Human Language Technology Research Institute

Beyond OntoNotes Coreference

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas

- Coreference-annotated corpora such as OntoNotes, MUC, and ACE have played a crucial role in coreference research
  - enabled the development of corpus-based approaches to coreference resolution in the past two decades
- But… they have somewhat limited people's attention to the coreference task they define

Entity Coreference Resolution

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Entity Coreference Resolution

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

- Inherently a clustering task

Entity Coreference Resolution

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

- Typically recast as the problem of selecting an antecedent for each mention

Entity Coreference Resolution

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

- Typically recast as the problem of selecting an antecedent for each mention
  - Does Queen Elizabeth have a preceding mention coreferent with it?

Entity Coreference Resolution

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

- Typically recast as the problem of selecting an antecedent for each mention
  - Does her have a preceding mention coreferent with it?

Entity Coreference Resolution

Identify the noun phrases (or *entity mentions*) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

- Typically recast as the problem of selecting an antecedent for each mention
  - Does husband have a preceding mention coreferent with it?

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

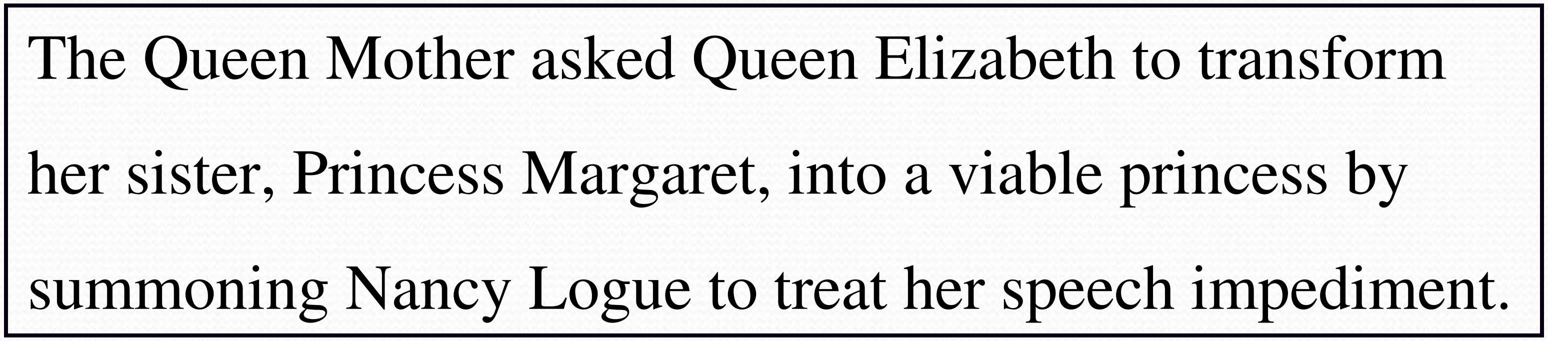
How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

How difficult is this task?

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Peter Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

A slightly different example



The Queen Mother asked Queen Elizabeth to transform her sister, Princess Margaret, into a viable princess by summoning Nancy Logue to treat her speech impediment.

A slightly different example

The Queen Mother asked Queen Elizabeth to transform her sister, Princess Margaret, into a viable princess by summoning Nancy Logue to treat her speech impediment.

A slightly different example

The Queen Mother asked Queen Elizabeth to transform her sister, Princess Margaret, into a viable princess by summoning Nancy Logue to treat her speech impediment.

A slightly different example

The Queen Mother asked Queen Elizabeth to transform her sister, Princess Margaret, into a viable princess by summoning Nancy Logue to treat her speech impediment.

A slightly different example

The Queen Mother asked Queen Elizabeth to transform her sister, Princess Margaret, into a viable princess by summoning Nancy Logue to treat her speech impediment.

Does this pronoun refer to The Queen Mother or Queen Elizabeth?

A slightly different example

The Queen Mother asked Queen Elizabeth to transform her sister, Princess Margaret, into a viable princess by summoning Nancy Logue to treat her speech impediment.

Does this pronoun refer to Princess Margaret or Nancy Logue?

How difficult is this task?

- Some coreference relations are more difficult to identify than the others
  - The difficult ones cannot be handled using grammatical constraints such as number and gender agreement
    - they typically require world knowledge and inference to identify
  - But… state-of-the-art coreference systems employ heuristics rather than world knowledge to handle the difficult cases
    - the difficult cases are rarely the focus of these systems

Plan for the Talk

- Solving hard entity coreference problems
  - Difficult cases of overt pronoun resolution
  - Zero pronoun resolution

Plan for the Talk

- Solving hard entity coreference problems
  - Difficult cases of overt pronoun resolution
  - Zero pronoun resolution

Hard-to-resolve Definite Pronouns

- Resolve definite pronouns for which traditional linguistic constraints on coreference and commonly-used resolution heuristics would not be useful

A Motivating Example (Winograd, 1972)

- The city council refused to give the demonstrators a permit because they feared violence.

- The city council refused to give the demonstrators a permit because they advocated violence.

Another Motivating Example (Hirst, 1981)

- When Sue went to Nadia's home for dinner, she served sukiyaki au gratin.

- When Sue when to Nadia's home for dinner, she ate sukiyaki au gratin.

Another Example

- James asked Robert for a favor, but he refused.
- James asked Robert for a favor, but he was refused.

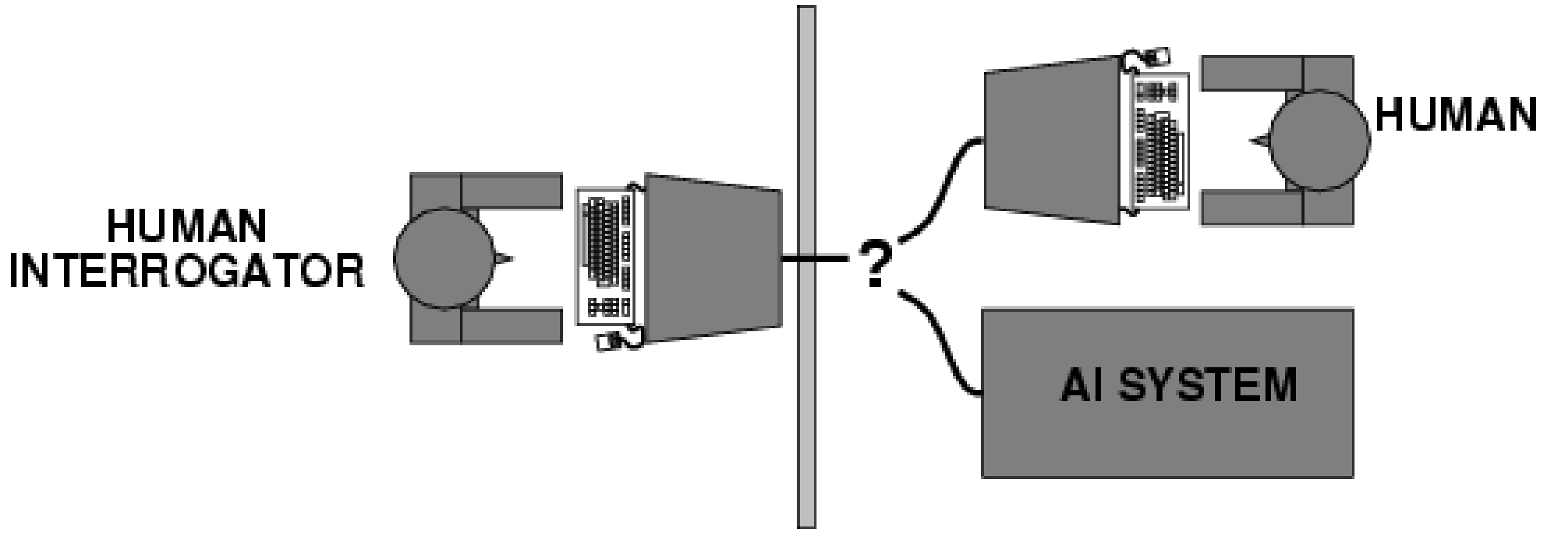
Broader Implications

- The ability to resolve such difficult pronouns has broader implications in artificial intelligence

The Turing Test

Alan Turing

- Turing (1950) "Computing machinery and intelligence"
- Operational test for intelligent behavior: the Imitation Game



Turing’s Prediction

- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes

Criticisms about the Turing Test

- There are tricks that can be used to fool the human interrogator

An appealing alternative to Turing Test

- Proposed by Levesque (2001)
  - resolve difficult pronouns in the twin sentences

Rahman & Ng EMNLP'12

Focus on difficult pronouns appearing in certain kinds of sentences

- The target pronoun should
  - appear in a sentence that has two clauses with a discourse connective, where the first clause contains two candidate antecedents and the second contains the pronoun
  - agree in gender, number, semantic class with both candidates
- We ensure that each sentence has a twin. Two sentences are twins if
  - their first clauses are the same
  - they have lexically identical pronouns with different antecedents

When Sue went to Nadia's home for dinner, she served sukiyaki au gratin.

When Sue when to Nadia's home for dinner, she ate sukiyaki au gratin.

Dataset

- 941 sentence pairs composed by 30 students who took my undergraduate machine learning class

Approach: Machine Learning via Ranking

- Create one ranking problem from each sentence
  - Each ranking problem consists of two instances
    - one formed from the pronoun and the first candidate
    - one formed from the pronoun and the second candidate

When Sue went to Nadia’s home for dinner, she served sukiyaki au gratin.

  - One instance created between she and Sue
  - One instance created between she and Nadia

- **Goal**: train a ranker that assigns a higher rank to the instance having the correct antecedent for each ranking problem

Eight Components for Deriving Features

- Narrative chains
- Google
- FrameNet
- Semantic compatibility
- Heuristic polarity
- Machine-learned polarity
- Connective-based relations
- Lexical features

Narrative Chains (Chambers & Jurafsky, 2008)

- Narrative chains are learned versions of **scripts**
  - Scripts represent knowledge of stereotypical event sequences that can aid text understanding
    - Reach restaurant, waiter sits you, gives you a menu, order food,..

- Partially ordered sets of events centered around a protagonist
  - e.g., **reach-s sit-o give-o order-s**
  - e.g., **borrow-s invest-s spend-s pay-s raise-s lend-s**
    - Someone who borrows something may invest, spend, pay, or lend it
  - can contain a mix of “s” (subject role) and “o” (object role)
    - e.g., the restaurant script

How can we apply narrative chains for pronoun resolution?

Ed punished Tim because he tried to escape.

1) Find the event in which the pronoun participates and its role
   - “he” participates in the “try” and “escape” events as a subject
2) Find the event(s) in which the candidates participate
   - Both candidates participate in the “punish” event
3) Pair each candidate event with each pronoun event
   - Two pairs are created: (punish, try-s), (punish, escape-s)
4) For each pair, extract chains containing both elements in pair
   - One chain is extracted, which contains punish-o and escape-s
5) Obtain role played by pronoun in the candidate’s event: object
6) Find the candidate that plays the extracted role: Tim

- Creates a binary feature that encodes this heuristic decision

Search Engine (Google)

Lions eat zebras because they are predators.

1) Replace the target pronoun with a candidate antecedent

Lions eat zebras because lions are predators.

Lions eat zebras because zebras are predators.

2) Generate search queries based on lexico-syntactic patterns

- Four search queries for this example: “lions are”, “zebras are”, “lions are predators”, “zebras are predators

3) Create features where the query counts obtained for the two candidate antecedents are compared

FrameNet

John killed Jim, so he was arrested.

- Both candidates are names, so search queries won’t return useful counts.
- Solution: before generating search queries, replace each name with its FrameNet semantic role
  - “John” with “killer”, “Jim” with “victim”
  - Search “killer was arrested”, “victim was arrested”, …

Semantic Compatibility

- Same as what we did in the Search Engine component, except that we obtain query counts from the Google Gigaword corpus

Heuristic Polarity

Ed was defeated by Jim in the election although he is more popular.

Ed was defeated by Jim in the election because he is more popular.

- Use polarity information to resolve target pronouns in sentences that involve comparison

1) Assign rank values to the pronoun and the two candidates
   - In first sentence, “Jim” is better, “Ed” is worse, “he” is worse
   - In second sentence, “Jim” is better, “Ed” is worse, “he” is better

2) Resolve pronoun to the candidate that has the same rank value as the pronoun

- Create features that encode this heuristic decision and rank values

Machine-Learned Polarity

- Hypothesis
  - rank values could be computed more accurately by employing a sentiment analyzer that can capture contextual information

- Same as Heuristic Polarity, except that OpinionFinder (Wilson et al., 2005) is used to compute rank values

Connective-Based Relations

Google bought Motorola because they are rich.

- To resolve “they”, we

  1) Count number of times the triple <“buy”, “because”, “rich”> appears in the Google Gigaword corpus

  2) If count is greater than a certain threshold, resolve pronoun to candidate that has the same deep grammatical role as pronoun

  3) Generate feature based on this heuristic resolution decision

Lexical Features

- Exploit information in the coreference-annotated training texts

- Antecedent-independent features
  - Unigrams
  - Bigrams (pairing word before connective and word after connective)
  - Trigrams (augmenting each bigram with connective)

- Antecedent-dependent features
  - pair a candidate's head word with
    - its governing verb
    - its modifying adjective
    - the pronoun's governing verb
    - the pronoun's modifying adjective

Evaluation

- **Dataset**
  - 941 annotated sentence pairs (70% training; 30% testing)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Evaluation metrics**: percentages of target pronouns that are
  - correctly resolved
  - incorrectly resolved
  - unresolved (no decision)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Unadjusted Scores**
  - Raw scores computed based on a resolver's output

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Unadjusted Scores**
  - Raw scores computed based on a resolver’s output
- **Adjusted Scores**
  - “Force” a resolver to resolve every pronoun by probabilistically assuming that it gets half of the unresolved pronouns right

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Three baseline resolvers**
  - Stanford resolver (Lee et al., 2011)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Three baseline resolvers**
  - Stanford resolver (Lee et al., 2011)
  - Baseline Ranker: same as our ranking approach, except that ranker is trained using the 39 features from Rahman & Ng (2009)

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- **Three baseline resolvers**
  - Stanford resolver (Lee et al., 2011)
  - Baseline Ranker: same as our ranking approach, except that ranker is trained using the 39 features from Rahman & Ng (2009)
  - The Combined resolver combines Stanford and Baseline Ranker:
    - Baseline Ranker is used only when Stanford can’t make a decision

Results

	Unadjusted Scores			Adjusted Scores		
	Correct	Wrong	No Dec.	Correct	Wrong	No Dec.
Stanford	40.07	29.79	30.14	55.14	44.86	0.00
Baseline Ranker	47.70	47.16	5.14	50.27	49.73	0.00
Combined resolver	53.49	43.12	3.39	55.19	44.77	0.00
Our system	**73.05**	26.95	0.00	**73.05**	26.95	0.00

- Stanford outperforms Baseline ranker
- Combined resolver does not outperform Stanford
- Our system outperforms Stanford by 18 accuracy points

Ablation Experiments

- Remove each of the 8 components one at a time
- Accuracy drops significantly (paired t-test, $p < 0.05$) after each component is removed
- Most useful: Narrative chains, Google, Lexical Features
- Least useful: FrameNet, Learned Polarity

Summary

- There is a recent surge of interest in these hard, but incredibly interesting pronoun resolution tasks

- They could serve as an alternative to the Turing Test (Levesque, 2011)
  - This challenge is known as the Winograd Schema Challenge
    - Announced as a shared task in AAAI 2014
    - Sponsored by Nuance

- Details can be found in the Rahman & Ng EMNLP 2012 paper

Plan for the Talk

- Solving hard entity coreference problems
    - Difficult cases of overt pronoun resolution
    - Zero pronoun resolution

What is a zero pronoun?

- A zero pronoun (ZP) is a gap in a sentence
  - found when a phonetically null form is used to refer to an entity

- An anaphoric zero pronoun (AZP) is a ZP that is anaphoric
  - a ZP that corefers with one or more preceding NPs in the text

俄罗斯作为米洛舍夫维奇一贯的支持者，
*pro***曾**经提出调停这场政治危机。

Russia is a consistent support of Milosevic,
pro has proposed to mediate the political crisis.

Zero Pronoun Resolution

- is the task of finding an antecedent for each AZP

- is an important task
  - AZPs are present in many languages
    - e.g., Chinese, Polish, Korean, Japanese, Spanish, Italian, …

- is more challenging than overt pronoun resolution
  - ZPs lack grammatical attributes useful for overt pronoun resolution such as gender and number

Zero Pronoun Resolution

- Typically composed of two steps
  - **AZP identification**
    - Extract from a text all the ZPs that are anaphoric
  - **AZP resolution**
    - Identify an antecedent of an AZP

In this talk,we will focus on the second step: **resolution**

- State of the art AZP resolvers: supervised approach
  - Train one classifier for AZP identification and another one for AZP resolution

But…

- except for a handful of the world’s natural languages, training data with manually resolved AZPs are not available

  → supervised approaches cannot be applied to these languages

Goal

- Develop unsupervised approaches to AZP resolution
    - Facilitate application to resource-poor languages

- Two unsupervised approaches
    - **Idea**:
        - Train an overt pronoun resolution model
        - Apply the resulting model to resolve AZPs
    - **Hypothesis**:

Overt and zero pronouns have similar linguistic properties, so knowledge learned from one can be applied to the other

First Unsupervised Approach to AZP Resolution

- Recast unsupervised ZP resolution as supervised ranking
  - Train a ranker in a supervised manner to resolve overt pronouns
  - Apply the resulting ranker to resolve zero pronouns
- Requires training data with manually resolved **overt** pronouns
  - But it is unsupervised in the sense that it does **not** require training data with manually resolved zero pronouns

Training the Ranking Model

- **Goal**: rank the candidate antecedents of an anaphoric overt pronoun so that its correct antecedents are ranked higher

Training the Ranking Model (Cont')

- Training instance creation
  - Each instance represents an anaphoric overt pronoun (***op***) and one of its candidate antecedents (***c***).
  - The set of instances created from the same anaphoric overt pronoun constitutes a ranking problem

$$rank\ value = \begin{cases} \dfrac{1}{|coref|}, & \text{if } op \text{ is coreferent with } c \\ 0, & \text{otherwise} \end{cases}$$

where |coref| is the number of correct antecedents ***op*** has.

Employ Chen & Ng's (2013) features plus other features

Train the ranker using Maximum Entropy

Training the Ranking Model (Cont')

- model trained on 10 Chinese overt pronouns
  - each pronoun is uniquely identified by 4 grammatical attributes

Pronoun	Number	Gender	Person	Animacy
我 (I)	singular	neuter	first	animate
你 (you)	singular	neuter	second	animate
他 (he)	singular	masculine	third	animate
她 (she)	singular	feminine	third	animate
它 (it)	singular	neuter	third	inanimate
你们 (you)	plural	neuter	second	animate
我们 (we)	plural	neuter	first	animate
他们 (they)	plural	masculine	third	animate
她们 (they)	plural	feminine	third	animate
它们 (they)	plural	neuter	third	inanimate

Applying the Ranking Model to resolve AZPs

- Problem: ranking model is trained on overt pronouns, so it can only be applied to resolve overt pronouns

What can we do?

- Idea: Fill the gap of each zero pronoun ***zp*** in the test set with each possible overt pronoun ***op***

Applying the Ranking Model to resolve AZPs

- Test instance creation
  - For each anaphoric ***zp***, create a set of test instances from **zp**
    - a test instance is created by pairing one of ***zp***'s candidate antecedents ***c*** and one of the 10 overt pronouns ***op***
      - Total no. of instances: 10 * number of candidate antecedents

- Antecedent selection
  - Select as ***zp***'s antecedent the ***c*** in the instance having the highest probability according to the ranker
    - simultaneously selecting both the antecedent and the overt pronoun for filling ***zp***'s gap

Are we done?

- We now have an unsupervised approach for AZP resolution
  - the ranking model trained on overt pronouns is applied to resolve AZPs
- But… can we improve it?

Yes! But how?

- Observation: When applying the ranker, we exhaustively fill the ZP gap with each possible overt pronoun
  - **Problem**: the ranker doesn't care whether the overt pronoun that fills the gap is compatible with the ZP's governing verb

pro **dresses up**

pro **cries**

So…

- The governing verb has its **probabilistic** preference for the overt pronouns that should fill the ZP’s gap

- But the ranking model also has its **probabilistic** preference for the overt pronouns that should fill the ZP’s gap

If their preferences aren’t the same, what should we do?

As humans, we know which overt pronouns a governing verb prefers, but how would a machine know??

Unsupervised Learning of Selectional Preferences

For each verb ***v*** in the test set,

- collect from the Chinese Gigaword corpus the set of NPs serving as the subject of ***v***
- heuristically determine each NP's **Gender**, **Number**, **Animacy**, and **Person**
- compute ***v***'s grammatical preferences

哭 (cry)	**Gender**	**Number**	**Animacy**	**Person**
小薇 (Mary)	Female	Singular	Animate	n/a
他 (he)	Male	Singular	Animate	n/a
她 (she)	Female	Singular	Animate	n/a

Unsupervised Learning of Selectional Preferences

For each verb ***v*** in the t

- collect from the Chi serving as the subje
- heuristically determi **Animacy**, and **Pers**
- compute ***v***'s grammatical preferences

<u>哭 (cry)</u>

Gender: 0.33 Male, 0.66 Female

Number: 1.0 Singular, 0.0 Plural

Animacy: 1.0 Animate, 0.0 Inanimate

Person: 1.0 n/a, 0.0 1st, 0.0 2nd, 0.0 3rd

<u>哭 (cry)</u>	**Gender**	**Number**	**Animacy**	**Person**
小薇 (Mary)	Female	Singular	Animate	n/a
他 (he)	Male	Singular	Animate	n/a
她 (she)	Female	Singular	Animate	n/a

So…

- The governing verb has its **probabilistic** preference for the overt pronouns that should fill the ZP's gap

- But the ranking model also has its **probabilistic** preference for the overt pronouns that should fill the ZP's gap

If their preferences aren't the same, what should we do?

As humans, we know which overt pronouns a governing verb prefers, but how would a machine know??

Idea

- Combine their preferences into one objective function
  - Maximize both of their preferences (by maximizing the objective function) subject to the constraint that they must end up choosing the same pronoun to fill the gap
  - **How**? Use Integer Linear Programming (ILP)
    - ILP is a **constrained optimization** framework
    - Create one ILP program for each AZP

$$\operatorname*{argmax}_{op,c}[\sum_{op\in PR}\sum_{c\in C}P(op,c)x(op,c)+\ \alpha\sum_{b\in V_{Num}}P_{Num}(b)y_{Num}(b)+$$

$$\beta\sum_{b\in V_{Gen}}P_{Gen}(b)y_{Gen}(b)+\ \gamma\sum_{b\in V_{Per}}P_{Per}(b)y_{Per}(b)+\delta\sum_{b\in V_{Ani}}P_{Ani}(b)y_{Ani}(b)] \tag{1}$$

subject to the following constraints:

$$x(op,c)\in\{0,1\},\forall op\in PR,\forall c\in C \tag{2}$$

$$\sum_{op\in PR}\sum_{c\in C}x(op,c)=1 \tag{3}$$

$$y_a(b)\in\{0,1\},\forall a\in A,\forall b\in V_a \tag{4}$$

$$\sum^{|a|}y_a(b)=1,\forall a\in A \tag{5}$$

$$y_{Ani}(animate)\geq y_{Gen}(masculine)+y_{Gen}(feminine) \tag{6}$$

$$\sum_{op_a=b}\sum_{c\in C}x(op,c)=y_a(b),\forall a\in A,\forall b\in V_a \tag{7}$$

Experimental Setup

- Corpus
  - Chinese portion of the OntoNotes 5.0 corpus
  - Supervised training of the overt pronoun resolution model
    - Chinese training set used in the CoNLL 2012 shared task
    - 1,391 documents (13,418 overt pronouns)
  - Testing (Applying the model to resolve gold AZPs)
    - Chinese development set used in the CoNLL 2012 shared task
    - 172 documents (1,713 AZPs)

- Evaluation measures
  - recall (R), precision (P), F-measure (F) on resolving **gold AZPs**

Baseline Systems

- 3 state-of-the-art Chinese AZP resolvers

- Zhao & Ng (2007)
  - trained a pairwise model using a decision tree learner

- Kong & Zhou (2010)
  - employed a syntactic parse tree as a structured feature
  - trained a SVM classifier using a convolution tree kernel

- Chen & Ng (2013)
  - extended Zhao & Ng's feature set with new contextual features
  - achieved state-of-the-art results on the OntoNotes 5.0 dataset
    - F-score of 47.7% for resolving **gold** AZPs

Results

		R	P	F
Baseline Systems	Duplicated Zhao and Ng (2007)	41.5	41.5	**41.5**
	Duplicated Kong and Zhou (2010)	44.9	44.9	**44.9**
	Chen and Ng (2013)	47.7	47.7	**47.7**
Our Approach	Ranking model	45.9	46.4	**46.1**
	Ranking model + ILP	48.4	48.9	**48.7**

- The best baseline is Chen & Ng (2013)

Results

		R	P	F
Baseline Systems	Duplicated Zhao and Ng (2007)	41.5	41.5	**41.5**
	Duplicated Kong and Zhou (2010)	44.9	44.9	**44.9**
	Chen and Ng (2013)	47.7	47.7	**47.7**
Our Approach	Ranking model	45.9	46.4	**46.1**
	Ranking model + ILP	48.4	48.9	**48.7**

- Ranking model underperforms Chen & Ng (2013) but outperforms the other two baselines
- After adding ILP, ranking model outperforms Chen & Ng (2013)
  - Unsupervised approach outperforming supervised approach

Should we be happy with the Ranking Model + ILP approach?

- **Advantage**: Unsupervised
  - Does not require training data with manually resolved AZPs

- **Disadvantage**:
  - Requires training data with manually resolved overt pronouns

  → Cannot be applied to languages for which such annotated data is not readily available

Can we train the overt pronoun resolution model in an unsupervised manner?

Second Unsupervised Approach to AZP Resolution

- Same as the first unsupervised approach, but…
  - Train an unsupervised overt pronoun resolution model
  - Apply the resulting model to resolve zero pronouns

- A **very challenging** setting

How can we train an overt pronoun resolution model in an unsupervised manner?

If we had annotated training data …

- we could adopt the standard supervised approach:
  - Train a coreference model to determine the probability that an overt pronoun p and a candidate antecedent c given their context k are coreferent, i.e., P(coref=+|p,c,k)

[玛丽]告诉[约翰][她]非常喜欢[他]。

[Mary] told [John] that [she] really likes [him].

Training Instances:

coref?	Overt Pronoun	Candidate Antecedent
No	she	John
Yes	she	Mary
No	he	she
Yes	he	John
No	he	Mary

If we had annotated training data …

- we could adopt the standard supervised approach:
  - Train a coreference model to determine the probability that an overt pronoun p and a candidate antecedent c given their context k are coreferent, i.e., P(coref=+|p,c,k)
  - Apply the model to each overt pronoun to select the candidate with the highest probability as its antecedent

But we don’t have annotated data…

coref?	Overt Pronoun	Candidate Antecedent
?	she	John
?	she	Mary
?	he	she
?	he	John
?	he	Mary

- **Idea**: design a generative model and use EM to iteratively
  - Fill in missing values probabilistically (**E-step**)
    - i.e., determine the probability each pair of mentions is coreferent

But we don’t have annotated data…

coref?	Overt Pronoun	Candidate Antecedent
0.3	she	John
0.8	she	Mary
0.7	he	she
0.6	he	John
0.7	he	Mary

- **Idea**: design a generative model and use EM to iteratively
  - Fill in missing values probabilistically (**E-step**)
    - i.e., determine the probability each pair of mentions is coreferent

But we don’t have annotated data…

coref?	Overt Pronoun	Candidate Antecedent
0.3	she	John
0.8	she	Mary
0.7	he	she
0.6	he	John
0.7	he	Mary

- **Idea**: design a generative model and use EM to iteratively
  - Fill in missing values probabilistically (**E-step**)
    - i.e., determine the probability each pair of mentions is coreferent
  - Estimate the model parameters using the filled values (**M-step**)

Generative Model

- use the generative model to fill in the missing class values
  - i.e., compute P(coref=+|p,c,k)

p: pronoun
c: candidate antecedent
k: their context

- Using Chain Rule,

$$P(coref=+\mid p,c,k)=\frac{P(p,c,k,coref=+)}{P(p,c,k)}$$

- Applying Chain Rule to the numerator,

$$P(p,c,k,coref=+)$$
$$=P(k)P(c\mid k)P(coref=+\mid c,k)P(p\mid coref=+,c,k)$$

This is our generative model!

Generative Model

p: pronoun
c: candidate antecedent
k: their context

$$P(p, c, k, coref = +)$$

$$= P(k)P(c \mid k)P(coref = + \mid c, k)P(p \mid coref = +, c, k)$$

Generative Model

p: pronoun
c: candidate antecedent
k: their context

$$P(p,c,k,coref=+)$$

$$=P(k)P(c\,|\,k)P(coref=+\,|\,c,k)P(p\,|\,coref=+,c,k)$$

generate
context k

Generative Model

p: pronoun
c: candidate antecedent
k: their context

$$P(p,c,k,coref=+)$$

$$=P(k)P(c\,|\,k)P(coref=+\,|\,c,k)P(p\,|\,coref=+,c,k)$$

generate
candidate c
given context k

Generative Model

p: pronoun
c: candidate antecedent
k: their context

$$P(p,c,k,coref=+)$$

$$=P(k)P(c\,|\,k)P(coref=+\,|\,c,k)P(p\,|\,coref=+,c,k)$$

↑

generate class label
given candidate c
and context k

Generative Model

p: pronoun
c: candidate antecedent
k: their context

$$P(p, c, k, coref = +)$$

$$= P(k)P(c \mid k)P(coref = + \mid c, k)P(p \mid coref = +, c, k)$$

generate pronoun p given class label, candidate c and context k

How to estimate each of these parameters?

p: pronoun
c: candidate antecedent
k: their context

$$P(p,c,k,coref=+)$$

$$=P(k)P(c\,|\,k)P(coref=+\,|\,c,k)P(p\,|\,coref=+,c,k)$$

These four are the **model parameters**

How to estimate each of these parameters?

p: pronoun
c: candidate antecedent
k: their context

$$P(p, c, k, coref = +)$$

$$= \underset{\uparrow}{P(k)} P(c \mid k) P(coref = + \mid c, k) P(p \mid coref = +, c, k)$$

Simplifying assumption: for each pronoun, the contexts generated from different candidate antecedents have the same probability

- Effectively ignoring this term

How to estimate each of these parameters?

p: pronoun
c: candidate antecedent
k: their context

$$P(p, c, k, coref = +)$$

$$= P(k)P(c \mid k)P(coref = + \mid c, k)P(p \mid coref = +, c, k)$$

↑

Probability of a candidate antecedent c given context k

How to estimate this probability?

Simplifying assumption: given context k, the candidate antecedents are generated with the same probability

- Effectively ignoring this term

How to estimate each of these parameters?

p: pronoun
c: candidate antecedent
k: their context

$$P(p, c, k, coref = +)$$

$$= P(k)P(c \mid k)\underset{\uparrow}{P(coref = + \mid c, k)}P(p \mid coref = +, c, k)$$

Probability that they are coreferent given candidate & context

How to estimate this probability?

- Assumption: k is sufficient for determining coreference
  - So c is not needed → we can drop c from the condition
- represent context k using 8 features
  - grammatical: is c a subject with same governing verb as p?, ..
  - semantic: is c is closest candidate with subject role?
  - positional: is p the first word of a sentence?, …
  - distance: sentence distance between p and c

How to estimate each of these parameters?

p: pronoun
c: candidate antecedent
k: their context

$$P(p, c, k, coref = +)$$

$$= P(k)P(c \mid k)P(coref = + \mid c, k)P(p \mid coref = +, c, k)$$

↑

Probability that they are coreferent given candidate & context

How to estimate this probability?

- Assumption: k is sufficient for determining coreference
  - So c is not needed → we can drop c from the condition
- represent context k using 8 features
  - model parameters to be estimated in the M-step

How to estimate each of these parameters?

p: pronoun
c: candidate antecedent
k: their context

$$P(p, c, k, coref = +)$$

$$= P(k)P(c \mid k)P(coref = + \mid c, k)\boxed{P(p \mid coref = +, c, k)}$$

Probability of p given everything else

How to estimate $P(p \mid coref = +, c, k)$ **?**

- simplify by dropping k, yielding

$$P(p \mid coref = +, c)$$

- represent p by its four grammatical attribute values, yielding

$$P(p_{Gen}, p_{Num}, p_{Per}, p_{Ani} \mid coref = +, c)$$

- assume attribute values are independent given class value

$$P(p_{Gen} \mid coref = +, c_{Gen})P(p_{Num} \mid coref = +, c_{Num})P(p_{Per} \mid coref = +, c_{Per})P(p_{Ani} \mid coref = +, c_{Ani})$$

How to estimate each of these parameters?

p: pronoun
c: candidate antecedent
k: their context

$$P(p, c, k, coref = +)$$

$$= P(k)P(c \mid k)P(coref = + \mid c, k)\prod_{a \in A} P(p_a \mid coref = +, c_a)$$

Probability of p given everything else

How to estimate $P(p \mid coref = +, c, k)$ **?**

- simplify by dropping k, yielding

$$P(p \mid coref = +, c)$$

- represent p by its four grammatical attribute values, yielding

$$P(p_{Gen}, p_{Num}, p_{Per}, p_{Ani} \mid coref = +, c)$$

- assume attribute values are independent given class value

$$P(p_{Gen} \mid coref = +, c_{Gen})P(p_{Num} \mid coref = +, c_{Num})P(p_{Per} \mid coref = +, c_{Per})P(p_{Ani} \mid coref = +, c_{Ani})$$

How to estimate each of these parameters?

p: pronoun
c: candidate antecedent
k: t

Model parameters (to be estimated in M-step)

$$P(p, c, k, coref = +)$$

$$= P(k)P(c \mid k)P(coref = + \mid c, k)\prod_{a \in A} P(p_a \mid coref = +, c_a)$$

Probability of p given everything else

How to estimate $P(p \mid coref = +, c, k)$ **?**

- simplify by dropping k, yielding

$$P(p \mid coref = +, c)$$

- represent p by its four grammatical attribute values, yielding

$$P(p_{Gen}, p_{Num}, p_{Per}, p_{Ani} \mid coref = +, c)$$

- assume attribute values are independent given class value

$$P(p_{Gen} \mid coref = +, c_{Gen})P(p_{Num} \mid coref = +, c_{Num})P(p_{Per} \mid coref = +, c_{Per})P(p_{Ani} \mid coref = +, c_{Ani})$$

M-step

p: pronoun
c: candidate antecedent
k: their context

- **Goal**: given $P(coref = + | p, c, k)$, estimate model parameters:

$P(p_a | c_a, coref = +)$

$P(coref = + | k)$

use maximum likelihood estimation

Example

- Start by initializing model parameters to uniform values
- Run E-step using these model parameters
  - Before E-step

coref?	Overt Pronoun	Candidate Antecedent
?	she	John
?	she	Mary
?	he	she
?	he	John
?	he	Murcury

Example

- Start by initializing model parameters to uniform values
- Run E-step using these model parameters
  - After E-step

coref?	Overt Pronoun	Candidate Antecedent
Yes 0.3	she	John
Yes 0.8	she	Mary
Yes 0.2	he	she
Yes 0.9	he	John
Yes 0.1	he	Mary

- Run M-step to estimate model parameters from the probabilistically labeled data
- Iterate until convergence

Applying the model to resolve AZPs

- Given an AZP z,
  - exhaustively search for the candidate antecedent c and overt pronoun p that maximize $P(l=1 \mid p,c,k)$ when p is used to fill the gap left behind by z

- since the model is trained on overt pronouns but is applied to ZPs, we have to fill each ZP's gap with every overt pronoun when applying the model

Experimental Setup

- Corpus
  - Chinese portion of the OntoNotes 5.0 corpus
  - Unsupervised training of the overt pronoun resolution model
    - Chinese training set used in the CoNLL 2012 shared task
    - 1,391 documents (13,418 overt pronouns)
  - Testing (Applying the model to resolve AZPs)
    - Chinese development set used in the CoNLL 2012 shared task
    - 172 documents (1,713 AZPs)

- Evaluation measures
  - R, P, and F on resolving gold AZPs

Results

		R	P	F
Baseline Systems	Duplicated Zhao and Ng (2007)	41.5	41.5	**41.5**
	Duplicated Kong and Zhou (2010)	44.9	44.9	**44.9**
	Chen and Ng (2013)	47.7	47.7	**47.7**
Our Approach	Ranking model	45.9	46.4	**46.1**
	Ranking model + ILP	48.4	48.9	**48.7**
	Generative model	47.5	47.9	**47.7**

- Chen & Ng (2013) is the best baseline

Results

		R	P	F
Baseline Systems	Duplicated Zhao and Ng (2007)	41.5	41.5	**41.5**
	Duplicated Kong and Zhou (2010)	44.9	44.9	**44.9**
	Chen and Ng (2013)	47.7	47.7	**47.7**
Our Approach	Ranking model	45.9	46.4	**46.1**
	Ranking model + ILP	48.4	48.9	**48.7**
	Generative model	47.5	47.9	**47.7**

- Ranking+ILP outperforms Chen & Ng (2013)
  - Unsupervised approach outperforming supervised approach

Results

		R	P	F
Baseline Systems	Duplicated Zhao and Ng (2007)	41.5	41.5	**41.5**
	Duplicated Kong and Zhou (2010)	44.9	44.9	**44.9**
	Chen and Ng (2013)	47.7	47.7	**47.7**
Our Approach	Ranking model	45.9	46.4	**46.1**
	Ranking model + ILP	48.4	48.9	**48.7**
	Generative model	47.5	47.9	**47.7**

- Generative model achieves same F-score as Chen & Ng (2013)
  - Unsupervised approach rivaling supervised approach

Results

		R	P	F
Baseline Systems	Duplicated Zhao and Ng (2007)	41.5	41.5	**41.5**
	Duplicated Kong and Zhou (2010)	44.9	44.9	**44.9**
	Chen and Ng (2013)	47.7	47.7	**47.7**
Our Approach	Ranking model	45.9	46.4	**46.1**
	Ranking model + ILP	48.4	48.9	**48.7**
	Generative model	47.5	47.9	**47.7**

- Generative model underperforms Ranking Model+ILP
  - But it was trained without manually resolved overt pronouns
  - Generative process is language-independent

Major Sources of Error

- Failure in tracking the discourse entity in focus

[八里乡] 位于台北盆地西北端。
[行政区]隶属于 台北县，*pro*
为台北县廿九个乡镇市之一。

[Bali Town] is located in the Northwest of Taipei Basin.
[Its administrative area] is affiliated with Taipei County,
pro is one of Taipei County's 29 towns and cities.

Major Sources of Error

- Failure in tracking the discourse entity in focus
- Errors in computing semantic compatibility

[一支海军陆战队] 杀死了约 [24
名手无寸铁的 伊拉克人],
pro 包括妇女和六名儿童。

[Marines] killed about [24 unarmed Iraqis], *pro*
include women and six children.

Major Sources of Error

- Failure in tracking the discourse entity in focus
- Errors in computing semantic compatibility
- Assumption that overt pronouns and ZPs occur in the same context is not always correct

*pro*不客气。

pro are welcome.

Summary

- Examined hard coreference problems
  - resolution of difficult overt pronouns and zero pronouns
- These are incredibly interesting but challenging problems
  - particularly challenging if we want to develop weakly-supervised, language-independent models
  - and… they are far from being solved