**Natural Language Processing (NLP)**

**Session 10**

1. Discourse (in the field of Pragmatics)
   1. We’ve looked at syntax and semantics, but we have to look at language use in different situations pragmatics.
   2. Discourse is, approximately, sequences of sentences.
      1. Monologue (one speaker)
      2. Dialogue (multiple speakers)
   3. When we look at discourse, interesting challenges arise.
   4. One of these challenges is interpreting pronouns, such as he, she and it. Vital for many other tasks, such as automatic Question-Answering.
2. Discourse (Pragmatics)
   1. Meaning beyond the sentence
   2. Context-sensitivity
      1. (of meaning: not to be confused with context-sensitive grammars)
   3. In particular:
      1. Anaphora (including temporal)
         1. “He sold them to himself two days after that.”
      2. Co-reference
         1. “The legendarily brash, anti-establishment renegade …”
      3. Ellipsis
         1. “She promised she would; but she didn’t.”
      4. Very important in semantic tasks (e.g. IE, QA)
         1. Less important for others (e.g. document/sentiment classification) – why?
      5. Implicature:
         1. Could you pass the salt? Do you have the time?
         2. There’s a garage round the corner
         3. His handwriting is very neat.
3. Referring Expressions
   1. Five types of referring expression (RE):
      1. Indefinite noun phrases introduce new referents
         1. “I saw a nice car today. Some other people noticed it too”
      2. Definite noun phrases refer to identifiable referents
         1. Sometimes identifiable from previous mention
            1. “I saw a nice car today. I’d like to buy the car tomorrow.”
         2. Sometimes identifiable from world knowledge or description itself:
            1. “The Prime Minister is coming to tea.”
            2. “The car of the year is the Ford Dustpan.”
         3. Pronouns refer to highly salient referents
            1. “I saw a nice car today. I’d like to buy it tomorrow.”
            2. “She’s coming to tea.”
         4. Demonstratives refer to near or distant referents (literally or metaphorically)
            1. “This new car is much faster than that old one.”
         5. Names can refer to old or new referents
            1. “The Prime Minister will visit President Trump today, Mr Johnson’s office announced.”
4. Pronouns (Anaphora)
   1. Pronouns refer to contextual elements
   2. Anaphora: referring back to items already mentioned
      1. (antecedents: previous referring expressions)
      2. “Sue left her coat behind”
   3. Cataphora: referring forward to items to be mentioned
      1. “Before she leaves, Sue always checks her coat”
   4. Deixis: referring to the current environment
      1. “I want you to leave now”
   5. Pleonastic or generic uses:
      1. “It’s raining. Always happens when you least expect it.”
   6. Discourse and temporal reference:
      1. “We’ll be ready then. Well, that’s good.”
   7. Bound variables:
      1. “Every student has their particular preference.”
5. Pronoun Resolution: Constraints
   1. For referential (non-generic) uses, we must identify (resolve) the antecedent
      1. huge potential ambiguity: any previous RE is a candidate antecedent
   2. Many hard constraints on possible antecedents:
      1. Number
         1. (English: singular vs plural)
         2. “I have a dog and two cats. It is/They are very fluffy.”
      2. Person
         1. (English: 1st, 2nd, 3rd)
         2. “I saw you with John. You were / He was happy.”
      3. Gender
         1. (English: male, female, nonpersonal)
         2. “Sue met John and his dog. She/he/it was happy.”
      4. Binding
         1. (English: reflexives with clause subjects)
         2. “John thinks Bill likes him/himself.”
   3. (We often don’t even notice these potential ambiguities, as humans!)
6. Pronoun Resolution: Preferences
   1. And many softer preferences:
   2. Recency
      1. (more recent > less recent)
      2. “Sue lives in Reading. Jane lives in Havant. She has a dog.”
   3. Grammatical role
      1. (subject > object > other)
      2. “Sue knows Jane. She is coming today.”
   4. Repetition
      1. (more mentions > fewer mentions)
      2. “Sue is a banker. She works in the City. Jane likes her. She drives a Jaguar.”
   5. Parallelism
      1. “Sue has known Jane since 1989. Gretel has known her since 1982.”
   6. Discourse / event semantics
      1. “Sue is annoyed with Jane. She spilt her drink.”
      2. “Sue loves her dog. We took her for a walk.”
   7. (We are more likely to notice these ambiguities)
7. Pronoun Resolution
   1. So we can use these constraints in resolution
   2. Rule-based:
      1. e.g. using Centering Theory (Grosz et al, 1995)
      2. Order REs and possible antecedent REs by prominence
         1. e.g. subject > object > other
         2. “Forward-looking centres” (FLCs) = all REs in sentence
         3. “Backward-looking centres” (BLCs) = all REs mentioned in previous sentence
      3. Filter possible pairings using hard constraints
   3. Statistical classification:
      1. Supervised classification
      2. Potential pronoun-antecedent pairs as instances
      3. Features chosen to relate to constraints/preferences:
      4. number, gender, person match
      5. word/sentence/syntactic distance
      6. grammatical/semantic role & parallelism
8. Co-Reference Resolution
   1. More general version of the problem
   2. Build chains of all co-referring expressions
   3. More general version of the problem
      1. Mentions evoke (introduce) or access referents.
   4. Candidate REs identified by POS-tagging & parsing
      1. Need to identify links i.e. pair REs with antecedents
   5. Approach similar to pronoun resolution:
      1. But we now need to add other RE types
      2. Particularly definite noun phrases
   6. Supervised classification:
      1. (same approach as for pronoun resolution)
      2. Pairs of correct/incorrect examples
      3. Features:
         1. Features used for pronoun resolution, plus:
         2. Lexical similarity (e.g. RE/antecedent edit distance)
         3. Semantic co-reference (e.g. dates)
         4. Syntactic relation (e.g. apposition (“X, the Y, …”)
         5. POS/lexical type of head noun
         6. Big search space: use e.g. graph-based methods
         7. Efficient search, while finding optimal chains (not just independent pairs)
9. Anaphora resolution as Structure Learning
   1. So far we have only seen examples of text analytics applications in which the task was to label a SINGLE OBJECT
   2. In the case of anaphora resolution/coreference, the task is to label a STRUCTURE
      1. In its simplest form, the antecedent / anaphor pair (MENTION PAIR)
   3. This is an example of so-called STRUCTURED LEARNING
10. Factors in interpreting anaphoric expressions
    1. Factors:
       1. Morphological features (agreement)
       2. Syntactic information
       3. Salience
       4. Lexical and common-sense knowledge
    2. Distinction often made between CONSTRAINTS and PREFERENCES.
11. Agreement
    1. GENDER strong CONSTRAINT for pronouns (in other languages: for other anaphors as well)
       1. [Jane] blamed [Bill] because HE spilt the coffee (Ehrlich, Garnham e.a, Arnold e.a)
    2. NUMBER also strong constraint
       1. [[Union] representatives] told [the CEO] that THEY couldn’t be reached.
12. Problems to be resolved by an AR system: mention identification
    1. Effect: recall
    2. Typical problems:
       1. Nested NPs (possessives)
          1. [a city] 's [computer system] 🡪 [[a city]’s computer system]
       2. Appositions:
          1. [Madras], [India] 🡪 [Madras, [India]] •
       3. Attachments
13. Problems for AR: agreement extraction
    1. The committee are meeting / is meeting
    2. The Union sent a representative. They ….
    3. The doctor came to visit my father. SHE told him …
14. Problems for AR: anaphoricity determination
    1. Expletives:
       1. IT’s not easy to find a solution
       2. Is THERE any reason to be optimistic at all?
    2. Non-anaphoric definites
15. Problems for AR: Complex attachments
    1. [The quality that’s coming out of [software from [India]]
       1. The quality that’s coming out of software from India is now exceeding the quality of software that’s coming out from the United States
    2. scanning through millions of lines of computer code
       1. ACE/bnews/devel/ABC19981001.1830.1257
16. Early systems
    1. Hobbs 1976 Naïve Algorithm
       1. Pronouns only
       2. Syntax based
       3. Still very competitive
    2. Sidner 1979
    3. Carter 1986
17. Modern Work in AR
    1. Availability of the first anaphorically annotated corpora circa 1993 (MUC6) made statistical methods possible.
    2. Most current anaphora resolution systems are based on machine learning, but there is one notable exception, the Stanford Coreference system.
18. MUC
    1. First big initiative in Information Extraction.
    2. Produced first sizeable-annotated data for coreference.
    3. Developed first methods for evaluating systems.
19. MUC terminology:
    1. MENTION: any markable
    2. COREFERENCE CHAIN: a set of mentions referring to an entity
    3. KEY: the (annotated) solution (a partition of the mentions into coreference chains)
    4. RESPONSE: the coreference chains produced by a system
20. The Stanford Deterministic Coreference Resolution System
    1. Part of the Stanford CORE Pipeline
    2. The best-performing system at CONLL 2011, and used as a component by two of the top three systems at CONLL 2012
    3. Key to its performance are
       1. A very high quality mention detection component based on the Stanford CORE pipeline
       2. A PRECISION-FIRST coreference resolution component based on 10 filters called SIEVES that implement many of the restrictions on anaphora resolution discussed in previous slides
21. The Sieves
    1. Speaker Identification: This sieve first identifies speakers, then matches first and second pronouns to these speakers.
    2. Exact Match: This sieve links together two mentions only if they contain exactly the same text, including both determiners and modifiers.
    3. Relaxed String Match: This sieve links together two mentions only if they con- tain exactly the same text after dropping the postmodifiers.
    4. Precise Constructs: This sieve links together two mentions if they occur in one of a series of high precision constructs: e.g., if they are in an appositive con- struction ([the speaker of the House], [Mr. Smith] . . . ), or if both mentions are tagged as NNP and one of them is an acronym of the other.
    5. Strict Head Match: This sieve links together a mention with a candidate antecedent entity if all of a number of constraints are satisfied: (a) the head of the mention matches any of the heads of the candidate antecedent; (b) all nonstop words of the mention are included in the non-stop words of the candidate antecedent; (c) all mention modifiers are included among the modifiers of the candidate antecedent; and (d) the two mentions are not in an i-within-i situation, i.e., one is not a child in the other.
    6. Variants of Strict Head Match: Sieve 6 relaxes the ‘compatible modifiers only’ constraint in the previous sieve, whereas Sieve 7 relaxes the ’word inclusion’ constraint.
    7. Proper Head Match: This sieve links two proper noun mentions if their head words match and a few other constraints apply.
    8. Relaxed Head Match: This sieve relaxes the requirement that the head word of the mention must match a head word of the candidate antecedent entity.
    9. Pronoun resolution: Finally, pronouns are resolved, by finding candidates matching the pronoun in number, gender, person, animacy, and NER label, and at most 3 sentences distant.
22. Statistical approaches to AR:
    1. UNSUPERVISED approaches
       1. Eg Cardie & Wagstaff 1999, Ng 2008
    2. SUPERVISED approaches
       1. Early (NP type specific) • Soon et al: general classifier + modern architecture
23. Soon et al 2001
    1. First ‘modern’ ML approach to anaphora resolution
       1. Resolves ALL anaphors
       2. Fully automatic mention identification
    2. Developed instance generation & decoding methods used in a lot of work since
24. Soon et al: AR as a Classification Problem
    1. Classify MENTION PAIR as coreferential or not
    2. Build a complete coreferential chain
25. Soon et al: Key Decisions
    1. ENCODING
       1. I.e., what positive and negative instances to generate from the annotated corpus?
          1. E.g. treat all elements of the coref chain as positive instances, everything else as negative.
    2. DECODING
       1. How to use the classifier to choose an antecedent?
          1. Some options: ‘sequential’ (stop at the first positive), ‘parallel’ (compare several options)
26. Soon et al: preprocessing
    1. Part-of-Speech (POS) tagger:
       1. HMM-based
       2. 96% accuracy
    2. Noun phrase identification module:
       1. HMM-based
       2. Can identify correctly around 85% of mentions.
    3. Named Entity Recognition (NER): reimplementation of Bikel Schwartz and Weischedel 1999
       1. HMM based
       2. 88.9% accuracy
27. Soon et al: Features
    1. NP type
    2. Distance
    3. Agreement
    4. Semantic class
28. Soon et al: generating training instances
    1. Marked antecedent used to create positive instance
    2. All mentions between anaphor and marked antecedent used to create negative instances
29. Soon et al: decoding
    1. Right to left, consider each antecedent until classifier returns true
30. Evaluation of coreference resolution systems
    1. Lots of different measures proposed
    2. ACCURACY:
       1. Consider a mention correctly resolved if
          1. Correctly classified as anaphoric or not anaphoric
          2. ‘Right’ antecedent picked up
    3. Measures developed for the competitions:
       1. Automatic way of doing the evaluation
    4. More realistic measures (Byron, Mitkov)
       1. Accuracy on ‘hard’ cases (e.g., ambiguous pronouns)
31. Vilain et al 1995: Evaluation
    1. The official MUC scorer
    2. Based on precision and recall of links
32. Vilain et al: Recall
    1. To measure RECALL, look at how each coreference chain Si in the KEY is partitioned in the RESPONSE, and count how many links would be required to recreate the original, then average across all coreference chains.  
         
       R\_T = - (sum of(Si – p(Si)) / (sum of (Si - 1))
33. Vilain et al: Example recall
    1. In the example above, we have one coreference chain of size 4 (S = 4)
    2. The incorrect response partitions it in two sets (p(S) = 2)
    3. R = 4-2 / 4-1 = 2/3
34. Vilain et al: Precision
    1. Count links that would have to be (incorrectly) added to the key to produce the response
    2. I.e., ‘switch around’ key and response in the equation before
35. Beyond Vilain et al
    1. Problems:
       1. Only gain points for links. No points gained for correctly recognizing that a particular mention is not anaphoric
       2. All errors are equal
    2. Proposals:
       1. Bagga & Baldwin’s B-CUBED algorithm
       2. Luo recent proposal
36. Error analysis (Soon et al, 2001)
    1. Errors most affecting precision:
       1. Prenominal modifiers identified as mentions and other errors in mention identification
       2. String match but noun phrases refer to different entities
    2. Errors most affecting recall:
       1. Errors in mention identification (11%)
       2. Errors in SEMCLASS determination (10%)
       3. Need more features (63.3%)
37. Soon et al examples of errors:
    1. Tarnoff, a former Carter administration official and president of the Council on foreign relations, is expected to be named [undersecretary] for political affairs … Former. Sen Tim Wirth is expected to get a newly created [undersecretary] post for global affairs
    2. [Ms Washington and Mr. Dingell] have been considered [allies] of [the Securities exchanges], while [banks] and [future exchanges] often have fought with THEM
38. After Soon et al 2001
    1. Different models of the task
    2. Different preprocessing techniques
    3. Using lexical / commonsense knowledge (particularly semantic role labelling)
    4. Salience
    5. Anaphoricity detection
    6. Development of AR toolkits (GATE, LingPipe, GUITAR)
39. Ranking models
    1. Idea: train a model that imposes a ranking on the candidate antecedents for an NP to be resolved so that it assigns the highest rank to the correct antecedent
    2. A ranker allows all candidate antecedents to be considered simultaneously and captures competition among them:
       1. Allows us find the best candidate antecedent for an NP.
    3. There is a natural resolution strategy for a ranking model:
       1. An NP is resolved to the highest-ranked candidate antecedent
40. How to train a ranking model
    1. Convert the problem of ranking m NPs into the a set of pairwise ranking problems
       1. Each pairwise ranking problem involves determining which of two candidate antecedents is better for an NP to be resolved
          1. Each one is essentially a classification problem
    2. Ranking rediscovered independently by
       1. Yang et al. (2003) (twin-candidate model)
       2. Iida et al. (2003) (tournament model)
    3. Denis & Baldridge (2007, 2008): train the ranker using maximum entropy
       1. model outputs a rank value for each candidate antecedent
41. Entity-mention models
    1. Classifiers that determine whether (or how likely) an NP belongs to a preceding coreference cluster
    2. more expressive than the mention-pair model
       1. can employ cluster-level features defined over any subset of NPs in a preceding cluster
42. Joint Entity Detection and Tracking
    1. Daume and Marcu 2005: Mention identification, classification, and linking take place at the same time
    2. Denis and Balridge 2007: Integer Linear Programming (ILP)
43. Recent state of the art in coreference: the 2012 CONLL Shared Task
    1. Data: OntoNotes
       1. 1.6M words English, 900K words Chinese, 300K words Arabic
       2. Annotated with: syntactic information, wordsenses, propositional information
    2. Tracks:
       1. Closed
       2. Open
    3. Metrics: MELA
       1. (a combination of MUC / B3 / CEAF)
44. Anaphora/Co-reference datasets/annotation tools:
    1. MUC6/MUC7 (small, old)
    2. ACE 2002/2005
    3. ONTONOTES
    4. Phrase Detectives (locally developed at QM and Essex) <http://anawiki.essex.ac.uk/phrasedetectives>
       1. Have a go at Phrase Detectives.
45. Toolkits for AR
    1. BART (Versley et al, 2008).
    2. Stanford Deterministic Coreference Resolver (Lee et al 2013).
    3. CORT (Martschat & Strube 2015).
46. What about the physical world of referents?
    1. How do process reference to real-world objects, say in a photograph?
    2. Recently this has become possible through improvements in computer vision (CV).
    3. One example is the words-as-classifiers model of reference res. (Kennington and Schlangen, 2015)
    4. Trains logistic regression classifiers for each word.
       1. Input: word, CV features of object
       2. Output: probability the word refers to the object
    5. Simple combination of classifiers gets surprisingly good results (65-70% F-score).
47. Verb Phrase Ellipsis
    1. “Branson did so”
    2. Similar problem, but with some important differences
    3. Identifying ellipsis sites is harder:
       1. “I can’t today but I will try tomorrow”
       2. “I can’t today but I will – try tomorrow”
       3. Use lexical, POS sequence, syntactic tree features
    4. Resolving ellipsis is more complex:
       1. Not a purely lexical or syntactic operation:
          1. “John likes tennis. So do you.”
       2. But what’s the semantic operation?   
            
          like(john,tennis) 🡪 like(you,tennis)
       3. Can use lambda calculus to generate possible antecedents
          1. abstract subject:   
               
             like(john,tennis) 🡪 lambda x.like(x,tennis)
          2. Resolve VPE sites to semantic antecedent functions   
               
             P(you) 🡪 P= lambda x.like(x,tennis) 🡪 like(you,tennis)
48. Ellipsis & Ambiguity
    1. Unfortunately, this brings in ambiguity again:
    2. Sometimes that’s appropriate:
    3. “Sue thinks John likes Jane. So does Bill.”  
         
       hink(sue,like(john, jane)) --> lambda x.think(x,like(john, jane))  
         
       think(sue,like(john, jane)) 🡪 lambda x.like(x, jane)
    4. But sometimes some are spurious:
    5. “John likes his teacher and his classmates. Sue does too.”  
         
       like(john,teacher(john) intersect mates(john)) 🡪 lambda x.like(x,teacher(x) intersect mates(x))  
       like(john,teacher(john) intersect mates(john)) 🡪 lambda x.like(x,teacher(John) intersect mates(john))  
       like(john,teacher(john) intersect mates(john)) 🡪 lambda x.like(x,teacher(x) intersect mates(John))  
       like(john,teacher(john) intersect mates(john)) 🡪 lambda x.like(x,teacher(john) intersect mates(x))
    6. Need ambiguity resolution e.g. machine learning using suitable features:
       1. Parallelism (e.g. subject-subject)
       2. Discourse coherence (e.g. from Rhetorical Structure Theory)
       3. Semantic plausibility & event/role restrictions
49. Perceived vs Potential Ambiguity
    1. Perceived ambiguity:
       1. Is it clear who “she” is?
       2. And what “do so” means?
    2. How many potential antecedents are there for:
       1. “she”
       2. “do so”?
50. (Next week) Extreme Ellipsis: Dialogue
    1. British National Corpus KSP 389-393:
    2. Christine 🡪 What have you been up to?
    3. Steve 🡪 Nothing.
    4. Michael 🡪 Eating.
    5. Leslie 🡪 Any phone calls?
    6. Steve 🡪 Nah.
    7. How could we summarise this dialogue?
       1. e.g. C asked what the others had been up to; S said he hadn’t been doing anything, M said he’d been eating. L asked whether there had been any phone calls; S said there hadn’t been any.
    8. (“Summary” is longer than the dialogue …)
51. Discourse Structure
    1. We can assign structure to whole texts
    2. Role relations between clauses/sentences
       1. “Discourse parsing” (e.g. Marcu, 2012)
       2. e.g. Rhetorical Structure Theory (Mann & Thompson, 1993)
    3. (Not core part of this course, though next week Dialogue Acts)