Constraint-driven Evaluation in UIMA Ruta

Estimating the quality of arbitrary models on unseen documents.

Andreas Wittek, Martin Toepfer, Georg Fette, Peter Klügl, Frank Puppe
Constraint-driven Evaluation

Performance: F1 (A)?
Constraint-driven Evaluation

Use Expectations:

• Case Study: Segmentation of References (Extracting BibTeX fields)
  • References should have a title containing more than 2 words
  • References should have 1! author field
  • „VHDL“ should be labeled as „title“

Performance:
F1 (A) ?
Good reasons for Constrained-driven Evaluation (CDE)

Choose one of several 3rd-party components, Debug rule-based systems, Decision support for make-or-buy
Reason 1: Development Support

- typical scenario in NLP applications:

- Constrained-driven evaluation (CDE) provides:
  - Ranking of unlabeled documents according to the expected quality of a component
  - F1 score prediction on unlabeled data sets
Reason 2: Component Selection

• typical scenario in NLP applications:
  – several 3rd-party components
    (e.g., CRF & SVM & Rule-based NER model trained on CoNLL)
  – Unlabeled data-set
    (e.g., a custom newspaper corpus)
  – Which component is the best (relatively)?

• Constrained-driven evaluation (CDE):
  – Ranking of components according to performance estimation of arbitrary models using constraints
Reason 3: make-or-buy

- **typical scenario in NLP applications:**
  - 3rd-party component
    (e.g., CRF NER model trained on CoNLL)
  - Unlabeled data-set
    (e.g., a custom newspaper corpus)
  - Are the results good enough for my application? Or do I have to develop my own system? (absolute quality estimation)

- **Constrained-driven evaluation (CDE):**
  - Automated performance estimation of arbitrary models with formalized expectations
CDE Tasks

- **F1 Prediction** task: Estimate F1-score of arbitrary models on unlabeled data
  - Reason 2 (component selection): *relative* performance
  - Reason 3 (make-or-buy) *absolute* performance
  - [Reason 1 (development support) ]

- **Ranking** task: Rank documents wrt. F1
  - Reason 1 (development support)
CDE Workflow

• Basic operations for rapid component development:

1. Gather constraints
   • Rule Constraints: formalize background-knowledge of domain experts
   • Annotation distribution (AD) Constraints: Collect statistics (using large data-bases)

2. Constraint-driven Evaluation
   • Apply model on unlabeled data-set
   • Apply constraints on unlabeled data-set
   • Compute aggregate CDE score for each document
CDE Score

• reflects how well the output of the component complies with the constraints:

\[
\text{CDE} = \frac{1}{\alpha} \sum_{i=1}^{n} w_i C_i
\]

where \( w_i \) is a weight, \( \alpha \) is a normalizing constant, and \( C_i \in [0,1] \) is either

− a *rule constraint score*:
  \#rule matched / \#rule has tried to match

− or an *annotation distribution (AD) constraint score*:
  cosine similarity between expected and observed frequency
Rule Constraints

• Formalize expectations of domain experts as UIMA Ruta rules

• Example:


Author (Title | Year);
Author {-CONTAINS(NUM)};
Title {CONTAINS(W,2,200)};
Rule Constraints

- **Title `{CONTAINS(W,2,200)}`;**
  \[ \rightarrow \text{score: } 4/10 \] (4 out of 10 titles have \( \geq 2 \) words)

Some errors are detected correctly by this Rule Constraint;
Some errors are not detected

AD constraints may help in such cases (see next slide)
Annotation Distribution Constraints

- Statistical assumptions gathered from databases, e.g., Bibsonomy dump

- Example: in DB:
  "VHDL": Author 0.001, Title 0.4, ...

- in Document:
  "VHDL"/Author: 1/1

\[ C_i = \cos \left( (1.0), (0.001) \right) = 0.51 \]
Constrained-driven Evaluation

• **Extended workflow:**
  – Gather constraints
  – ...  
  – Constraint Development
  – Create a small dev.-set
  – Test constraints; specify weights

– Constraint-driven Evaluation
  – ...
Constraint Development

- measures to compare CDE score against ground-truth F1-score/ranking on a set of labeled documents:
  - Spearman (ranking)
  - Pearson (linear dependency)
Experimental Study

Segmentation of References of Scientific Papers
Setting

- \(D_{\text{ruta}}\): used to develop rules for reference segmentation (219 references in 8 docs)
- \(D_{\text{dev}}\): labeled by the constructed rules; used for developing the constraints (192 references in 8 docs)
- \(D_{\text{test}}\): labeled by the constructed rules; used to evaluate the constraints (155 references in 7 documents)
- \(D_{\text{crf}}\): labeled by training and applying CRFs (5-fold-cross) (ruta+dev+test: 566 references)
- \(D_{\text{gen}}\): different source; unknown style guides; labeled by the constructed rules (452 references in 28 documents)
Setting

• Three constraint sets:
  1. $C_{\text{ruta}}$:
     15 Rule constraints (Author, Title, Date); weight of each constraint is 1
  2. $C_{\text{ruta+bib}}$:
     $C_{\text{ruta}}$ + AD constraints (entity distribution of words extracted from Bibsonomy); weight of each constraint is set to 1
  3. $C_{\text{ruta+5*bib}}$:
     weight of each AD constraint set to 5
Results

Table 3. Spearman’s $\rho$ and Pearson’s $r$ given for the predicted CDE score (for each document) compared to the actual $F_1$ score.

| Dataset | $C_{ruta}$ | | $C_{ruta+bib}$ | | $C_{ruta+5xbib}$ | |
|---------|------------|------------|------------|------------|------------| |
|         | $\rho$     | $r$        | $\rho$     | $r$        | $\rho$     | $r$        | |
| $D_{dev}$ | 0.8708    | 0.9306    | 0.9271    | 0.9405    | 0.8051    | 0.6646    | |
| $D_{test}$ | 0.9615    | 0.9478    | 0.9266    | 0.8754    | 0.8154    | 0.6758    | |
| $D_{crf}$  | 0.6793    | 0.7881    | 0.7429    | 0.8011    | 0.7117    | 0.7617    | |
| $D_{gen}$  | 0.7089    | 0.8002    | 0.7724    | 0.8811    | 0.8150    | 0.9504    | |

- Strong correlation ($\rho \geq 0.6$, $r \geq 0.6$ for all $D_i, C_j$):
  - constraint rules really estimate the performance of the models on the new unlabeled data
- Even with different data (new styles):
  - use AD constraints in this case
UIMA Ruta CDE Plugin
Constrained-driven Evaluation in the UIMA Ruta Workbench
UIMA Ruta CDE Plug-in
- Constraint Specification -
UIMA Ruta CDE Plug-in
- Constraint Development & Results -
UIMA Ruta CDE Plug-in
- Document Details -
Summary: Constraint-driven Evaluation in UIMA Ruta

Predict performance of arbitrary models on unseen data by constraints

- Rule-constraints
- Annotation Distribution constraints

UIMA Ruta workbench plug-in

- Supports rapid prototyping
- and component selection

Experimental study on reference segmentation shows usefulness
Thank you for your attention!

Try CDE in UIMA Ruta 2.1.0:
http://uima.apache.org/ruta.html
References