Information Extraction: Beyond Named Entities

Adam Meyers
New York University
Outline

• What is Information Extraction?
• ACE Entities, Relations and Events
• Timex and TimeML
What is Information Extraction?

• The automatic extraction of information
• Input: possibly limited set of documents
• Output: usually a task-defined template to fill in

Definitions:
– Typically idiosyncratically defined for task
– Can include technology (SRL, etc.) that helps IE

Comparison with Question Answering
– QA more opened ended – depends on questions
– QA: paragraph output vs. IE structured output
– Some IE techniques, e.g., if answer = short phrase
– Some IR techniques, e.g., if answer = paragraph
– Not covering QA in this class
Some Sample IE Tasks

• Extract instances of people starting jobs and ending jobs
  – Identify: person, start or stop time, company
• Extract instances of Entity1 attacking Entity2, where entities include people, GPEs (locs), facilities or vehicles
  – Identify: aggressor, victim, weapon, time
• Extract instances of disease outbreak
  – Identify: victims, disease, start time, time span, location
• Extract advertisements for cameras
  – Identify: seller, brand, model, price, date
• Identify family, social and business relations between individuals
Some ACE History

• Entities: English, Chinese, Arabic, Spanish
• Relations: English, Chinese, Arabic
• Events: English, Chinese, Arabic
• Documentation for various versions of tasks:
• Different years (from 2000 to 2008)
  – Different tasks and subtasks
  – Different versions of specifications
  – We discuss latest available versions of English tasks
Named Entity Review

• Tend to be phrases consisting of proper nouns
  – Capitalization, uniquely identify entity in real world, ...
  – Ex: *The Association for Computational Linguistics*

• Internal structure may differ from common NPs
  – Ex: *Adam L. Meyers, Ph.D.*

• Only certain types are marked
  – Task-specific
  – ACE task: GPE, Person, Organization, Location, Facility
    • In some versions: Vehicle and Weapon
ACE Entities

• An Entity = a list of coreferential NPs
  – Each of these NPs is a “mention” of the entity
  – Finding coreference will be part of a different lecture

• Types of mentions: names, common nouns, pronouns

• Names: what we have been calling named entities

• Nominal mentions: phrases headed by common nouns
  – same semantic classes: GPE, ORGANIZATION, ...
  – EX: that country, the government, the agency, the whimsical pediatrician, the terrorist wearing a hat

• Pronominal mentions: pronouns
  – Must refer to markable semantic class (e.g., by coreference)
  – He, she, it, they, themselves, their, her, everyone, ...
Detecting ACE Entity Mentions

- Detecting ACE name mentions
  - Sequence labeling, typically with BIO tags (Nymbol, HW6, etc.)
- Detecting ACE common noun mentions:
  - Find common nouns from training corpus
  - Generalize
    - Stemming
    - WordNet, clustering, or a list of words
  - Identify non-generic cases
    - *Gardners are lousy plumbers.* [Generic]
    - *The gardner was a lousy plumber.* [Non-Generic]
    - Baseline: definite determiners plus past tense → non-generic
- Pronoun Mention – dependent on coreference techniques
- Coreference Component – described in past lecture
ACE Relations and Events

• Predicate + Arguments
• Annotation of Predicate triggers
  – Event mention triggers: words
    • Specs discuss choice of nouns/verbs: launch an attack
  – Relation mention triggers: grammatical constructions
    • ACE specs refer to these constructions as relation classes
    • ML must learn which words trigger which relations

• Arguments of Event and Relation Mentions
  – Usually, NPs belonging to ACE Entity classes:
    • Named Entities, common noun phrases, pronouns
  – Values – times, extents, crimes, ...
    • https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-values-guidelines-v1.2.4.pdf
  – Relations always take exactly 2 arguments
  – Event arguments vary in number (and a given argument may be absent)
ACE Relations

• Relation Entity: set of coreferential relation mentions
  – Same arguments
  – Refer to same predication

• Relation types
  – Physical: Location and Near
  – Part-Whole: Geographical and Subsidiary
  – Per-Social: Business, Family, Lasting-Personal
  – Org-Affiliation: Employee, Owner, Member, ...
  – Agent-Artifact: User-Owner-Inventor-Manufacturer
  – Gen-Affiliation: Citizen-Resident-Religion-Ethnicity, Org-Location-Origin

• Relation Classes: Syntactic environments (sentence internal only)
  – Verbal, Possessive, PreMod, Coordination, Preposition, Formulaic, Participial, Other
ACE Relation Examples

- **George Bush traveled to France on Thursday for a summit.**
  - Physical.located(*George Bush, France*)
  - Class = Verbal, Modality = Asserted, Tense = Past

- **Microsoft's chief scientist**
  - Org-Aff.employment(*Microsoft's chief scientist, Microsoft*)
  - Class = Possessive, Modality = Asserted, Tense = Unspecified

- **New York police**
  - Part-Whole.Subsidiary(*New York police, New York*)
  - Class = PreMod, Modality = Asserted, Tense = Unspecified

- **Dick Cheney and a hunting partner**
  - Per-Social.Lasting(*Dick Cheney, a hunting partner*)
  - Class = Coordination, Modality = Asserted, Tense = Present

- **A linguist from New York**
  - Gen-Aff.CRRE(*A linguist from New York, New York*)
  - Class = Preposition, Modality = Asserted, Tense = Unspecified
ACE Relation Detection

- General
  - Most Systems use ML and a variety of features
  - 2 Possible Testing environments
    - Entity detection system first and use results
    - Hand-annotated (“true”) entity mentions – separate entity from relation detection

  - [http://www.aclweb.org/anthology/P05-1053](http://www.aclweb.org/anthology/P05-1053)
  - Support Vector Machines – ML algorithm, details omitted
  - Features similar to those used for semantic role labeling:
    - words in arguments, entity types, nearby words, chunking features, parsing features, dependency features, name features from gazetteers, WordNet features ...
  - Observation: Parsing (and dependency) features helped very little
    - Probably because most relations are between nearby words
  - Results: Precision = 63.1, Recall = 49.3, F-score = 55.5
    - F-scores vary by type from 36.4 (Physical.near) to 72.6 (Gen-Aff.CRRE)
ACE Events

- Event Entity: set of coreferential entity mentions
  - Nonconflicting arguments
    - A mention may include a subset of the arguments
  - Refer to same predication (event, state, etc.)

- Event types
  - Life: be-born, marry, divorce, injure, die
  - Movement: transport
  - Transaction: transfer-ownership, transfer-money
  - Business: start-org, end-org, merge-org, declare-bankruptcy
  - Conflict: attack, demonstrate
  - Contact: meet, phone-write
  - Personnel: start-position, end-position, nominate, elect
  - Justice: arrest-jail, release-parole, sue, appeal, pardon, ...
ACE Event Example

- **On Thursday, Pippi sailed the ship from Sweden to the South Seas**
  - ANCHOR = sailed
  - ARTIFACT-ARG = Pippi
  - VEHICLE-ARG = the ship
  - ORIGIN-ARG = Sweden
  - DESTINATION-ARG = the South Seas
  - TIME-ARG = Thursday

- Similar to Semantic Role Labeling, but limited to several Frames
  - Like FrameNet
    - fewer frames
    - annotation-based instead of lexicon based
  - Targeted towards specific tasks (unlike PropBank/NomBank)
ACE Event Detection

• Very few published system descriptions
  – Official ACE scores are hard to understand
    • Much more complex than F-score
    • Includes (subjective) weights based on utility value (e.g., names are weighted higher than common nouns because they carry more info)
  – Task is complex including entity detection, coreference, event coreference, etc.
  – Only for ACE years 2004 (English) and 2005 (Chinese also)
  – Scores tended to be low

• Best performing systems use parsing features
  – Unlike relation extraction, non-local features matter

• Task often broken down into subtasks
  – Identify event anchor, identify arguments, coreference, ...
Example System: Ahn 2006

- http://anthology.aclweb.org/W/W06/W06-0901.pdf
- Maximum Entropy Based System
- Detecting and Classifying Event Anchors:
  - Features: word, regularized (upper/lower, lemma, POS, depth in parse tree, WordNet features, left/right context (case, POS), dependency relations (info about words/relations above and below anchor, path features, etc.)
    - Precision = .735, Recall = .513, F-score = .601
- Argument Identification
  - Features: anchor word (with/without regularization), Event type, argument (determiner, head, POS, class, depth in parse tree, mention type, same info about sibling arguments, dependency path from anchor to argument
    - Precision = .689, Recall = .490, F-score = .573
- Other subtasks: time, +/-generic, modality, polarity
Time

- **Timex**
  - Identifying Absolute Time Expressions
    - Regularization
  - Relative Time Expressions
    - Regularization
    - Relation to document time

- **TimeML** – temporal relations between 2 args
  - Event and Time  [Event \(\approx\) ACE Event Mention]
    - Event is before/after/at/during/.... Time
  - Event1 and Event2
    - Time(Event1) is before/after/at/during/.... Time(Event2)
TIMEX (TIMEX2, TIMEX3, ...)  
• Identifies several types of time expressions in text  
  – Absolute Time (January 3, 2011 at 5:00 AM)  
  – Relative Time (last Thursday)  
  – Duration (5 days)  
• 2 Types of Markup (XML)  
  – Inline:  
    • <TIMEX3 tid="t18" type="DATE" temporalFunction="true" functionInDocument="NONE" value="1990-01-02" anchorTimeID="t17" >Jan. 2</TIMEX3>  
  – Offset: <TIMEX3.... start="2015" end="2021"/>  
    • Other than start and end, all the same features
value in ISO 8861 Standard TIMEX3

- Fills the XML value slot
- Time values: month, day, year, hour, second, quarter, half, week, ...
- Examples:
  - December 14, 2011 at 10:49:01AM → 2011-12-14-T10:49:01
  - 3:49PM → T15:49
  - December 14 → XXXX-12-14
  - A Sunday in November → XXXX-11-SU
  - 2011, 3rd Quarter → 2011-Q3
- Values of relative times are calculated
  - Last Thursday → 2011-12-08 if the publishing date is 12/14/2011
- Values of absolute times are looked up and filled in
  - December 14 → 2011-12-14 (from context, e.g., past tense, before 12/14/2012, ...)
- Duration values: numbers and units
  - 5 months → P5M
  - 5 minutes → P5TM
Timex Systems

• Identifying Time Expressions
  – Manual rules, HMM, etc.

• Encoding values already in the text
  – Manual rules: very small number of terms with clear values –
    simple regular expressions or patterns with look up table

• Calculating values relative to
  – Document Time: publication date (news articles)
  – Other times found in the text [not always implemented]

• Examples for article published Wed, Dec 14, 2011
  – Yesterday→2011-12-13
  – Last Thursday→2011-12-08
  – November 3 →2011-11-03
  • may be 2012 depending on month and modifiers (next, last, ...)
Sample Times Rules from NYU Proteus

• Look at Ralph's JET file: time_rules.yaml
TimeML Relations

• There are several different TimeML Relations
  – **Tlink**: [We will focus on this one]
    • Link between time and event
    • Link between time(event1) and time(event2)
    • Overlaps with Penn Discourse Treebank Relations (PDTB)
      – PDTB
        » PDTB also covers non-temporal relations
        » But only links sentences (verbs), not temporal phrases (NPs)
  – **Slink**:
    • Link between event and event (subordination)
  – **Alink**
    • Link between aspectual marker (start, end, etc.) and event
Arguments of TLink Relations

- **Event** (different than in ACE):
  - Word anchoring something that has a time
  - All verbs (event those that represent states)
    - PDTB uses sentences (phrase vs. dependency representation)
    - For TimeML, coordinated verbs counted separately
  - Some nouns (though not consistently marked)
    - Those that represent states
    - Not in PDTB

- **Time**:
  - Temporal Expression
  - Document Time
  - Time(Event) – only one used in comparable PDTB relations
Tlink Features

• Signal: word or phrase that anchors relation
  – Same as predicate for Penn Discourse Treebank
  – Optional

• RelType: Classification of temporal relation
  – BEFORE, AFTER – before or after
  – INCLUDES, IS_INCLUDED – time spans event
  – DURING – duration
  – SIMULTANEOUS – at same time
  – IBEFORE, IAFTER – Immediately Before/After
  – IDENTITY – same event
  – BEGINS, ENDS, BEGUN_BY, ENDED_BY – marks boundary
Simple Cases: Signals and Modification

- Relation **from** Event Instance (red) **to** Time/Event (white)
  - PDTB: ARG1 = **from**, ARG2 = **to** due to **Signal** (blue)
- Prepositions and subordinate conjunction signals
  - *They left the room after 5 o'clock.* (AFTER)
  - *They left the room while the mayor was announcing the new law.* (During)
- Discourse adverb signal
  - *The mayor announced the law. Simultaneously, they sang the song.* (Simultaneous)
- Modification
  - *The mayor announced it Last Thursday.* (IS_INCLUDED)
Sequences of Simple Tenses

- Two instances of simple past tense
  - *John had a headache. He took two aspirin.* (BEFORE)
  - *The lamp fell. It shattered into a million pieces.* (IBEFORCE)
  - *They ate steak. They drank wine.* (SIMULTANEOUS)
  - *He slept for hours. He dreamed about monsters.* (INCLUDES)

- Two instances of simple present tense
  - *I have a big problem. I have a headache.* (IDENTITY)
  - *The fish swims. The bird flies.* (SIMULTANEOUS)

- Different Tenses
  - *Mary's head hurts. She left school early.* (AFTER)
  - *Mary left school early. Her head hurts.* (BEFORE)
+-Progressive and +/-Perfective

- Progressive: – be + -ing (continuous action)
- Perfective: have + -en (past relative to a reference point)
- Examples:
  - I see a ghost. I am leaving. (IBEFOREx)
  - They are laughing. They see the ghost. (SIMULTANEOUS)
  - He was leaving. He saw a ghost. (IAFTER)
  - They saw a ghost. They were leaving. (SIMULTANEOUS)
  - I am leaving. They have won the game. (AFTER)
  - They have won the game. I am leaving. (BEFORE)
  - She left. She had eaten a sandwich. (AFTER)
  - She had eaten a sandwich. She left. (BEFORE)
  - She left. She had been eating a sandwich. (AFTER)
  - She had been eating a sandwich. She left. (BEFORE)
Vendler's Aspectual Verb Classes

- **States**: *be, know, love, have, own, ..*
- **Process**: *run, eat, fly, …*
  - Process describes all subevents
- **Accomplishment**: *draw a circle, run a race, …*
  - Time period measures entire event duration
- **Achievement**: *won, die, …*
  - Time measures end point
- **Interaction**: aspect classes and aspect
  - Progressive: state → process, process → state, …
- **Vendler, Zeno “Verbs and Times”**
  - Originally published in 1957 in *The Philosophical Review*, but easier to find in Vendler (1967) *Linguistics in Philosophy*
Factors in the Ordering of Events

• Signals
• Sequence of Tenses
• Sequences of Aspect
• Sequences of Aspectual Verb Classes
  – Sense disambiguation-like problem
• Real world knowledge
  – e.g., breaking tends to occur after falling
Manual Rules

• Lexical signals
  – Most common signals (subord conj/preps) easy
  – Others (adverbs) may require a lexicon (manually or automatically created)

• Tense and Aspect Sequences
  – There is some descriptive work
  – General rules may only describe typical cases
    • (Past | Perfective) + Present → Before
    • Present + (Past | Perfective) → After
    • Past + Past-Participle → After [reliable rule]
      – *Mary left. She had eaten her dinner.*
    • Past + Past → Before [not reliable]
      – *Mary left. She ate dinner.*
  – Exception: *The dish broke. It fell.*
Machine Learning

• TimeBank – Annotation for Supervised Methods
• Patterns to Acquire
  – Rare signals → Relation Type
    • Lexical information
    • Ex: whence → SIMULTANEOUS, ...
  – Predicate/Predicate Pairs → Relation Type
    • Modeling real world knowledge
    • Ex: fall/break → BEFORE, …
  – Tense/Aspect Pair Probabilities
    • Past/Past → BEFORE relation with 72% probability
TimeML Systems

- 2010 shared task
  - [http://www.timeml.org/tempeval2/](http://www.timeml.org/tempeval2/)
- Best System Performance for English:
  - Task A (recognition/regularization of timex3)
    - Recall/Precision/F-score – all about 85%
  - Task B (identifying events)
    - Best Recall: 81%, Precision: 86%, F-score: 83%
  - Best F-scores for Relation Tasks
    - Task C (relation betw timex and event in 1 sentence): 63%
    - Task D (relation betw event and document time): 82%
    - Task E (relation betw main events in adjacent sentences): 56%
    - Task F (relation betw superordinate/subordinate events): 60%
Other High Level IE-like Tasks

• Sentiment Analysis
  – Identify differing opinions
  – Detect positive/negative/other views/language
    • Application: Divide reviews into positive and negative
      – Movies, products, politicians, etc.
    • Use IR techniques

• Detect Attribution
  – Whose view does a given sentence represent
  – John said that Mary said …. [Author:John:Mary]

• Factivity
  – Is the statement reported to be true/false/other
  – According to whom
Other Types of Entities to Extract

• Terminology
  – Terms that are specific to particular genres
  – genes, chemicals, species, formulas, ..

• Numeric terms
  – Numbers, Money, Percent

• Commercial
  – Product Names, Brand Names, …
  – ID numbers, …
Summary

• Information Extraction:
  – The automatic extraction of information from text to produce structured output that, e.g., can be put into a database

• Named Entities: classified instances of names

• ACE Relations and Events: predications with entities and other nouns as arguments

• Timex: An NE-like classification for temporal expressions, with missing information filled in.

• TLink: Temporal relation (before, after, etc.) between 2 events
Events and Relations Readings

• J & M Chapters 22.2 to 22.4 (required)
• ACE Relation Guidelines (optional):
• ACE Event Guidelines (optional):
• ACE Relation and Event System papers (read 1 paper)
  – http://www.aclweb.org/anthology/P/P05/P05-1053.pdf
Time Annotation and Documentation

• TimeBank corpus (optional)
  – http://timeml.org/site/timebank/timebank.html
  – TimeBank1.1 Corpus – I may be able to make this available if needed

• A good resource (optional)
  – Mani, Pustovoytsky and Gaizauskas (2005).
    Language of Time: A Reader.
    Oxford University Press.
  • Includes reprint

• Trips/Trio – An Example TimeML system (read):
  – http://www.aclweb.org/anthology/S10-1062