Timeline

• April 6: Java API, documentation, code stubs due. Coding begun.
  – First drafts circulated by Monday April 3
  – Including spec for RTW architecture which will use them
  – Form central registry of labels, their meanings, their sources

• April 13: Working project code (standalone) due
  – and integrated system based on code stubs

• April 20: Integrated system based on project code
  – Shortcomings identified

• April 27: Experiment with integrated code, and extensions

• May 4: Final system evaluation

• May 11: Final writeups due, in form to integrate into single unified RtW project report.
Task Threads

- Module development
  - Weeks 1-5
- Module-level evaluation
- Architecture development
- System integration
- End-to-end System Evaluation
- Final write-ups (modules)
- Final project report for the class
Action Items

• Tabulate steps in your module’s operation (as per Andy S’s example) by Monday
• Document the steps (on Monday or ASAP)
  – Text description, i/o, assumptions, constraints, etc.
  – Data flow diagram (reading/writing from disk, ADB, etc.)
• Domain model / type system
  – EHN distribute GALE type system
  – Set up submeeting to discuss further
    Jon, Jaime, Andy, Kevin, Justin, Nguyen, Laura, Ben, Scott
• EHN draft an overall vision document (Monday)
  – Global version of the tabulation of steps
APIs Posted to Kiva Site

- sfung: Coreferent Resolution API
- hazen: Coreference & Data Flow
- acarlson: Active Learning API
- belamber: Scone API
- jarguell: Relation Extraction API
- rcwang: Entity Association API
- jbetteri: Nominalization SRL initial API / Contribution Spec
## The Big Picture

<table>
<thead>
<tr>
<th>Module</th>
<th>LEARNING Input</th>
<th>LEARNING Output</th>
<th>RUN-TIME Input</th>
<th>RUN-TIME Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coreferent Resolution</td>
<td>Person names, web pages containing the name, profile info</td>
<td>Clustered profile vectors for each web page and name pair</td>
<td>Person name, web page containing the name</td>
<td>Information profiles for that name; best profile match</td>
</tr>
<tr>
<td>Coreference</td>
<td>Gold-standard data (text and annotations), features, unlabelled data</td>
<td>Classifiers &amp; models</td>
<td>Document</td>
<td>Document with referent and antecedent annotations</td>
</tr>
<tr>
<td>Active Learning</td>
<td>Active Learning problem elements conforming to API</td>
<td>Active Learning outcomes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Scone API</td>
<td>Store facts that are learned by any module</td>
<td>Updates to shared KB</td>
<td>API calls to query the knowledge base</td>
<td>Relevant facts, relations, etc.</td>
</tr>
<tr>
<td>Relation Extraction</td>
<td>Relations &amp; seeds; INDRI index with NE &amp; SENTENCE annotations</td>
<td>Trained model (entity pairs and extraction rules) w/confidence for each relation</td>
<td>A trained model; an INDRI index; a file to annotate</td>
<td>For each relation, the most confident entity pairs that participate in the relation</td>
</tr>
<tr>
<td>Entity Association</td>
<td>Seed inputs, types</td>
<td>Entity co-occurrence graph, ranked entities for each type</td>
<td>Input string, entity type</td>
<td>Ranked entities of that type which co-occur with the input</td>
</tr>
<tr>
<td>Nominalized SRL</td>
<td>Text annotated with NE, NP, VP, PP, &amp; SRL information</td>
<td>Trained SRL model for training data</td>
<td>Document</td>
<td>Semantic role labels (annotations)</td>
</tr>
</tbody>
</table>
Levels of Representation in RTW

• Mention level: root text
• Instance level: instance of a KB fact, concept, event, attribute, etc. (e.g. “PERSON_1234”)
• Concept level: abstract class, relation, etc. in the KB (e.g. “PERSON”)
• Recognition_1: linking spans/tags at the mention level to tags at the instance level
• Reference resolution_1: linking mentions that refer to the same instance
• Etc.
Another form of recognition

• Mentions lead to a local hypothesis of an entity; use EQ links on concepts
• Scone can represent multiple entity hypotheses and weights, with links to possible concepts
• Could be context-specific
• Outputs of name profile module (Nguyen and Simon)
Starting from the raw text

• Based on text & grammatical structure:
  – Annotate “referring expressions”
  – Link RE annotations with “refers_to” attribute

• Based on meaning of the text
  – Creating clusters of attributes
  – Checking clusters in Scone
  – Finding most likely referent for e.g. “Jim”
Egypt imports Scud-C missiles. It also possesses 50 Scud-Bs.
Heuristic Reference Resolution

- Identify predicate with anaphor in argument
- Search for “earlier” predicates that have potential antecedents
- Use Scone to sanity-check possible antecedents
Possible Approach
API Calls at Layer Boundaries

Store Mention (Retract Mention);

Create=Find Instance (Retract Mention);

Store Instance (Retract Mention, Concept)
Danzin & General

Danzin

Test

Some Labels

Ace Labels

"Intensive"

Person

"Mental"

Level

Person-Mental

"Text"

Span, POS, NP

Assert
Action Items

• Get input/output annotations (initial set plus examples) from each team
• Jon provide ACE annotation labels (Kiva)
• EHN provide ENAMEX scheme
• Question: can we bootstrap the RTW type system using ACE & current ADB annotation types?
• EHN: provide an end-to-end example showing how to use existing APIs (ADB, M3 & Scone) to decorate a sample text.
ENTITY
SL-REACTION
ENTITY

AR60
P-JACKET
AR61

ENAMED

SPAN

"Egypt bait reel tackle."

SPAN

SPAN

REFERENCE

SCOPE

ASSERT SQL

BBN/INTERACT
(MNC LABEL)
Annotations and Referents

Fred Center (Entity) -> CeoOf (Relation Annotation) -> Center Micros (Entity)

Person: P1 (Entity Annotation)
Organization (Entity Annotation)
Person (Entity Annotation)

.... Fred Center is the CEO of Center Micros. ..... He is a graduate of State University.

Center Micros CEO, Fred Centers... balked at the idea of a take-over...
Sample Type System

- Top
  - String
  - Int
  - Annotation
    - Begin: int
    - End: int
  - ... (other unspecified types)

  - Entity Annotation
    - Begin: int
    - End: int
    - Person
    - Location
    - Gov Official

  - Relation Annotation
    - Arg1: Entity Ant.
    - Arg2: Entity Ant.

  - Gov Title

  - Token

  - Gram. Struc.
    - NP
    - VP
    - PP

  - Located In
    - Arg1: Entity Ant.
    - Arg2: Location
Partial *HUTT* Type System (254 concepts in total)