ANAPHORA RESOLUTION

Olga Uryupina
DISI, University of Trento
Anaphora Resolution

Example
Sophia Loren says she will always be grateful to Bono. The actress revealed that the U2 singer helped her calm down when she became scared by a thunderstorm while travelling on a plane.

Coreference Chains:

- \{Sophia Loren, she, the actress, her, she\}
- \{Bono, the U2 singer\}
- \{a thunderstorm\}
- \{a plane\}
Anaphora Resolution

The interpretation of most expressions depends on the context in which they are used

- Studying the semantics & pragmatics of context dependence a crucial aspect of linguistics

Developing methods for interpreting anaphoric expressions useful in many applications

- Information extraction: recognize which expressions are mentions of the same object
- Summarization / segmentation: use entity coherence
- Multimodal interfaces: recognize which objects in the visual scene are being referred to
Outline

- Terminology
- A brief history of anaphora resolution
  - First algorithms: Charniak, Winograd, Wilks
  - Pronouns: Hobbs
  - Salience: S-List, LRC
- The MUC initiative
- Early statistical approaches
  - The mention-pair model
- Modern ML approaches
  - ILP
  - Entity-mention model
  - Work on features
- Evaluation
Anaphora resolution: a specification of the problem

Example
Sophia Loren says she will always be grateful to Bono. The actress revealed that the U2 singer helped her calm down when she became scared by a thunderstorm while travelling on a plane.

- she ⇒ Sophia Loren
- the actress ⇒ Sophia Loren
- the U2 singer ⇒ Bono
- her ⇒ Sophia Loren
- she ⇒ Sophia Loren
Interpreting anaphoric expressions

Interpreting (‘resolving’) an anaphoric expressions involves at least three tasks:

- Deciding whether the expression is in fact anaphoric
- Identifying its antecedent (possibly not introduced by a nominal)
- Determining its meaning (cfr. identity of sense vs. identity of reference)

(not necessarily in this order!)
Anaphoric expressions: nominals

- **PRONOUNS:**
  
  Definite pronouns: Ross bought {a radiometer | three kilograms of after-dinner mints} and gave {it | them} to Nadia for her birthday. (Hirst, 1981)

  Indefinite pronouns: Sally admired Sue’s jacket, so she got one for Christmas. (Garnham, 2001)

  Reflexives: John bought himself an hamburger

- **DEFINITE DESCRIPTIONS:**
  
  A man and a woman came into the room. The man sat down.

  Epiteths: A man ran into my car. The idiot wasn’t looking where he was going.

- **DEMONSTRATIVES:**
  
  Tom has been caught shoplifting. That boy will turn out badly.

- **PROPER NAMES:**
Factors that affect the interpretation of anaphoric expressions

- Factors:
  - Surface similarity
  - Morphological features (agreement)
  - Syntactic information
  - Salience
  - Lexical and commonsense knowledge
Coreference vs. Anaphora

Example
Sophia Loren says she will always be grateful to Bono. The actress revealed that the U2 singer helped her calm down when she became scared by a thunderstorm while travelling on a plane.

Coreference Chains:
- {Sophia Loren, she, the actress, her, she}
- {Bono, the U2 singer}
- {a thunderstorm}
- {a plane}
Coreference vs. Anaphora

COREFERENT, not ANAPHORIC
- two mentions of same object in different documents

ANAPHORIC, not COREFERENT
- identity of sense: John bought a shirt, and Bill got ONE, too
- Dependence on non-referring expressions: EVERY CAR had been stripped of ITS paint
- Non-identity relations (bridging, “other” etc)
- Reference to events
A brief history of early work on anaphora resolution

- Algorithms heavily based on (hand-coded) commonsense knowledge: Charniak 72, Winograd 74, Hobbs 1974, Wilks 1975
- Syntax-based algorithms: Hobbs 1974
- Salience-based algorithms: Brennan et al, Strube & Hahn, Tetreault
First approaches to anaphora resolution: Charniak, Winograd, Wilks

- Heavy emphasis on inference
- Hand-coded
- No formal evaluation
1970s Example: SHRDLU

http://hci.stanford.edu/~winograd/shrdlu/

Terry Winograd. 1971.
MIT Ph.D. Thesis.

• This is one of the only dialogs it knows.
• SHRDLU is too stupid to make mistakes.
• Beautiful “Demo-ware”

Person: PICK UP A BIG RED BLOCK.
Computer: OK. (does it)
Person: GRASP THE PYRAMID.
Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
Person: FIND A BLOCK WHICH IS TALLER THAN THE ONE YOU ARE HOLDING AND PUT IT INTO THE BOX.
Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.
Computer: OK. (does it)
Person: WHAT DOES THE BOX CONTAIN?
Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.
Person: WHAT IS THE PYRAMID SUPPORTED BY?
Computer: THE BOX.
Person: HOW MANY BLOCKS ARE NOT IN THE BOX?
Computer: FOUR OF THEM.
Person: IS AT LEAST ONE OF THEM NARROWER THAN THE ONE WHICH I TOLD YOU TO PICK UP?
Computer: YES, THE RED CUBE.
Anaphora in SHRDLU

- First example of HISTORY LIST algorithm
- Uses a combination of agreement features and semantic constraints
- Check all possibilities and assign a global score rather than simply find the first match
- Score incorporates syn component: entities in subj position higher score than entities in object position, in turn ranked more highly than entities in adjunct position
- Performance made more impressive by including solutions to a number of complex cases, such as reference to events (Why did you do it?) – often ad hoc
Hobbs’ `Naïve Algorithm’ (Hobbs, 1974)

- The reference algorithm for PRONOUN resolution (until Soon et al it was the standard baseline)
  - Interesting since Hobbs himself in the 1974 paper suggests that this algorithm is very limited (and proposes one based on semantics)
- The first anaphora resolution algorithm to have an (informal) evaluation
- Purely syntax based
Hobbs: example

- **Mr. Smith** saw a driver of *his* truck.

- Mr. Smith saw *a driver* in *his* truck.
Hobbs’ `Naïve Algorithm’ (Hobbs, 1974)

- Works off ‘surface parse tree’
- Starting from the position of the pronoun in the surface tree,
  - first go up the tree looking for an antecedent in the current sentence (left-to-right, breadth-first);
  - then go to the previous sentence, again traversing left-to-right, breadth-first.
  - And keep going back
Hobbs’ algorithm: Intrasentential anaphora

- Steps 2 and 3 deal with intrasentential anaphora and incorporate basic syntactic constraints:

- Also: John’s portrait of him
Hobbs’ Algorithm: intersentential anaphora

candidate

NP
Bill

V
is

NP
a good friend

S

NP
John

V
likes

NP
him

X

NP
him

V
likes

NP
Bill

S

V
is

NP
a good friend

NP
John

V
likes

NP
him

X
Evaluation

The first anaphora resolution algorithm to be evaluated in a systematic manner, and still often used as baseline (hard to beat!)

Hobbs, 1974:
- 300 pronouns from texts in three different styles (a fiction book, a non-fiction book, a magazine)
- Results: 88.3% correct without selectional constraints, 91.7% with SR
- 132 ambiguous pronouns; 98 correctly resolved.

Tetreault 2001 (no selectional restrictions; all pronouns)
- 1298 out of 1500 pronouns from 195 NYT articles (76.8% correct)
- 74.2% correct intra, 82% inter

Main limitations
- Reference to propositions excluded
- Plurals
- Reference to events
Salience-based algorithms

- Common hypotheses:
  - Entities in discourse model are RANKED by salience
  - Salience gets continuously updated
  - Most highly ranked entities are preferred antecedents

- Variants:
  - DISCRETE theories (Sidner, Brennan et al, Strube & Hahn): 1-2 entities singled out
  - CONTINUOUS theories (Alshawi, Lappin & Leass, Strube 1998, LRC): only ranking
Factors that affect prominence

- Distance
- Order of mention in the sentence
  - Entities mentioned earlier in the sentence more prominent
- Type of NP (proper names > other types of NPs)
- Number of mentions
- Syntactic position (subj > other GF, matrix > embedded)
- Semantic role (‘implicit causality’ theories)
- Discourse structure
Salience-based algorithms

- **Sidner 1979:**
  - Most extensive theory of the influence of salience on several types of anaphors
  - Two FOCI: discourse focus, agent focus
  - never properly evaluated

- **Brennan et al 1987 (see Walker 1989):**
  - Ranking based on grammatical function
  - One focus (CB)

- **Strube & Hahn 1999**
  - Ranking based on information status (NP type)

- **S-List (Strube 1998): drop CB**
  - LRC (Tetreault): incremental
Topics & pronominalization: linguistic evidence

Grosz et al (1995): texts in which other entities are pronominalized (rather than the ‘central entity’) less felicitous

(1) a. Something must be wrong with John.
    b. He has been acting quite odd.
    c. He called up Mike yesterday.
    d. John wanted to meet him quite urgently.

(2) a. Something must be wrong with John.
    b. He has been acting quite odd.
    c. He called up Mike yesterday.
    d. He wanted to meet him quite urgently.
## Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>PTB-News (1694)</th>
<th>PTB-Fic (511)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRC</td>
<td>74.9%</td>
<td>72.1%</td>
</tr>
<tr>
<td>S-List</td>
<td>71.7%</td>
<td>66.1%</td>
</tr>
<tr>
<td>BFP</td>
<td>59.4%</td>
<td>46.4%</td>
</tr>
</tbody>
</table>
Comparison with ML techniques of the time

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>All 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRC</td>
<td>76.7%</td>
</tr>
<tr>
<td>Ge et al. (1998)</td>
<td>87.5% (*)</td>
</tr>
<tr>
<td>Morton (2000)</td>
<td>79.1%</td>
</tr>
</tbody>
</table>
ARPA’s Message Understanding Conference (1992-1997)
First big initiative in Information Extraction
Changed NLP by producing the first sizeable annotated data for semantic tasks including
  - named entity extraction
  - ‘coreference’
Developed first methods for evaluating anaphora resolution systems
MUC terminology:

- MENTION: any markable
- COREFERENCE CHAIN: a set of mentions referring to an entity
- KEY: the (annotated) solution (a partition of the mentions into coreference chains)
- RESPONSE: the coreference chains produced by a system
Since MUC

- **ACE**
  - Much more data
  - Subset of mentions
  - IE perspective

- **SemEval-2010**
  - More languages
  - CL perspective

- **Evalita**
  - Italian (ACE-style)

- **CoNLL-OntoNotes**
MODERN WORK IN ANAPHORA RESOLUTION

- Availability of the first anaphorically annotated corpora from MUC6 onwards made it possible
  - To evaluate anaphora resolution on a large scale
  - To train statistical models
PROBLEMS TO BE ADDRESSED BY LARGE-SCALE ANAPHORIC RESOLVERS

- Robust mention identification
  - Requires high-quality parsing
- Robust extraction of morphological information
- Classification of the mention as referring / predicative / expletive
- Large scale use of lexical knowledge and inference
Problems to be resolved by a large-scale AR system: mention identification

- Typical problems:
  - Nested NPs (possessives)
    - [a city] 's [computer system] → [[a city’’s computer system]]
  - Appositions:
    - [Madras], [India] → [Madras, [India]]
  - Attachments
Computing agreement: some problems

- Gender:
  - [India] withdrew HER ambassador from the Commonwealth
  - “…to get a customer’s 1100 parcel-a-week load to its doorstep”
    - [actual error from LRC algorithm]

- Number:
  - The Union said that THEY would withdraw from negotiations until further notice.
Problems to be solved: anaphoricity determination

- Expletives:
  - IT’s not easy to find a solution
  - Is THERE any reason to be optimistic at all?
- Non-anaphoric definites
PROBLEMS: LEXICAL KNOWLEDGE, INFERENCE

- Still the weakest point
- The first breakthrough: WordNet
- Then methods for extracting lexical knowledge from corpora
- A more recent breakthrough: Wikipedia
MACHINE LEARNING APPROACHES TO ANAPHORA RESOLUTION

- **First efforts:** MUC-2 / MUC-3 (Aone and Bennet 1995, McCarthy & Lehnert 1995)
- **Most of these:** SUPERVISED approaches
  - Early (NP type specific): Aone and Bennet, Vieira & Poesio
  - McCarthy & Lehnert: all NPs
  - Soon et al: standard model
- **UNSUPERVISED approaches**
  - Eg Cardie & Wagstaff 1999, Ng 2008
ANAPHORA RESOLUTION AS A CLASSIFICATION PROBLEM

1. Classify NP1 and NP2 as coreferential or not
2. Build a complete coreferential chain
SUPERVISED LEARNING FOR ANAPHORA RESOLUTION

- Learn a model of coreference from training labeled data
- need to specify
  - learning algorithm
  - feature set
  - clustering algorithm
SOME KEY DECISIONS

- **ENCODING**
  - I.e., what positive and negative instances to generate from the annotated corpus
  - Eg treat all elements of the coref chain as positive instances, everything else as negative:

- **DECODING**
  - How to use the classifier to choose an antecedent
  - Some options: ‘sequential’ (stop at the first positive), ‘parallel’ (compare several options)
Early machine-learning approaches

- Main distinguishing feature: concentrate on a single NP type
- Both hand-coded and ML:
  - Aone & Bennett (pronouns)
  - Vieira & Poesio (definite descriptions)
- Ge and Charniak (pronouns)
Mention-pair model

- Soon et al. (2001)
- First ‘modern’ ML approach to anaphora resolution
- Resolves ALL anaphors
- Fully automatic mention identification
- Developed instance generation & decoding methods used in a lot of work since
Soon et al. (2001)

Wee Meng Soon, Hwee Tou Ng, Daniel Chung Yong Lim, A Machine Learning Approach to Coreference Resolution of Noun Phrases, Computational Linguistics 27(4):521–544
MENTION PAIRS

<ANAPHOR (j), ANTECEDENT (i)>
Sophia Loren says she will always be grateful to Bono. The actress revealed that the U2 singer helped her calm down when she became scared by a thunderstorm while travelling on a plane.
Sophia Loren says she will always be grateful to Bono. The actress revealed that the U2 singer helped her calm down when she became scared by a thunderstorm while travelling on a plane.
Mention-pair: encoding

- Sophia Loren
- she
- Bono
- The actress
- the U2 singer
- U2
- her
- she
- she
- a thunderstorm
- a plane
Mention-pair: encoding

- Sophia Loren → none
- she → (she,S.L,+)
- Bono → none
- The actress → (the actress, Bono,-),(the actress,she,+)
- the U2 singer → (the U2 s., the actress,-), (the U2 s.,Bono,+)
- U2 → none
- her → (her,U2,-),(her,the U2 singer,-),(her,the actress,+)
- she → (she, her,+)
- a thunderstorm → none
- a plane → none
Mention-pair: decoding

- Right to left, consider each antecedent until classifier returns true
Preprocessing: Extraction of Markables

- Tokenization & Sentence Segmentation
- Morphological Processing
- POS tagger
- Standard HMM based tagger
- NP Identification
- Named Entity Recognition
- Nested Noun Phrase Extraction
- Semantic Class Determination

Markables

- HMM based, recognizes organization, person, location, date, time, money, percent
- 2 kinds: prenominals such as ((wage) reduction) and possessive NPs such as (his) dog.
- More on this in a bit!
Soon et al: preprocessing

- POS tagger: HMM-based
  - 96% accuracy
- Noun phrase identification module
  - HMM-based
  - Can identify correctly around 85% of mentions
- NER: reimplementation of Bikel Schwartz and Weischedel 1999
  - HMM based
  - 88.9% accuracy
Soon et al 2001: Features of mention - pairs

- NP type
- Distance
- Agreement
- Semantic class
**Soon et al: NP type and distance**

<table>
<thead>
<tr>
<th>NP type of anaphor j (3)</th>
<th>j-pronoun, def-np, dem-np (bool)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP type of antecedent i</td>
<td>i-pronoun (bool)</td>
</tr>
<tr>
<td>Types of both</td>
<td>both-proper-name (bool)</td>
</tr>
<tr>
<td>DIST</td>
<td>0, 1, ...</td>
</tr>
</tbody>
</table>
Soon et al features: string match, agreement, syntactic position

**STR_MATCH**

**ALIAS**
- dates (1/8 – January 8)
- person (Bent Simpson / Mr. Simpson)
- organizations: acronym match (Hewlett Packard / HP)

**AGREEMENT FEATURES**
- number agreement
- gender agreement

**SYNTACTIC PROPERTIES OF ANAPHOR**
- occurs in appositive contraction
Soon et al: semantic class agreement

SEMCLASS = true iff semclass(i) <= semclass(j) or vice versa
Soon et al: evaluation

- MUC-6:
  - $P=67.3$, $R=58.6$, $F=62.6$

- MUC-7:
  - $P=65.5$, $R=56.1$, $F=60.4$

- Results about 3rd or 4th amongst the best MUC-6 and MUC-7 systems
Basic errors: synonyms & hyponyms

Toni Johnson pulls a tape measure across the front of what was once [a stately Victorian home].

.....

The remainder of [THE HOUSE] leans precariously against a sturdy oak tree.

Most of the 10 analysts polled last week by Dow Jones International News Service in Frankfurt . . . expect [the US dollar] to ease only mildly in November

.....

Half of those polled see [THE CURRENCY] …
[Bach]’s air followed. Mr. Stolzman tied [the composer] in by proclaiming him the great improviser of the 18th century.

[The FCC] .... [the agency]
FALSE NEGATIVE:
A new incentive plan for advertisers ...
.... The new ad plan ....

FALSE NEGATIVE:
The 80-year-old house
....
The Victorian house ...
Soon et al. (2001): Error Analysis (on 5 random documents from MUC-6)

### Types of Errors Causing Spurious Links (→ affect precision)

<table>
<thead>
<tr>
<th>Error Description</th>
<th>Frequency</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prenominal modifier string match</td>
<td>16</td>
<td>42.1%</td>
</tr>
<tr>
<td>Strings match but noun phrases refer to different entities</td>
<td>11</td>
<td>28.9%</td>
</tr>
<tr>
<td>Errors in noun phrase identification</td>
<td>4</td>
<td>10.5%</td>
</tr>
<tr>
<td>Errors in apposition determination</td>
<td>5</td>
<td>13.2%</td>
</tr>
<tr>
<td>Errors in alias determination</td>
<td>2</td>
<td>5.3%</td>
</tr>
</tbody>
</table>

### Types of Errors Causing Missing Links (→ affect recall)

<table>
<thead>
<tr>
<th>Error Description</th>
<th>Frequency</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inadequacy of current surface features</td>
<td>38</td>
<td>63.3%</td>
</tr>
<tr>
<td>Errors in noun phrase identification</td>
<td>7</td>
<td>11.7%</td>
</tr>
<tr>
<td>Errors in semantic class determination</td>
<td>7</td>
<td>11.7%</td>
</tr>
<tr>
<td>Errors in part-of-speech assignment</td>
<td>5</td>
<td>8.3%</td>
</tr>
<tr>
<td>Errors in apposition determination</td>
<td>2</td>
<td>3.3%</td>
</tr>
<tr>
<td>Errors in tokenization</td>
<td>1</td>
<td>1.7%</td>
</tr>
</tbody>
</table>
Mention-pair: locality

- Bill Clinton .. Clinton .. Hillary Clinton
- Bono .. He .. They
Subsequent developments

- Improved versions of the mention-pair model: Ng and Cardie 2002, Hoste 2003
- Improved mention detection techniques (better parsing, joint inference)
- Anaphoricity detection
- Using lexical / commonsense knowledge (particularly semantic role labelling)
- Different models of the task: ENTITY MENTION model, graph-based models
- Salience
- Development of AR toolkits (GATE, LingPipe, GUITAR, BART)
Modern ML approaches

- ILP: start from pairs, impose global constraints
- Entity-mention models: global encoding/decoding
- Feature engineering
Integer Linear Programming

- Optimization framework for global inference
- NP-hard
- But often fast in practice
- Commercial and publicly available solvers
ILP: general formulation

- Maximize objective function
  \[ \sum \lambda_i X_i \]
- Subject to constraints
  \[ \sum \alpha_i X_i \geq \beta_i \]
- \( X_i \) – integers
ILP for coreference

- Klenner (2007)
- Denis & Baldridge
- Finkel & Manning (2008)
ILP for coreference

- Step 1: Use Soon et al. (2001) for encoding. Learn a classifier.
- Step 2: Define objective function:
  \[ \sum \lambda_{ij} * X_{ij} \]
  - \( X_{ij} = -1 \) – not coreferent
  - \( 1 \) – coreferent
  \( \lambda_{ij} \) – the classifier's confidence value
ILP for coreference: example

- Bill Clinton .. Clinton .. Hillary Clinton
- (Clinton, Bill Clinton) $\rightarrow$ +1
- (Hillary Clinton, Clinton) $\rightarrow$ +0.75
- (Hillary Clinton, Bill Clinton) $\rightarrow$ -0.5 / -2

- $\max(1 \cdot X_{21} + 0.75 \cdot X_{32} - 0.5 \cdot X_{31})$
- Solution: $X_{21} = 1$, $X_{32} = 1$, $X_{31} = -1$
- This solution gives the same chain.
ILP for coreference

- Step 3: define constraints
- transitivity constraints:
  - $i < j < k$
  - $X_{ik} \geq X_{ij} + X_{jk} - 1$
Back to our example

- Bill Clinton .. Clinton .. Hillary Clinton
- (Clinton, Bill Clinton) → +1
- (Hillary Clinton, Clinton) → +0.75
- (Hillary Clinton, Bill Clinton) → -0.5 /-2

- \[ \text{max}(1 \times X_{21} + 0.75 \times X_{32} - 0.5 \times X_{31}) \]
- \[ X_{31} \geq X_{21} + X_{32} - 1 \]
Solutions

- \[ \text{max}(1 \cdot X_{21} + 0.75 \cdot X_{32} + \lambda_{31} \cdot X_{31}) \]
- \[ X_{31} \geq X_{21} + X_{32} - 1 \]
- \[ X_{21}, X_{32}, X_{31}, \lambda_{31} = -0.5 \quad \lambda_{31} = -2 \]
- \[ 1,1,1 \quad \text{obj}=1.25 \quad \text{obj}=-0.25 \]
- \[ 1,-1,-1 \quad \text{obj}=0.75 \quad \text{obj}=2.25 \]
- \[ -1,1,-1 \quad \text{obj}=0.25 \quad \text{obj}=1.75 \]
- \[ \lambda_{31} = -0.5: \text{same solution} \]
- \[ \lambda_{31} = -2: \{\text{Bill Clinton, Clinton}\}, \{\text{Hillary Clinton}\} \]
ILP constraints

- Transitivity
- Best-link
- Agreement etc as hard constraints
- Discourse-new detection
- Joint preprocessing
Entity-mention model

- Bell trees (Luo et al, 2004)
- Ng
- And many others..
Entity-mention model

- Mention-pair model: resolve mentions to mentions, fix the conflicts afterwards

- Entity-mention model: grow entities by resolving each mention to already created entities
Sophia Loren says she will always be grateful to Bono. The actress revealed that the U2 singer helped her calm down when she became scared by a thunderstorm while travelling on a plane.
Example

- Sophia Loren
- she
- Bono
- The actress
- the U2 singer
- U2
- her
- she
- a thunderstorm
- a plane
Mention-pair vs. Entity-mention

- Resolve “her” with a perfect system
- Mention-pair – build a list of candidate mentions:
  - Sophia Loren, she, Bono, The actress, the U2 singer, U2
- process backwards.. {her, the U2 singer}
- Entity-mention – build a list of candidate entities:
  - {Sophia Loren, she, The actress}, {Bono, the U2 singer}, {U2}
First-order features

- Using pairwise boolean features and quantifiers
  - Ng
  - Recasens
  - Unsupervised

- Semantic Trees
History features in mention-pair modelling

- Yang et al (pronominal anaphora)
- Salience
Entity update

- Incremental
- Beam (Luo)
- Markov logic – joint inference across mentions (Poon & Domingos)
Ranking

- Coreference resolution with a classifier:
  - Test candidates
  - Pick the best one

- Coreference resolution with a ranker
  - Pick the best one directly
Features

- Ng & Cardie (2003): 50+ features
- Uryupina (2007): 300+ features
- Bengston & Roth (2008): feature analysis
- BART: around 50 features
New features

- More semantic knowledge, extracted from text (Garera & Yarowsky), Wordnet (Harabagiu) or Wikipedia (Ponzetto & Strube)
- Better NE processing (Bergsma)
- Syntactic constraints (back to the basics)
- Approximate matching (Strube)
Evaluation of coreference resolution systems

- Lots of different measures proposed
- **ACCURACY:**
  - Consider a mention correctly resolved if
    - Correctly classified as anaphoric or not anaphoric
    - ‘Right’ antecedent picked up
- Measures developed for the competitions:
  - Automatic way of doing the evaluation
- More realistic measures (Byron, Mitkov)
  - Accuracy on ‘hard’ cases (e.g., ambiguous pronouns)
Vilain et al. (1995)

- The official MUC scorer
- Based on precision and recall of links
- Views coreference scoring from a model-theoretical perspective
  - Sequences of coreference links (= coreference chains) make up entities as SETS of mentions
  - Takes into account the transitivity of the IDENT relation
Identify the minimum number of link modifications required to make the set of mentions identified by the system as coreferring perfectly align to the gold-standard set.

- Units counted are link edits.
Given that A, B, C and D are part of a coreference chain in the KEY, treat as equivalent the two responses:

And as superior to:
MUC-6 Coreference Scoring Metric: Computing Recall

- To measure RECALL, look at how each coreference chain $S_i$ in the KEY is partitioned in the RESPONSE, and count how many links would be required to recreate the original.
- Average across all coreference chains.
MUC-6 Coreference Scoring Metric: Computing Recall

- $S$ => set of key mentions
- $p(S)$ => Partition of $S$ formed by intersecting all system response sets $R_i$
  - Correct links: $c(S) = |S| - 1$
  - Missing links: $m(S) = |p(S)| - 1$

- **Recall**: $\frac{c(S) - m(S)}{c(S)} = \frac{|S| - |p(S)|}{|S| - 1}$
- **Recall$_T$**: $\frac{\sum |S| - |p(S)|}{\sum |S| - 1}$
MUC-6 Coreference Scoring Metric: Computing Recall

- Considering our initial example

- KEY: 1 coreference chain of size 4 (|S| = 4)
- (INCORRECT) RESPONSE: partitions the coref chain in two sets (|p(S)| = 2)
- R = 4-2 / 4-1 = 2/3
MUC-6 Coreference Scoring
Metric: Computing Precision

To measure PRECISION, look at how each coreference chain $S_i$ in the RESPONSE is partitioned in the KEY, and count how many links would be required to recreate the original

- Count links that would have to be (incorrectly) added to the key to produce the response
- I.e., ‘switch around’ key and response in the previous equation
MUC-6 Scoring in Action

- **KEY** = [A, B, C, D]
- **RESPONSE** = [A, B], [C, D]

Recall \( \frac{4 - 2}{3} = 0.66 \)

Precision \( \frac{(2 - 1) + (2 - 1)}{(2 - 1) + (2 - 1)} = 1.0 \)

F-measure \( \frac{2 * 2/3 * 1}{2/3 + 1} = 0.79 \)
Beyond MUC Scoring

- Problems:
  - Only gain points for links. No points gained for correctly recognizing that a particular mention is not anaphoric.
  - All errors are equal.
Not all links are equal
Beyond MUC Scoring

- Alternative proposals:
  - Bagga & Baldwin’s B-CUBED algorithm (1998)
  - Luo’s recent proposal, CEAF (2005)
B-CUBED (BAGGA AND BALDWIN, 1998)

- MENTION-BASED
  - Defined for singleton clusters
  - Gives credit for identifying non-anaphoric expressions

- Incorporates weighting factor
  - Trade-off between recall and precision normally set to equal
B-CUBED: PRECISION / RECALL

\[
\text{Precision}_i = \frac{\text{number of correct elements in the output chain containing entity}_i}{\text{number of elements in the output chain containing entity}_i}
\]

\[
\text{Recall}_i = \frac{\text{number of correct elements in the output chain containing entity}_i}{\text{number of elements in the truth chain containing entity}_i}
\]

\[
\text{Final Precision} = \sum_{i=1}^{N} w_i \times \text{Precision}_i
\]

\[
\text{Final Recall} = \sum_{i=1}^{N} w_i \times \text{Recall}_i
\]
Comparison of MUC and B-Cubed

- Both rely on intersection operations between reference and system mention sets
- B-Cubed takes a MENTION-level view
  - Scores singleton, i.e. non-anaphoric mentions
  - Tends towards higher scores
    - Entity clusters being used “more than once” within scoring metric is implicated as the likely cause
  - Greater discriminability than the MUC metric
Comparison of MUC and B-Cubed

- MUC prefers large coreference sets
- B-Cubed overcomes the problem with the uniform cost of alignment operations in MUC scoring
Entity-based score metrics

- **ACE metric**
  - Computes a score based on a mapping between the entities in the key and the ones output by the system
  - Different (mis-)alignments costs for different mention types (pronouns, common nouns, proper names)

- **CEAF (Luo, 1995)**
  - Computes also an alignment score score between the key and response entities but uses no mention-type cost matrix
Precision and recall measured on the basis of the SIMILARITY $\Phi$ between ENTITIES (= coreference chains)
- Difference similarity measures can be imagined

Look for OPTIMAL MATCH $g^*$ between entities
- Using Kuhn-Munkres graph matching algorithm
ENTITY-BASED PRECISION AND RECALL IN CEAF

\[ p = \frac{\Phi(g^*)}{\sum_i \phi(S_i, S_i)} \]

\[ r = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)} \]

\[ F = \frac{2pr}{p + r} \]
Recast the scoring problem as **bipartite matching**

Find the best match using the Kuhn-Munkres Algorithm

Matching score = 6

Recall = $\frac{6}{9} = 0.66$

Prec = $\frac{6}{12} = 0.5$

F-measure = 0.57
MUC vs B-CUBE vs. CEAF (from Luo 2005)

Figure 1: Example entities: (1) truth; (2) system response (a); (3) system response (b); (4) system response (c); (5) system response (d)

<table>
<thead>
<tr>
<th>System response</th>
<th>MUC</th>
<th>B-cube</th>
<th>CEAF</th>
<th>$\phi_3(\cdot,\cdot)$</th>
<th>$\phi_4(\cdot,\cdot)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>0.947</td>
<td>0.865</td>
<td>0.833</td>
<td>0.733</td>
<td></td>
</tr>
<tr>
<td>(b)</td>
<td>0.947</td>
<td>0.737</td>
<td>0.583</td>
<td>0.667</td>
<td></td>
</tr>
<tr>
<td>(c)</td>
<td>0.900</td>
<td>0.545</td>
<td>0.417</td>
<td>0.294</td>
<td></td>
</tr>
<tr>
<td>(d)</td>
<td>–</td>
<td>0.400</td>
<td>0.250</td>
<td>0.178</td>
<td></td>
</tr>
</tbody>
</table>
Set vs. entity-based score metrics

- **MUC** underestimates **precision errors**
  - More credit to larger coreference sets
- **B-Cubed** underestimates **recall errors**
  - More credit to smaller coreference sets
- **ACE** reasons at the entity-level
  - Results often more difficult to interpret
Practical experience with these metrics

- BART computes these three metrics
- Hard to tell which metric is better at identifying better performance
Byron 2001:
- Many researchers remove from the reported evaluation cases which are ‘out of the scope of the algorithm’
- E.g. for pronouns: expletives, discourse deixis, cataphora
- Need to make sure that systems being compared are considering the same cases

Mitkov:
- Distinguish between hard (= highly ambiguous) and easy cases
GOLD MENTIONS vs. SYSTEM MENTIONS

- Apparent split in performance on same datasets:
  - ACE 2004:
    - Luo & Zitouni 2005: ACE score of 80.8
    - Yang et al 2008: ACE score of 67

- Reason:
  - Luo & Zitouni report results on GOLD MENTIONs
  - Yang et al results on SYSTEM mentions
Anaphora:
  Difficult task
  Needed for NLP applications
  Requires substantial preprocessing

First algorithms:
  Charniak, Winograd, Wilks
  Pronouns: Hobbs
  Salience: S-List, LRC
  MUC, ACE, SemEval

Mention-pair model:
  Based on (anaphor, antecedent) pairs
  Widely accepted as a baseline
  Very local
Modern Coreference Resolution:
    ILP
    Entity-mention models
    Features
Evaluation metrics
    MUC
    BCUBED, ACE
    CEAF
Thank you!

Next time: lab on coreference resolution with BART

Please download BART from http://bart-coref.org/
Readings

- Kehler’s chapter on Discourse in Jurafsky & Martin
  - Alternatively: Elango’s survey
  - Also in Readings in Natural Language Processing,
- Ng and Cardie 2002, Improving machine learning approaches to coreference resolution, Proc. ACL