Machine Learning for Coreference Resolution: From Local Classification to Global Ranking

Vincent Ng

Human Language Technology Research Institute

University of Texas at Dallas

Goal

Improve the robustness of the standard machine learning approach to noun phrase coreference resolution

Plan for the Talk

- Noun phrase coreference resolution
  - standard machine learning approach
  - why the standard approach is not robust
- A ranking approach to coreference
- Evaluation

Noun Phrase Coreference

- Identify the noun phrases (NPs) that refer to the same real-world entity

- Inherently a clustering problem
  - partition the NPs into coreference classes

- The coreference relation $\mathrm{R}\colon \mathrm{NP} \times \mathrm{NP} \rightarrow \{\, \textit{coref}, \textit{not coref} \,\}$ is transitive
  - A & B coreferent, B & C coreferent → A & C coreferent

Standard Machine Learning Approach

- **Step 1: Classification**
  - given a description of two noun phrases, NP_i and NP_j, classifies the pair as *coreferent* or *not coreferent*
  - does not guarantee transitivity

- **Step 2: Clustering**
  - coordinates pairwise classification decisions

Aone and Bennett (1995), McCarthy and Lehnert (1995), Soon et al. (2001), Ng and Cardie (2002), Strube et al. (2002), Kehler et al. (2004), Yang et al. (2004)

Why It’s Not Robust

- Design decisions (e.g., the learning algorithm and clustering algorithm to be employed) are too ad-hoc
  - *closest-first* clustering algorithm is a common choice
    - selects the closest preceding coreferent NP to be the antecedent
    - performs no search in the space of possible partitions

    Is it better than the other clustering procedures? Too greedy?

- Does not optimize for clustering-level accuracy
  - coreference classifier is trained and optimized independently of the clustering algorithm to be used

A Ranking Approach to Coreference

Given a set of NPs to be clustered,

1. generate *n* candidate NP partitions ?
   - use *n* pre-selected learning-based coreference systems

2. rank the candidate partitions ?
   - use a learned ranking model

3. select the top-ranked partition to be the final partition

Why Might the Ranking Approach be Better?

Standard Approach

- Design decisions too ad-hoc

Ranking Approach

- Avoid ad-hoc design decisions
  - should learner *A* or learner *B* be used?
  - construct two coreference resolvers with one employing *A* and the other *B*
  - add both to our pre-selected set of *n* coreference systems

Why Might the Ranking Approach be Better?

Standard Approach

- Design decisions too ad-hoc
- Does not optimize for clustering-level accuracy

Ranking Approach

- Avoid ad-hoc design decisions
- Optimize the ranking model w.r.t. the coreference scoring program
  - Idea: train the model such that it behaves like the scoring program

A Ranking Approach to Coreference

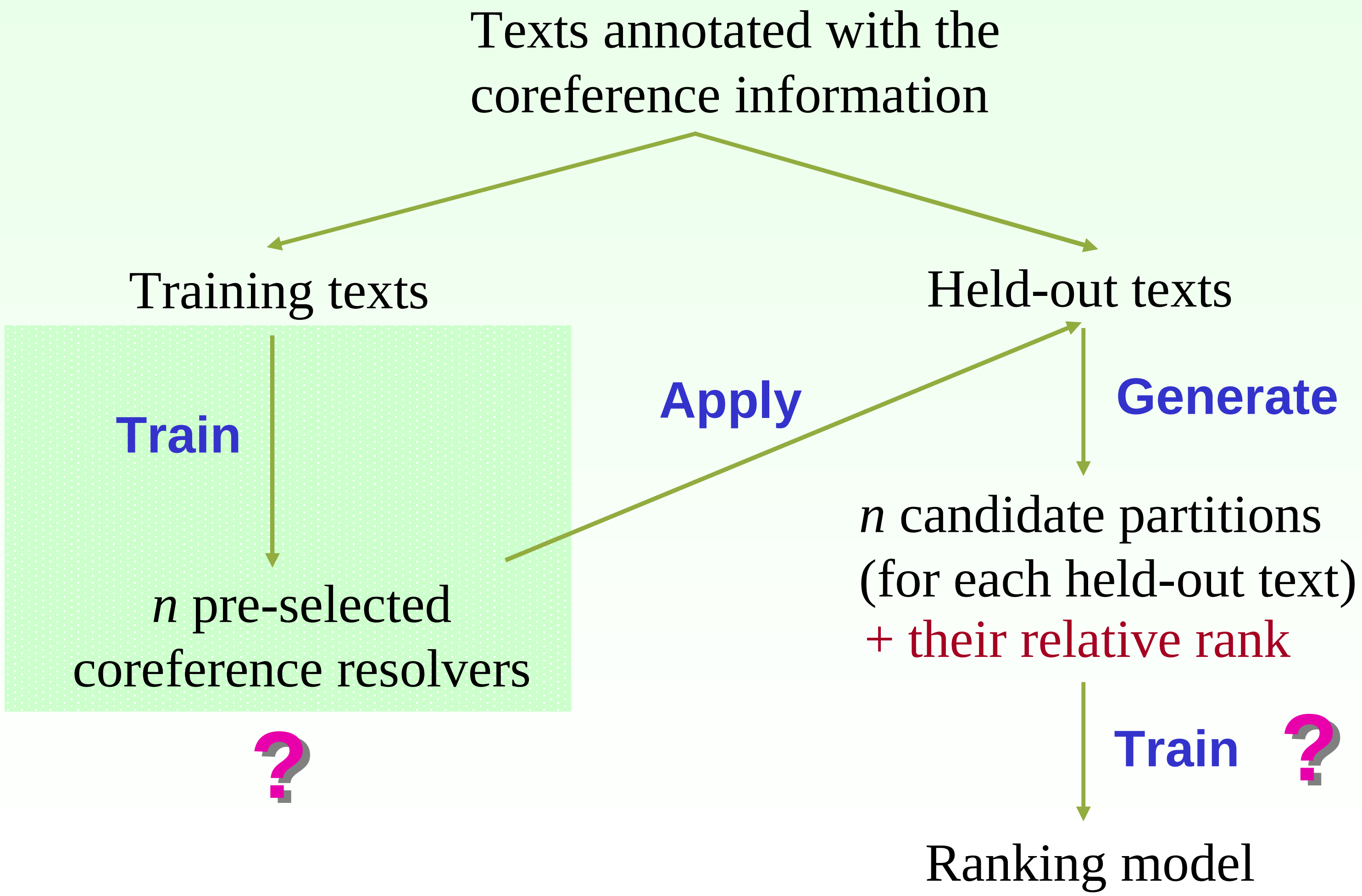
Given a set of NPs to be clustered,

1. generate *n* candidate NP partitions
   - use *n* pre-selected learning-based coreference systems

2. rank the candidate partitions
   - use a learned ranking model

3. select the top-ranked partition to be the final partition

Overview of the Training Procedure



Texts annotated with the coreference information

Training texts

Train

n pre-selected coreference resolvers

?

Apply

Held-out texts

Generate

n candidate partitions (for each held-out text)

+ their relative rank

Train ?

Ranking model

Selecting *n* Coreference Systems

- A learning-based coreference system can be defined by
  1. the learning algorithm used to train the classifier
  2. the method of creating training instances for the learner
  3. the feature set used to represent an instance
  4. the clustering algorithm used to induce a partition

- To select *n* learning-based coreference systems
  - select n_1 learning algorithms ?
  - select n_2 method of creating training instances ?
  - select n_3 feature sets ?
  - select n_4 clustering algorithms ?

→ yields $n = n_1 * n_2 * n_3 * n_4$ distinct coreference systems

Learning Algorithms

- C4.5 decision tree learner [Quinlan, 1993]
- RIPPER rule learner [Cohen, 1995]
- Maximum entropy classification [Berger et al., 1996]

- Learned classification models
  - Input: test instance (represents a pair of NPs)
  - Output: the likelihood that the two NPs are coreferent (*coreferent* if likelihood >= 0.5; *not coreferent* otherwise)

Training Instance Creation Methods

- McCarthy and Lehnert’s (1995) method
  - one instance for each (ordered) pair of noun phrases [n NPs Ł ${}_{n}C_{2}$ training instances]
    - class value: + or -
  - class distributions too skewed?

Training Instance Creation Methods

- McCarthy and Lehnert's (1995) method
- Soon et al.'s (2001) method
  - less skewed class distributions
  - but … a positive instance may have a non-pronominal NP paired with a pronominal antecedent (e.g., [*he*, *Mr. Smith*])

Training Instance Creation Methods

- McCarthy and Lehnert’s (1995) method
- Soon et al.’s (2001) method
- Ng and Cardie’s (2002) method
  - same as Soon et al., except that a non-pronominal NP is paired with the closest non-pronominal antecedent

Feature Sets

- The Soon et al. (2001) feature set
  - 12 features: lexical, grammatical, semantic, and positional
- The Ng and Cardie (2002) feature set
  - expands Soon et al.'s feature set to a deeper set of 53

Clustering Algorithms

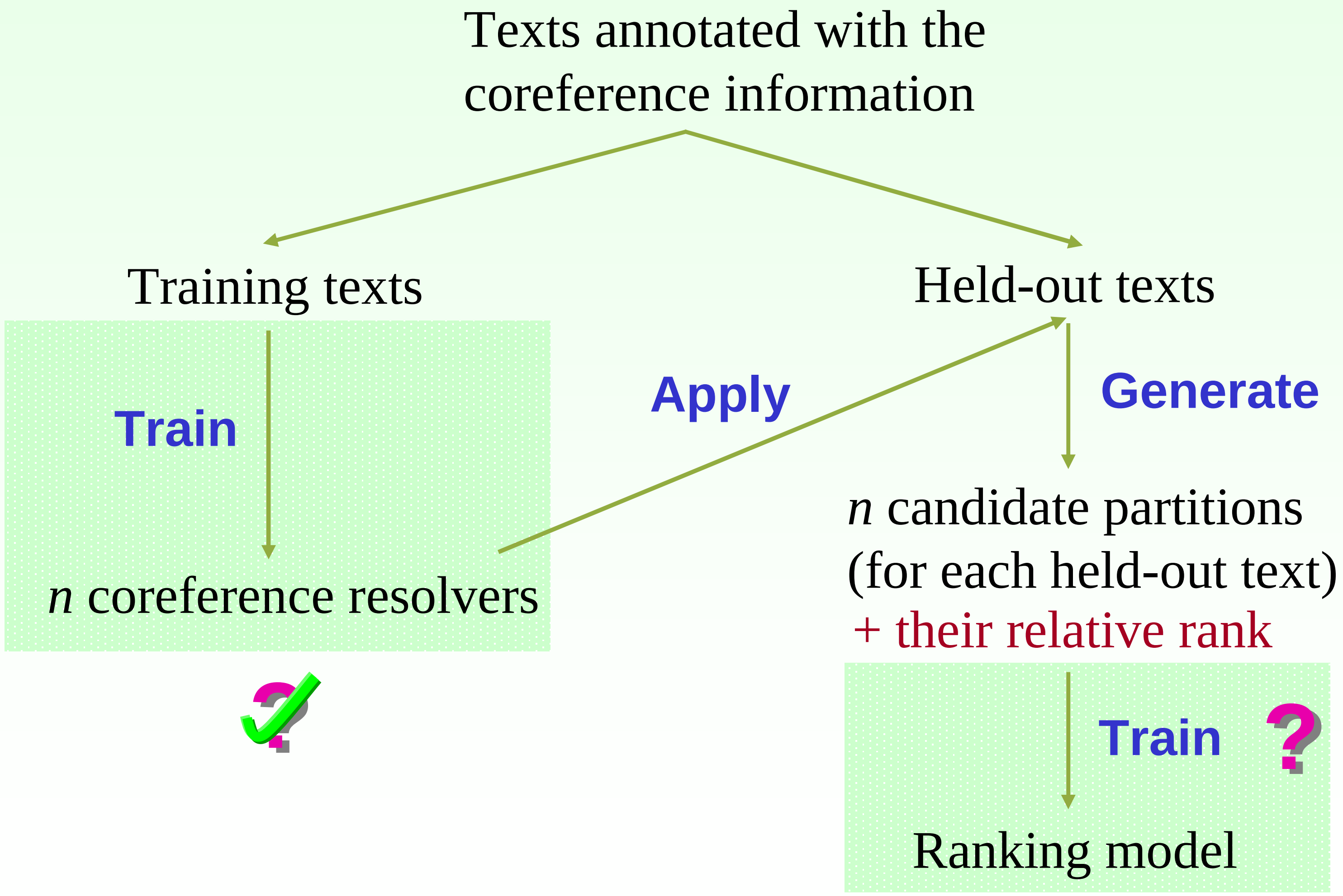
- Closest-first clustering [Soon et al, 2001; Strube et al, 2002]
  - puts an NP and its closest preceding coreferent NP into the same cluster
- Best-first clustering [Aone and Bennett, 1995]
  - puts an NP and its *most likely* preceding coreferent NP into the same cluster
  - potentially improves precision
- Aggressive-merge clustering [McCarthy and Lehnert, 1995]
  - puts an NP and all of its preceding coreferent NPs into the same cluster
  - potentially improves recall

Training *n* Coreference Systems

- Learning algorithms (3)
  - C4.5, RIPPER, maximum entropy
- Training instance creation methods (3)
  - McCarthy and Lehnert, Soon et al., Ng and Cardie
- Feature sets (2)
  - Soon et al, Ng and Cardie
- Clustering algorithms (3)
  - closest-first, best-first, aggressive-merge

54 coreference systems

Overview of the Training Procedure



Texts annotated with the coreference information

Training texts

Held-out texts

Train

n coreference resolvers

✓

Apply

Generate

n candidate partitions (for each held-out text)

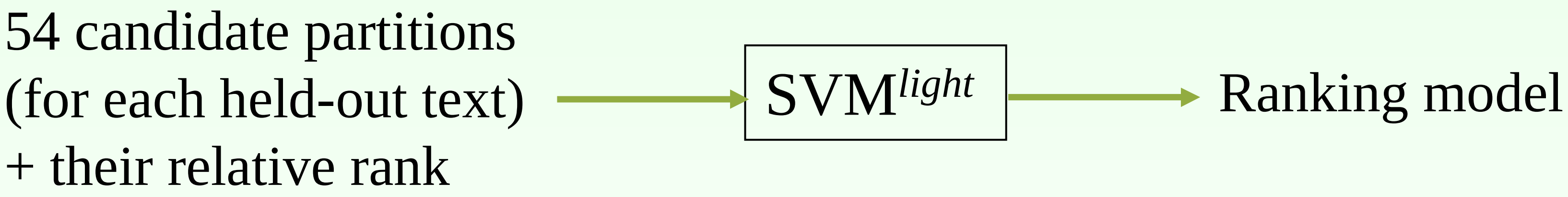
+ their relative rank

Train ?

Ranking model

Learning to Rank Candidate Partitions

- Use SVM^{light} [Joachims, 2002]



54 candidate partitions
(for each held-out text)
+ their relative rank

SVM^{light}

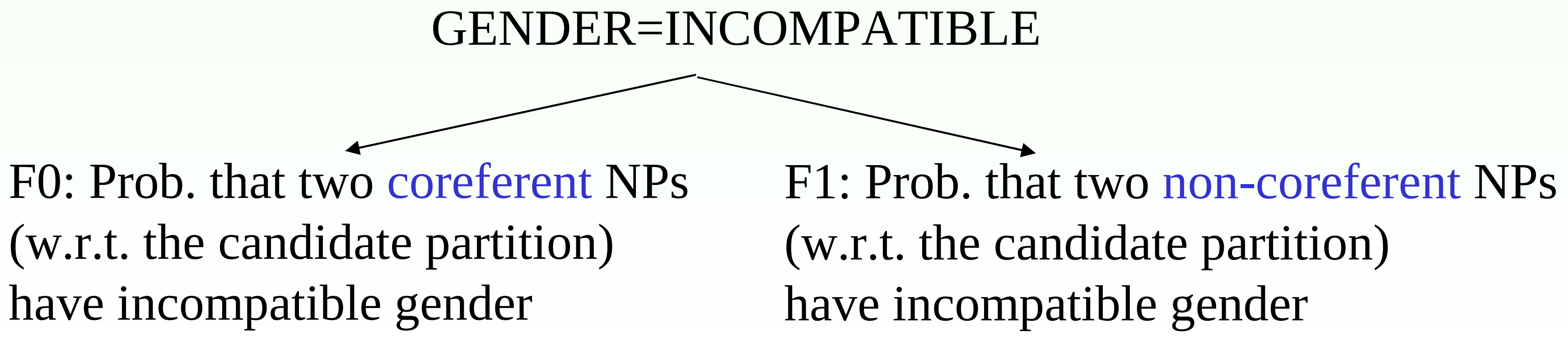
Ranking model

- To create training data, we need to
  - represent each candidate partition as a feature vector
  - compute the ranks of the candidate partitions

Instance Representation of a Candidate Partition

1. Partition-based features
   - characterize a partition
   - computed based on the features in the Ng and Cardie feature set

Derive two partition-based features from each attribute-value pair in the Ng and Cardie feature set



GENDER=INCOMPATIBLE

F0: Prob. that two coreferent NPs (w.r.t. the candidate partition) have incompatible gender

F1: Prob. that two non-coreferent NPs (w.r.t. the candidate partition) have incompatible gender

Good partitions should have a small value for F0

Instance Representation of a Candidate Partition

2. Method-based features

   - encode the identity of the coreference resolver that generated the candidate partition
   - one binary feature representing each of the 54 resolvers
     - feature value is 1 if the corresponding resolver generated the partition and 0 otherwise
   - could be useful if some resolvers perform consistently better than the others

Computing the Rank Value

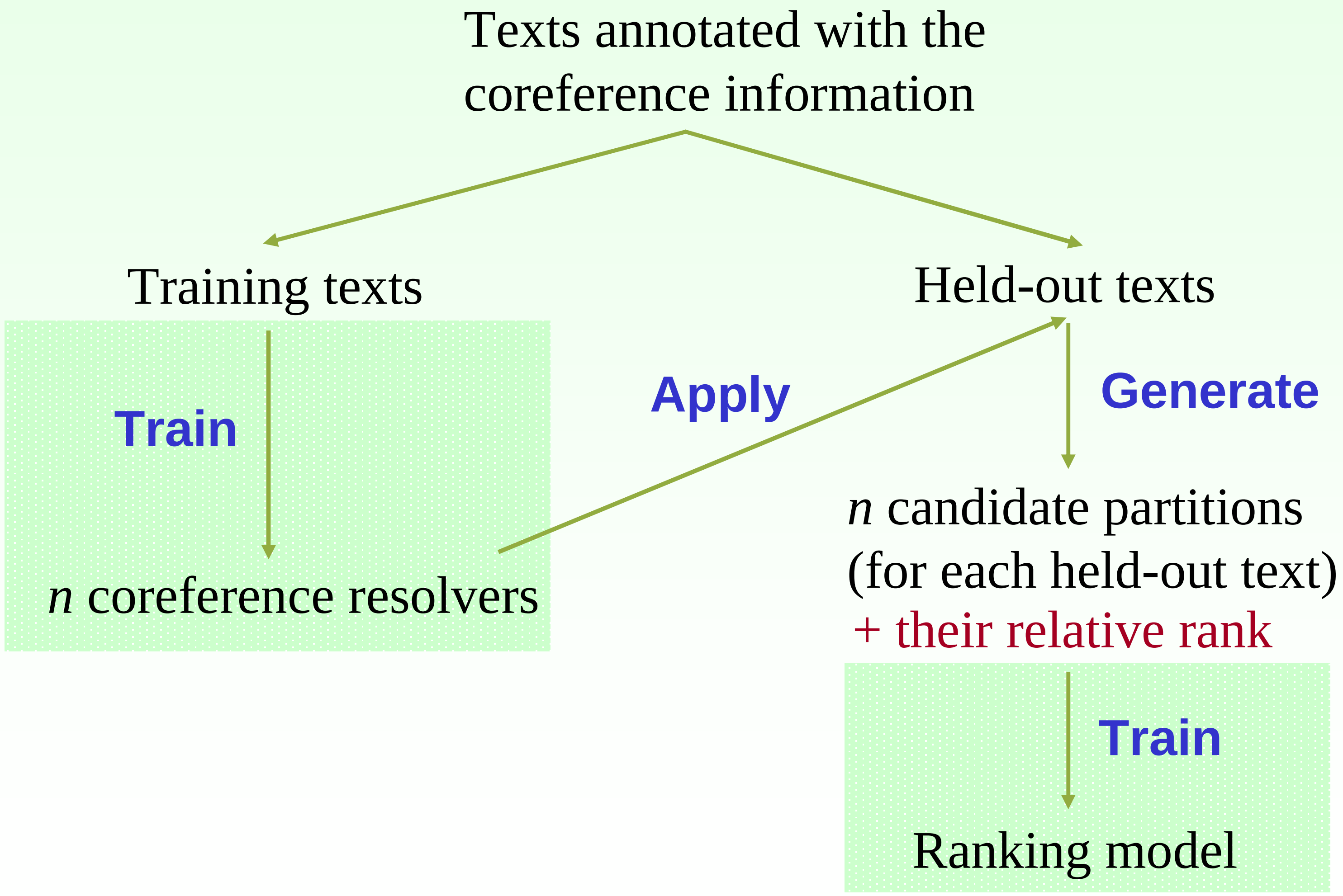
Task: Given a set of of 54 candidate partitions, compute the rank of each partition

Method:

1. Score each candidate partition using the target coreference scoring program
2. Assign rank *i* to the *i*-th lowest-scoring partition

Learning algorithm learns a model that assigns a higher rank to a higher-scoring candidate partition

Overview of the Training Procedure



Texts annotated with the coreference information

Training texts

Held-out texts

Train

n coreference resolvers

Apply

Generate

n candidate partitions (for each held-out text)

+ their relative rank

Train

Ranking model

Applying the Ranking Approach

Given a test text,

1. extract the NPs
2. generate 54 candidate NP partitions as in training
3. rank the candidate partitions using the ranking model
4. select the top-ranked partition to be the final partition

Plan for the Talk

- Noun phrase coreference resolution
  - standard machine learning approach
  - why the standard approach is not robust
- A ranking approach to coreference
- Evaluation

Evaluation

- The ACE coreference corpus
  - 3 data sets (Broadcast News, Newspaper, Newswire)
  - each data set comprises a training set and a test set
- NPs extracted automatically
- MUC scoring program (Vilain et al., 1995)
  - recall, precision, F-measure

Baseline Systems

Duplicated Soon et al. (2001)

- Decision tree learner (C4.5)
- Soon’s training instance creation method
- Soon’s feature set
- Closest-first clustering

Ng and Cardie (2002)

- RIPPER
- Ng and Cardie’s training instance creation method
- Ng and Cardie’s feature set
- Best-first clustering

Results (Baseline Systems)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**

Results (Baseline Systems)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**

Results (Baseline Systems)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**

Results (Baseline Systems)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**

Results (Baseline Systems)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**

Experiments with the Ranking Framework

- ½ of training texts for training coreference resolvers; ½ for training the ranking model

Results (Ranking Framework)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**

- 4-7% increase in F-measure over the Ng and Cardie baseline
- simultaneous increase in recall and precision

Ranking Experiment 1: Random Ranking

- Is supervised ranking necessary?
  - if all of the candidate partitions are “good”, supervised ranking may not be important

- Apply a random ranking model
  - randomly chooses a candidate partition for each test text

Results (Random Ranking Model)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Random ranking model	48.6	54.8	**51.5**	57.4	63.3	**60.2**	40.3	44.3	**42.2**

Results (Random Ranking Model)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Random ranking model	48.6	54.8	**51.5**	57.4	63.3	**60.2**	40.3	44.3	**42.2**

- supervised ranker outperforms random ranker by 9-13% in F-measure

Ranking Experiment 2: Perfect Ranking

- Is the supervised ranker performing at its upper limit?
  - further performance improvements beyond this point would require enlarging the candidate set

- Apply a perfect ranking model
  - uses an oracle to choose the best candidate partition for each test text

Results (Perfect Ranking Model)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Random ranking model	48.6	54.8	**51.5**	57.4	63.3	**60.2**	40.3	44.3	**42.2**
Perfect ranking model	66.0	69.3	**67.6**	70.4	71.2	**70.8**	56.6	59.7	**58.1**

Results (Perfect Ranking Model)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Random ranking model	48.6	54.8	**51.5**	57.4	63.3	**60.2**	40.3	44.3	**42.2**
Perfect ranking model	66.0	69.3	**67.6**	70.4	71.2	**70.8**	56.6	59.7	**58.1**

- Supervised ranker underperforms the perfect ranker by 1-3% in F-measure

Additional Results

- More experiments with the ranking model
- Results using the B-CUBED scoring program [Bagga and Baldwin, 1998]

Related Work

- Statistical relational learning (Getoor et al., MLJ 2001; Taskar et al., UAI 2002) for
  - proper name coreference (Pasula et al., NIPS 2002; McCallum and Wellner, IJCAI 2003)
  - information extraction (Bunescu and Mooney, ACL 2004)

- Supervised clustering
  - Daume and Marcu (JMLR, accepted), Li and Roth (CoNLL 2005), Finley and Joachims (ICML 2005)

Related Work

- Statistical relational learning (Getoor et al., MLJ 2001; Taskar et al., UAI 2002) for
  - proper name coreference (Pasula et al., NIPS 2002; McCallum and Wellner, IJCAI 2003)
  - information extraction (Bunescu and Mooney, ACL 2004)
- Supervised clustering
  - Daume and Marcu (JMLR, accepted), Li and Roth (CoNLL 2005), Finley and Joachims (ICML 2005)
- Multi-task bootstrapping for information extraction
  - Riloff and Jones (AAAI 1999), Wellner et al. (UAI 2004), Mann and Yarowsky (ACL 2005)

Ranking Experiment 3: Feature Contribution

- Are both partition-based features and method-based features useful for ranking partitions?
  - apply each type of features in isolation to re-train the ranking model

Results (Ranking Framework)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Partition-based features only	54.5	55.5	**55.0**	66.3	63.0	**64.7**	50.7	51.2	**51.0**
Method-based features only	62.0	68.5	**65.1**	67.5	61.2	**64.2**	51.1	49.9	**50.5**

Results (Ranking Framework)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Partition-based features only	54.5	55.5	**55.0**	66.3	63.0	**64.7**	50.7	51.2	**51.0**
Method-based features only	62.0	68.5	**65.1**	67.5	61.2	**64.2**	51.1	49.9	**50.5**

Results (Ranking Framework)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Partition-based features only	54.5	55.5	**55.0**	66.3	63.0	**64.7**	50.7	51.2	**51.0**
Method-based features only	62.0	68.5	**65.1**	67.5	61.2	**64.2**	51.1	49.9	**50.5**

- using either type of features
  - yields weaker performance than using both types of features

Results (Ranking Framework)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Partition-based features only	54.5	55.5	**55.0**	66.3	63.0	**64.7**	50.7	51.2	**51.0**
Method-based features only	62.0	68.5	**65.1**	67.5	61.2	**64.2**	51.1	49.9	**50.5**

- using either type of features
  - yields weaker performance than using both types of features
  - but still gives better results than both baselines

Results (Ranking Framework)

	Broadcast News			Newspaper			Newswire		
System Variation	R	P	F	R	P	F	R	P	F
Duplicated Soon et al. baseline	52.7	47.5	**50.0**	63.3	56.7	**59.8**	48.7	40.9	**44.5**
Ng and Cardie baseline	56.5	58.6	**57.5**	57.1	68.0	**62.1**	43.1	59.9	**50.1**
Ranking framework	62.2	67.9	**64.9**	67.4	71.4	**69.3**	50.1	60.3	**54.7**
Partition-based features only	54.5	55.5	**55.0**	66.3	63.0	**64.7**	50.7	51.2	**51.0**
Method-based features only	62.0	68.5	**65.1**	67.5	61.2	**64.2**	51.1	49.9	**50.5**

- using either type of features
  - yields weaker performance than using both types of features
  - but still gives better results than both baselines
- for Broadcast News, the method-based features alone are strongly predictive of good partitions

Summary

- Evaluated four combinations of
  - local vs. global optimization and
  - constraint-based vs. feature-based representation

  of anaphoricity information in terms of their effectiveness in improving a learning-based coreference system

- Showed that the constraint-based, globally-optimized approach is the most effective