Abstract

• Information Extraction (IE) can be viewed as a machine learning problem.
• Use a relational learning algorithm (SRV): make no assumptions about document structure and kinds of information available; instead, uses a token-oriented feature set.
• Compare results of a standard memorization agent to SRV in extracting data from university course and research web pages.

SRV

• Top down relational algorithm.
• Token-oriented features, easy to implement and add to the system.
• Domain-specific features are separate from the core system – easier to port to other domains.
• Easy to add to the basic feature set for different formatted text, i.e. HTML.

Terminology

• Title – field
• Actual title – instantiation or instance
  – A page may have multiple fields of interest and multiple instantiations of each field.
• In traditional IE terms, a collection of tasks is called a template. Each field is a slot, and each instantiation a slot fill.

Main Problem

• IE involves many sub-tasks, including syntactic and semantic pre-processing, slot filling, etc.
• SRV was developed to solve just the slot filling problem.
• Find the best unbroken fragment(s) of text to fill a given slot in the answer template.

Extraction = Text Classification

• Every candidate instance is presented to a classifier which is asked to "rate" each as likely project members.

Relational Learning

• RL, or inductive logic programming, suited to domains with rich relational structure.
• Instance attributes are not isolated, but related to each other logically.
• Predicates based on attribute values are greedily added to a rule under construction.
• Can logically derive new attributes from existing ones.
• DATA USED FOR LEARNING IS IMPORTANT!
SRV
• Differs from previous systems by learning over an explicit set of simple and relational features.
  – Simple maps a token to a discrete value.
  – Relational maps a token to another token.

SRV – Predefined Predicate Types
• length(Relop N) Relop: > < =
  – length(= 2) accepts fragments with 2 tokens.
• some(Var Path Feat Value)
  – some(?A {} capitalizedp true) means: “the fragment contains some capitalized token.”
  – some(?A [prev_token prev_token] capitalized true) means: “The fragment contains a token where two tokens before was capitalized.”

SRV – More Predicate Types
• every(Feat Value)
  – every(numericp false) means: “every token in the fragment is non-numeric.”
• position(Var From Relop N)
  – position(?A fromfirst < 2) means: “the token ?A is either first or second in the fragment.”
• relpos(Var1 Var2 Relop N)
  – relpos(?A ?B > 1) means: “the token ?B is more than one token away from ?A.”

Validation
• Each 1/3 of the training data is set aside one at a time to “validate” the rules generated from the other 2/3 against.
• # of total matches and correct matches is stored with each validated rule set.
• The 3 rule sets generated are concatenated, all that match a given fragment are used to generate the confidence score in the extracted data.

Adapting SRV for HTML
• Easier than previous methods, because SRV has “core” features and is extensible.
• Simply add features to deal with different domains.

Table 1: SRV-core token features. The two features in bold face are relational features.

Table 2: HTML features added to the core feature set. Features in bold face are relational.
HTML Testing (SRV)

- 105 class web pages and 96 research project pages from Cornell, University of Texas, University of Washington and University of Wisconsin.
- Tested twice using random partitioning and LOUO (leave one university out).

Results (One-Per-Document)

The system only has to return one prediction despite having more than one result – just return the prediction with the highest confidence.

Results (Many Per Document)

More difficult task, as when multiple possibilities are found, you have to return all of them.

Baseline Approach Comparison

A simple memorizing agent

Some Conclusions

- SRV performs better than a simple machine learning agent in all cases.
- When HTML features added, it performs much better; though note that these features are NOT NECESSARY for SRV to function.
- Random partitioning works better, probably because each university in the test data had certain web page formatting similarities within the university, but not between them.

Some Conclusions

- Recall/accuracy and precision/coverage can be "tweaked" by throwing out rules with confidence less than a certain x%, so that a system can be tuned to a particular application.
- Domain specific information is separate from the learning mechanism (allows adaptation to other domains).
- Top-down induction - general to specific, rather than previous bottom-up models which relied on heuristics, implicit features to trim constraints.