Supervised and Unsupervised Ensemble Learning for the Semantic Web

PhD Viva

Andreas Heß
University College Dublin, Ireland
Supervisor: Nicholas Kushmerick

December 19, 2005
Outline Thesis

- Semantic Web Services
- Ontology Alignment
- Triskel: Ensembles of Biased Classifiers
- Conclusion

Outline Thesis

- Multi-View Learning (chapter 3)
- Relational Learning (chapter 5)
- Triskel (chapter 7)
- Clustering (chapter 4)
- Ontology Mapping (chapter 6)

Web Services

Supervised

Unsupervised

Relational

Andreas Heß
Ensemble Learning for the Semantic Web
Outline Thesis

- Multi-View Learning (chapter 3)
- Relational Learning (chapter 5)
- Triskel (chapter 7)
- Clustering (chapter 4)
- Ontology Mapping (chapter 6)

covered
shortly covered
Outline

1. Semantic Web Services
   - Motivation
   - Background: Relational Learning
   - Iterative Algorithms for Relational Classification
   - Evaluation

2. Ontology Alignment

3. Triskel: Ensembles of Biased Classifiers
   - Background
   - The Triskel Algorithm
   - Evaluation

4. Conclusion
Outline

1. **Semantic Web Services**
   - Motivation
   - Background: Relational Learning
   - Iterative Algorithms for Relational Classification
   - Evaluation

2. **Ontology Alignment**

3. **Triskel: Ensembles of Biased Classifiers**
   - Background
   - The Triskel Algorithm
   - Evaluation

4. **Conclusion**
Web Services

- Web-accessible software
- XML (SOAP) over HTTP
- Just RPC? Forms?
- Data Integration?
Semantic Web Services

Desired Features
- Automatic discovery
- Automatic composition
- Automatic invocation
Scenario: Buying a book

- Congo
  - author
  - title
  - quantity

- WindingStair
  - authName
  - bookT
  - ISBN

- Teatime
  - region
  - qlty
  - qty
Simple Scenario

Global Ontology

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

Congo
- author
- title
- quantity

WindingStair
- authName
- bookT
- ISBN

Teatime
- region
- qty
Simple Scenario

- **Global Ontology**
  - Item
    - Quantity
    - Price
  - Book
    - Author
    - Title
    - ISBN
  - Tea
    - Region
    - Quality

- **Semantic Metadata**
  - (handcrafted)

- **Congo**
  - author
  - title
  - quantity

- **WindingStair**
  - authName
  - bookT
  - ISBN

- **Teatime**
  - region
  - qty
  - qty
Simple Scenario

Global Ontology

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

Congo
- author
- title
- quantity

WindingStair
- authName
- bookT
- ISBN

Teatime
- region
- qty
# Semantic Metadata

## Assumptions

- Semantic annotation
- Shared ontology

## Problem

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

## Our Solution

- Learn mappings from text to ontology
Semantic Metadata

Assumptions

- Semantic annotation
- Shared ontology

Problem

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

Our Solution

- Learn mappings from text to ontology
Semantic Metadata

Assumptions
- Semantic annotation
- Shared ontology

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

Our Solution
- Learn mappings from text to ontology
Now for Something Completely Different

1. **Semantic Web Services**
   - Motivation
   - Background: Relational Learning
   - Iterative Algorithms for Relational Classification
   - Evaluation

2. **Ontology Alignment**

3. **Triskel: Ensembles of Biased Classifiers**
   - Background
   - The Triskel Algorithm
   - Evaluation

4. **Conclusion**
Relational Learning

Relational Data
- consists of objects and relations between objects
- can be represented as a graphs

Methods for relational learning
- Iterative algorithms
- Statistical methods
Relations: Example

Category | Domain | Datatype
------- | ------ | -------
Books   | Query book price | Book title
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

JAMES JOYCE
ULYSSES
Relations: Example

Category

Domain

Datatype

Query book price

Book title

Motivation
Background: Relational Learning
Iterative Algorithms for Relational Classification
Evaluation
Relations: Example

Category

Domain

Datatype

Books

Query book price

Book title
Two Views

---

**Intrinsic View**
- Features inherent to instance
- e.g. text from web page

**Extrinsic View**
- Relations between instances
- e.g. class labels of linked web pages

(Following Neville and Jensen)
Iterative Algorithm for Relational Classification

- 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

![Diagram of iterative ensemble algorithm](image)
Specialised Classifiers

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

```
intrinsic

predictions

select

predictions

2
```

Andreas Heß

Ensemble Learning for the Semantic Web
Specialised Classifiers: Example

intrinsic

predictions

select

2

intrinsic
Specialised Classifiers

- Useful if extrinsic view not sufficient on its own
- Extrinsic features serve as selector
- Idea can be applied in various ways
- For example, see Triskel (next part of the talk)
Our Dataset

- 164 Web Services in 22 Categories
- 1138 Operations in 136 Domains
- 5452 Parameters with 312 Datatypes
- Available on the web
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 omitted: always worse than baseline
Evaluation: Datatype of Parameters

The graph shows the accuracy percentage for different tolerance values, with three types of models: Omniscient, Specialised, and Baseline. The accuracy increases as the tolerance increases for all models. The Omniscient model shows the highest accuracy, followed by the Specialised model, and then the Baseline model.
Evaluation: Domain of Operations

Accuracy %

Tolerance

Omniscient
Iterative Ensemble
Baseline

Accuracy %

Tolerance

Omniscient
Iterative Ensemble
Baseline

Andreas Heß
Ensemble Learning for the Semantic Web
Evaluation: Category of Service

The graph shows the accuracy % plotted against tolerance for different methods:
- Omniscient
- Iterative Ensemble
- Baseline

The accuracy increases as the tolerance increases for all methods.
Evaluation: Category of Service

Why no improvement on category?

- Accuracy of intrinsic classifier too good
- Nothing gained from using extrinsic features
Outline

1. Semantic Web Services
   - Motivation
   - Background: Relational Learning
   - Iterative Algorithms for Relational Classification
   - Evaluation

2. Ontology Alignment

3. Triskel: Ensembles of Biased Classifiers
   - Background
   - The Triskel Algorithm
   - Evaluation

4. Conclusion
Motivation

Recall: Web Services
- Motivation behind Web Services: Data Integration
- Data Integration Problem much more general

Recall: Shared Ontology
- Problem with Semantic Web Services: Global/Shared Ontology
- Web Service Annotation = special form of ontology mapping
The Algorithm

General Schema Mapping Algorithm
- Calculate pairwise similarities between concepts
- Create mappings

Iterative Similarity Calculation
- Compute intrinsic similarity
- Iteratively compute extrinsic similarity
Differences from Prior Work

- Use standard string-distance metrics
- Use vector representation for extrinsic similarity
- Use maximum-weighted matching for final mapping
- Ability to use training data \(^1\)

\(^1\)work done after completion of PhD thesis
Evaluation

Andreas Heß  Ensemble Learning for the Semantic Web
Outline

1. Semantic Web Services
   - Motivation
   - Background: Relational Learning
   - Iterative Algorithms for Relational Classification
   - Evaluation

2. Ontology Alignment

3. Triskel: Ensembles of Biased Classifiers
   - Background
   - The Triskel Algorithm
   - Evaluation

4. Conclusion
Motivation

- Specialised classifiers successful in relational learning
- Idea: Use specialised classifiers for non-relational tasks
- Classifiers specialise on regions of the input space
Related Work: Covering Algorithms

**To train**
- Learn rule that covers training examples
- Remove covered instances and continue

**To classify**
- Find rule that covers instance
Ensemble Learning: General Idea

To train

- Learn *diverse* classifiers on same data

To classify

- Use weighted vote: \( p = \text{sgn}(\sum w_i p_i) \)
Related Work: Boosting

To train

- Train base classifier
- Evaluate classifier on training set
- Misclassified instances get higher weight
- Iterate

To classify

- Use weighted vote
Motivational Example

- Single linear classifier doesn’t work
- Boosting doesn’t work here
- Yet, correct hypothesis is linear combination of linear classifiers
Problems with Boosting on this Dataset

- Boosting: Weights on hard instances are increased for next iteration
- \textit{Hard} instances are defined as misclassified instances
- Need other definition for hard instances!
Easy and hard instances

- Easy instances: Perfectly separable from the rest
- Remove easy instances and continue (as in covering)
- Iterate (as in boosting)
Biased Classifiers

- Identify easy instances by **biasing** towards either class
- Biasing is done by resampling
Triskel Algorithm

To train
- Learn biased classifier for each class
- Learn *arbiter*, where biased classifiers disagree (hard instances)
- Arbiter can be another Triskel (iterate) or simple classifier

To classify
- Ask biased classifiers
- If biased classifiers disagree, ask arbiter
Experimental Setup: Datasets

- Several multiclass datasets from UCI repository
- SMO handles only binary datasets
- Use 1-vs-1 scheme
- For Triskel, can have compromise:
  - Use 1-vs-all biased classifiers and 1-vs-1 arbiter ("Triskel M")
Experimental Setup

- Base classifier SVM/SMO
- Use Triskel with M1, 1, 2, 4 rounds
- Compare against AdaBoost with 3, 10, 50 rounds
Evaluation: Autos Dataset

![Graph showing the comparison of accuracy and training time for SMO, Triskel, and AdaBoost algorithms on the Autos dataset.](image-url)

- **Y-axis:** Accuracy (autos)
- **X-axis:** Training time (seconds)

### Algorithms Compared:
- SMO
- Triskel
- AdaBoost

The graph illustrates the performance of these algorithms in terms of accuracy and training time. SMO, Triskel, and AdaBoost show different trade-offs between accuracy and training time.
Evaluation: Hypothyroid Dataset

![Graph showing evaluation results for Hypothyroid Dataset.](attachment:image.png)
Evaluation: Segment Dataset

The graph illustrates the performance of different algorithms over the segment dataset in terms of accuracy and training time. The x-axis represents the training time in seconds, while the y-axis shows the accuracy. The algorithms compared are SMO (blue), Triskel (red), and AdaBoost (green).

- **SMO** (Support Vector Machines Optimization Software): This algorithm shows a consistent increase in accuracy with increasing training time, achieving high accuracy even at lower training times.
- **Triskel**: This ensemble of biased classifiers demonstrates a steady improvement in accuracy as the training time increases.
- **AdaBoost**: Although AdaBoost starts with lower accuracy compared to SMO and Triskel, it shows a promising trend of catching up with increasing training time.

The data suggests that both SMO and Triskel perform better in terms of accuracy, with SMO initially leading but Triskel closing the gap as training time increases. AdaBoost, on the other hand, starts behind but gains on the other algorithms as training progresses.
Evaluation: Vehicle Dataset

![Graph showing accuracy vs training time for different algorithms.](image-url)
Observations

- Most of the time curve for Triskel above Boosting curve
- Indicates Triskel has better tradeoff between time and accuracy
- Same accuracy as boosting reached faster
- In same time achieve better accuracy
More observations

- On some datasets, Triskel-M can be faster than base classifier without loss of accuracy
- Endpoint of Triskel curve usually above Boosting curve
- Indicates Triskel can often yield better accuracy than Boosting, given enough time
Outline

1. Semantic Web Services
   - Motivation
   - Background: Relational Learning
   - Iterative Algorithms for Relational Classification
   - Evaluation

2. Ontology Alignment

3. Triskel: Ensembles of Biased Classifiers
   - Background
   - The Triskel Algorithm
   - Evaluation

4. Conclusion
Summary

I have presented...
- An iterative ensemble method for relational learning
- Specialised classifiers for relational learning

Evaluation has shown...
- If features high-dimensional: ensemble better than single classifier
- If extrinsic view alone not sufficient: specialised classifiers
Summary

I have presented...

- An iterative ensemble method for relational learning
- Specialised classifiers for relational learning

Evaluation has shown...

- If features high-dimensional: ensemble better than single classifier
- If extrinsic view alone not sufficient: specialised classifiers
I have presented...

- A novel ontology mapping algorithm

Evaluation has shown...

- The algorithm performs well compared to others
Summary

I have presented...
- A novel ontology mapping algorithm

Evaluation has shown...
- The algorithm performs well compared to others
I have presented...

- A novel ensemble learning algorithm with biased classifiers called Triskel

Evaluation has shown...

- Triskel outperforms Boosting in terms of time and accuracy
I have presented...

- A novel ensemble learning algorithm with biased classifiers called Triskel

Evaluation has shown...

- Triskel outperforms Boosting in terms of time and accuracy
Thank You for Your Attention

Questions?
Outline

5 HyperPipes

6 Relational Learning for Web Services
   • Mapping Web Services to Relational Learning
   • ASSAM Annotator

7 Extrinsic Feature Vector
Outline

5 HyperPipes

6 Relational Learning for Web Services
  - Mapping Web Services to Relational Learning
  - ASSAM Annotator

7 Extrinsic Feature Vector
“For each category a HyperPipe is constructed that contains all points of that category (essentially records the attribute bounds observed for each category). Test instances are classified according to the category that most contains the instance.” (sic)
Outline

5 HyperPipes

6 Relational Learning for Web Services
   - Mapping Web Services to Relational Learning
   - ASSAM Annotator

7 Extrinsic Feature Vector
Relational Learning

Relational Data
- consists of objects and relations between objects
- can be represented as a graph

Three Types of Learning Tasks (following Slattery)
- classify nodes
- classify graphs
- classify subgraphs
Relational Learning

Relational Data
- consists of objects and relations between objects
- can be represented as a graph

Three Types of Learning Tasks (following Slattery)
- classify nodes
- classify graphs
- classify subgraphs
Mapping Web Services to Relational Learning

Learn from:
- Web Service as a whole
- Operations
- Parameters

Features used:
- Names
- Comments (if available)
Classifying Datatypes

Task
- Learn labels for parameters

Instances
Classifying Operations

Task

- Learn labels for operations

Instances
Classifying Web Services

Task

- Learn labels for web service

Instances
The ASSAM Annotator

Andreas Heß

Ensemble Learning for the Semantic Web
The ASSAM Annotator

- **Category ontology**
- **Domain/Datatype ontologies**
- **List of web services**
- **Documentation found in WSDL**
- **Plain WSDL view**
- **Tree view on service**
The ASSAM Annotator

Recommended annotations
OWL-S Export

```xml
<owl:Class rdf:ID="WeatherFetcher">
  <rdfs:subClassOf rdf:resource="#Weather"/>
</owl:Class>

<owl:DatatypeProperty rdf:ID="Weather"/>
<rdfs:subPropertyOf rdf:resource="http://moguntia.ucd.ie/owl/Datatypes.owl#Temperature"/>
</owl:DatatypeProperty>
</owl:DatatypeProperty>
```
ASSAM generates...

- Grounding
- Profile
- Process Model
- Concepts

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

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Domain</th>
<th>Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description of service as a whole</td>
<td>Purpose of single operation</td>
<td>Meaning of single parameter</td>
</tr>
<tr>
<td>e.g. Weather, Finance, Books</td>
<td>e.g. Query Price, Purchase Book</td>
<td>e.g. Title, Author’s Name</td>
</tr>
</tbody>
</table>

- **Category**: Description of service as a whole. Examples: Weather, Finance, Books.
- **Domain**: Purpose of single operation. Examples: Query Price, Purchase Book.
- **Datatype**: Meaning of single parameter. Examples: Title, Author’s Name.
Definitions: What? Why?

No new standard!

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

5 HyperPipes

6 Relational Learning for Web Services
   - Mapping Web Services to Relational Learning
   - ASSAM Annotator

7 Extrinsic Feature Vector
Extrinsic Feature Vector

Andreas Heß  
Ensemble Learning for the Semantic Web