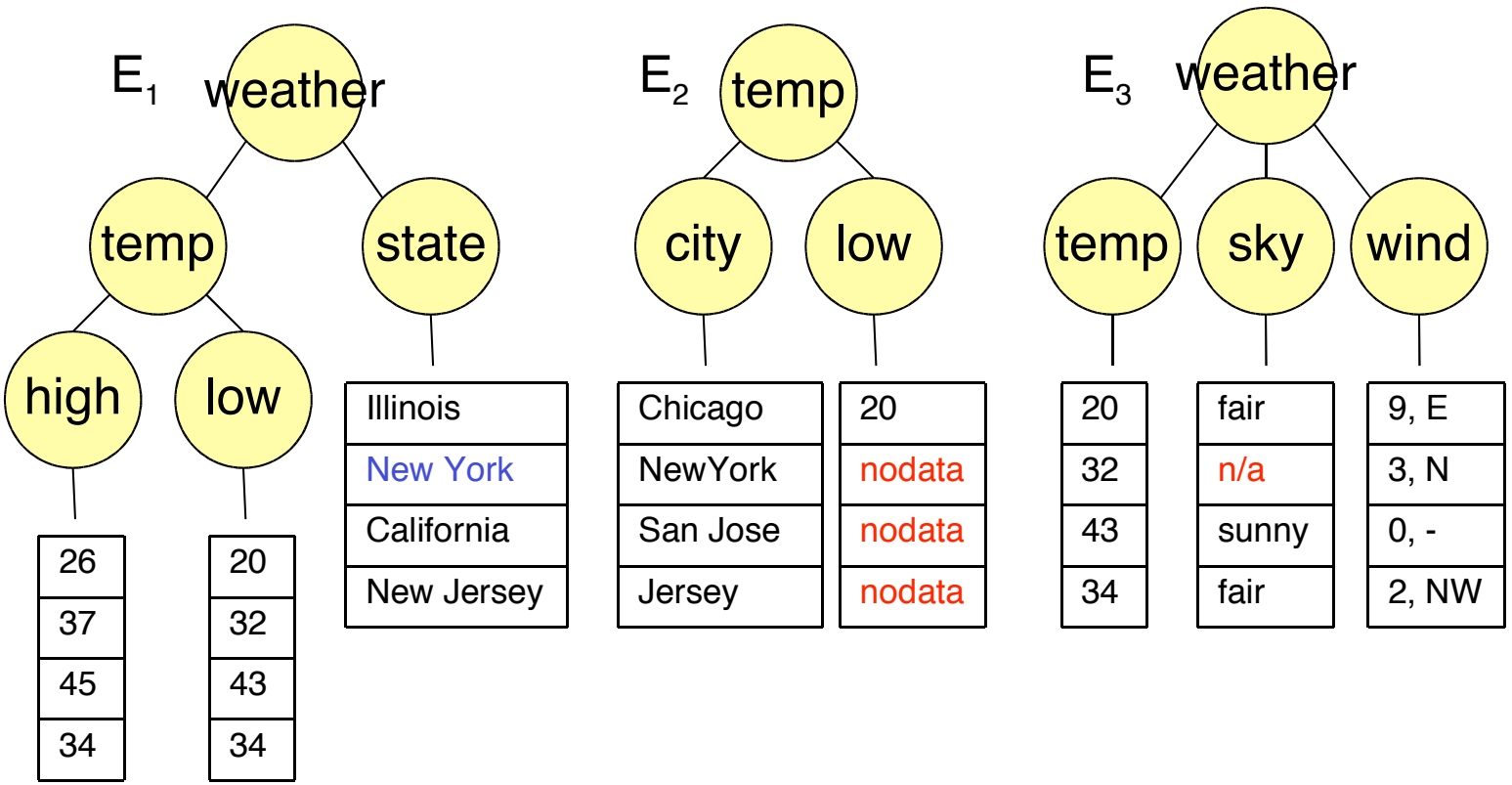


Ensemble of String Distance Metrics



Multiple Probes

Multiple probes minimize the effect of spurious data



OATS – Algorithm

The OATS Algorithm

- Compute pairwise distance between outputs
- Perform HAC clustering
- Match closest pairs

OATS – Advanced

Advanded Features

- Learning weights for ensemble

For more details...

- ... see Eddie Johnston, EKAW 2004

OATS – Advanced

Advanded Features

- Learning weights for ensemble

For more details...

- ... see Eddie Johnston, EKAW 2004

Outline

1. Introduction
   - Motivation
   - Progress made
2. ASSAM Annotator
   - The Annotator Application
   - Relational Learning
   - Evaluation
3. OATS
4. Conclusion

Summary

We have presented:

- Relational Ensemble Learning for Web Service Annotation
- ASSAM Annotator Application
- OATS Algorithm for output aggregation

The Bigger Picture

ASSAM's Limitations

- Supervised Learning: Need training data

Pieces in the Puzzle

- Create and/or select domain ontologies
- Process composition
- ...

Discussion

Thank You for Your Attention

- Questions?

URLs

- http://moguntia.ucd.ie/projects/annotator/
- http://smi.ucd.ie/RSWS

Demo

Taste ASSAM in the Demo session: 17.00-18.30, Rooms 5 & 6