19 Introduction to semantics

Semantic = meaning

Sentential semantics: Who did what to whom ; when, how, why

Lexical semantics: meanings of individual words

sentential syntax (way of sentence):

Constituency

captures linguistic generalisations about grammaticality (substitutability)

generates an unbounded set of grammatical sentences via a finite lexicon and finite rules (recursion)

CNF

can induce probabilistic grammars from a treebank, and so tackle (pervasive) syntactic ambiguity . PCFG

Can get meaning from Sentential syntax

Resolve syntactic ambiguity (partially) resolves semantic ambiguity

Compositionality:

meaning of sentence = meaning of parts + rules

argument grammar, derives logical form (LF) (= following FoL)

By Lexical meanings (word = FoL) + Composition rules (CFG rules + instructions to combine FoL)

Use **First Order Logi**

unambiguous, automatic inference, verifiable

Capture entails relationshi

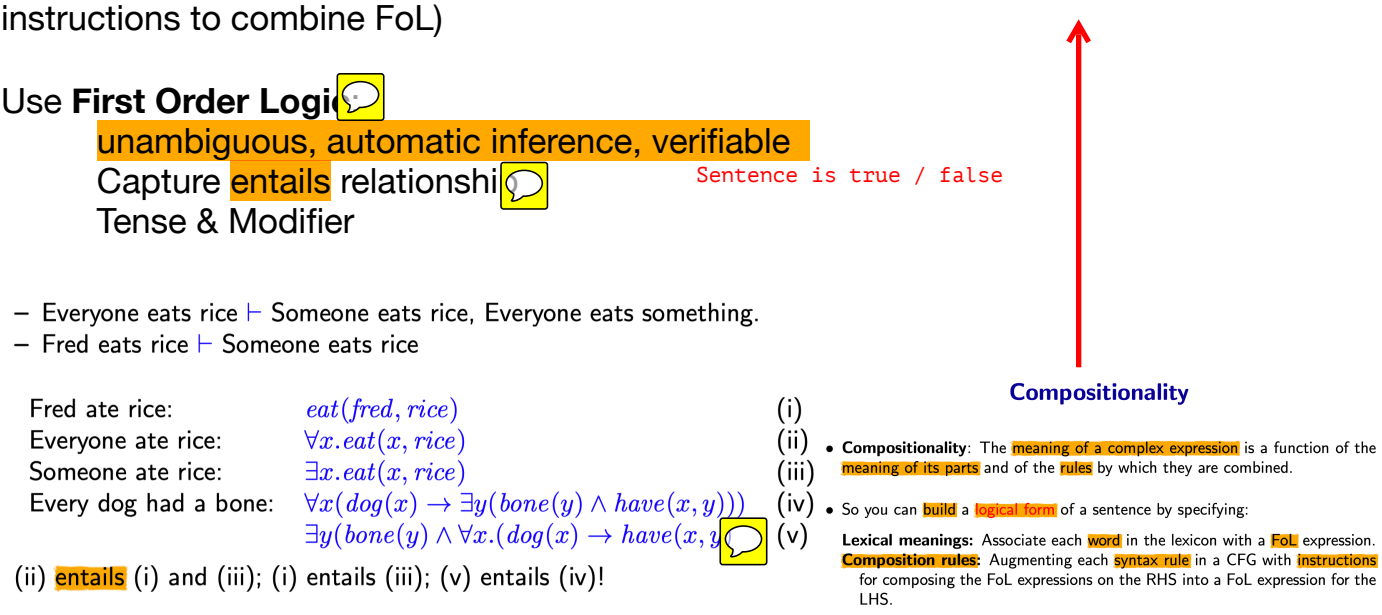
Sentence is true / false

Tense & Modifier

- Everyone eats rice ⊢ Someone eats rice, Everyone eats something.
- Fred eats rice ⊢ Someone eats rice

Fred ate rice:	$eat(fred, rice)$	(i)
Everyone ate rice:	$\forall x.eat(x, rice)$	(ii)
Someone ate rice:	$\exists x.eat(x, rice)$	(iii)
Every dog had a bone:	$\forall x(dog(x) \rightarrow \exists y(bone(y) \wedge have(x, y)))$	(iv)
	$\exists y(bone(y) \wedge \forall x.(dog(x) \rightarrow have(x, y$	(v)

(ii) entails (i) and (iii); (i) entails (iii); (v) entails (iv)!



|— = entails

[For all] entails example & there exist; Example entails there exist; specific entails general

A entails B: If A is true, B is true

Lambda calculus: combine FoL

Replace X, drop lambda, everything else the same

Resolve lambda inside bracket first

Capitalized lambda first

- If φ is a well-formed FoL expression and x is a variable, then $\lambda x \varphi$ is a well-formed FoL expression. It's a function, known as a λ-term.
- λ-terms have interesting semantics, but they also offer a way of substituting (free) variables in an FoL expression with values.

$$\lambda x \varphi(a) = \varphi[x/a]$$

- Creating a function $\lambda x \varphi$ from an expresion φ is called Lambda (λ) abstraction
  Function application is called Beta (β) reduction.

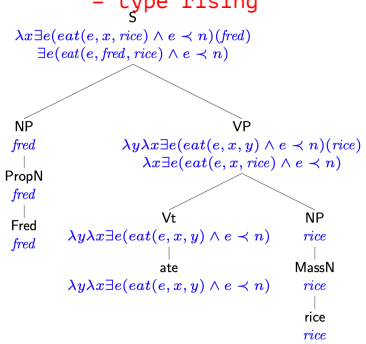
Example:

- $\lambda y \lambda x(\exists e(eat(e, x, y) \wedge e \prec n))(rice)$ becomes
  $\lambda x(\exists e(eat(e, x, rice) \wedge e \prec n))$

Example Composition for Fred ate rice

But we'll see in a minute why it's problematic. . .
. . . and why λ-abstraction on higher types provides a solution!

= type rising



Problematic!

Every man ate rice: $\forall x(man(x) \rightarrow \exists e(eat(e, x, rice) \wedge e \prec n))$

Breaking it down:

- What is the meaning of *Every man* anyway?
  $\forall x(man(x) \rightarrow Q(x))$
- If so, the subject NP needs to be: Previous NP: fred
  $\lambda Q \forall x(man(x) \rightarrow Q(x))$
- But in our grammar we had the VP as the functor:
  S $\rightarrow$ NP VP *VP.Sem(NP.Sem)*
- $\lambda z \exists e(eat(e, z, rice) \wedge e \prec n)(\lambda Q \forall x(man(x) \rightarrow Q(x)))$ becomes
  $\lambda z \exists e(eat(e, \lambda Q \forall x(man(x) \rightarrow Q(x)), rice) \wedge e \prec n)$
- That's not even syntactically well-formed!!

The Grammar that generates that tree

Append FOL

S $\rightarrow$ NP VP *VP.Sem(NP.Sem)*	**(Sentences)**
NP $\rightarrow$ MassN *MassN.Sem* | PropN *PropN.Sem*	**(Noun phrases)**
VP $\rightarrow$ Vi *Vi.Sem* | Vt NP *Vt.Sem(NP.Sem)*	**(Verb phrases)**
PropN $\rightarrow$ Fred *fred* | Jo *jo*. . .	**(Proper nouns)**
MassN $\rightarrow$ rice *rice* | wood *wood* . . .	**(Mass nouns)**
Vi $\rightarrow$ talked $\lambda x \exists e(talk(e, x) \wedge e \prec n)$ | . . .	**(Intransitive verbs)**
Vt $\rightarrow$ ate $\lambda y \lambda x . \exists e(eat(e, x, y) \wedge e \prec n)$ | . . .	**(Transitive verbs)**

Observations:

- λ-term for Vt ensures NP values are in right positions to predicate *eat*
- Rules with two daughters specify in semantics which daughter is the functor and which the argument
  - S rule: VP is the functor.
  - Transitive VP rule: Vt is the functor.

  Functor: argument contains lambda expression
  Argument contains the lambda expression
  Put it in front
- Unary rules 'pass up' the semantics from the daughter.

Solution

Make NP the functor and VP the argument.

S $\rightarrow$ NP VP *NP.Sem(VP.Sem)*

$\lambda Q \forall x(man(x) \rightarrow Q(x))(\lambda z \exists e(eat(e, z, rice) \wedge e \prec n))$
$\forall x(man(x) \rightarrow \lambda z \exists e(eat(e, z, rice) \wedge e \prec n))(x))$
$\forall x(man(x) \rightarrow \exists e(eat(e, x, rice) \wedge e \prec n))$

But this means NPs must all look like this: $\lambda P.P(x)$.
Fred $\mapsto \lambda P.P(fred)$ etc.

Build:
Lambda P.P(sth)
Goal: replace smallest element with sth
Lambda P.Sth ^ P(x)
Goal: AND
Lambda x.do(x)
Goal: Combine with First, replace x with sth
And type raising

Reduction rule:
Reduce things inside bracket first

Else capitalized typed raised first

Grammar Refined! (Changes in purple)

S →	NP VP *NP.Sem(VP.Sem)*	(Sentences)
NP →	MassN *MassN.Sem* | PropN *PropN.Sem* | Det N *Det.Sem(N.Sem)*	(Noun phrases)
VP →	Vi *Vi.Sem* | Vt NP *VT.Sem(NP.Sem)*	(Verb phrases)
PropN →	Fred $\lambda P.P(fred)$ | . . .	(Proper nouns)
MassN →	rice $\lambda P.P(rice)$ | . . .	(Mass nouns)
Vi →	talked $\lambda x \exists e(talk(e,x) \wedge e \prec n)$ | . . .	(Intransitive verbs)
Vt →	ate $\lambda R.\lambda z.R(\lambda y.\exists e(eat(e,z,y) \wedge e \prec n))$	(Transitive verbs)
N →	man $\lambda x.man(x)$	(Count Nouns)
Det →	a $\lambda P \lambda Q \exists x(P(x) \wedge Q(x))$ | every $\lambda P \lambda Q \forall x(P(x) \rightarrow Q(x))$	(Determiners)

Type raising: add "lambda R"

Type Raising to the rescue again

Vt: Transitive verbs (ate)

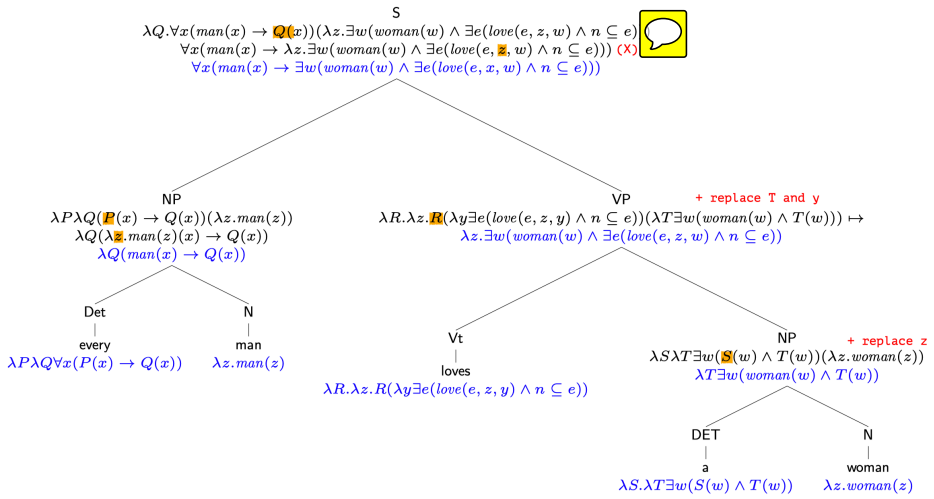
VP → Vt NP *Vt.Sem(NP.Sem)*

Vt → ate $\lambda R.\lambda z.R(\lambda y.\exists e(eat(e,z,y) \wedge e \prec n))$

ate every grape:

$\lambda R.\lambda z.R(\lambda y.\exists e(eat(e,z,y) \wedge e \prec n))(\lambda Q \forall x(grape(x) \rightarrow Q(x)))$ becomes
$\lambda z \lambda Q \forall x(grape(x) \rightarrow Q(x))(\lambda y.\exists e(eat(e,z,y) \wedge e \prec n))$ becomes
$\lambda z \forall x(grape(x) \rightarrow \lambda y.\exists e(eat(e,z,y) \wedge e \prec n)(x))$ becomes
$\lambda z \forall x(grape(x) \rightarrow \exists e(eat(e,z,x)))$

example: every man loves a woman



S → NP VP *VP.Sem(NP.Sem)*
NP → MassN *MassN.Sem* | PropN *PropN.Sem*
VP → Vi *Vi.Sem* | Vt NP *Vt.Sem(NP.Sem)*
PropN → Fred *fred* | Jo *jo*. . .
MassN → rice *rice* | wood *wood* . . .
Vi → talked $\lambda x \exists e(talk(e,x) \wedge e \prec n)$ | . . .
Vt → ate $\lambda y \lambda x.\exists e(eat(e,x,y) \wedge e \prec n)$ | . . .

Scope: causing ambiguity
e.g. every man loves a woman

Interpretation 1: every **has scope over** a Different woman per man
Interpretation 2: a **has scope over** every Same woman

When encountering "some" "a", Think of one specific VS not specific

Details bellow

number of interpretations grows ponentially with the number of scopeoperators (every, some, not, a)

To ensure that the ambiguity is reflected in the LF
surrounding context might provide info

Solution: **Semantic Underspecification**

Semantic Underspecification: Build LFs via syntax that **underspecify** the relative semantic scopes of the quantifiers

- Partial description of a FoL formula By tree
- So Syntax-Tree:LF is 1:1, but the LF describes several FoL formulae and hence several interpretations By assigning h value

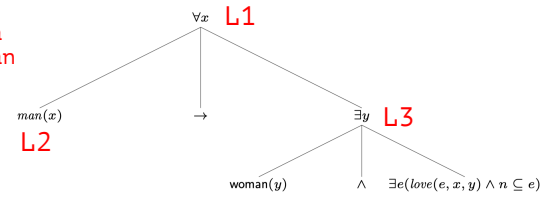
Semantic Underspecification

- The LF constructed in the grammar features:
  1. FoL bits
  2. constraints on how they can combine into an FoL formula

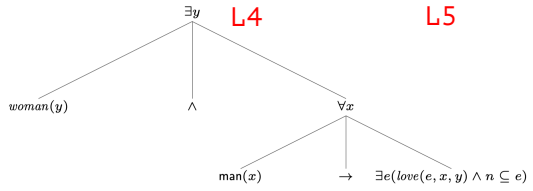
  Thus ambiguity
- The constraints are satisfied by more than one FoL formula.

A Picture showing common bits and different bits

Every man loves a
Not specific woman



Every man loves a
Specific woman



Technique to resolve scope:

- Label nodes of the tree: $l_1, l_2 \ldots$
- Supply constraints on what FoL expressions appear at those labels

Every man loves a woman.

Ignoring $\exists e$ and $n \subseteq e$. . .

$$
\begin{array}{ll}
l_1: & \forall x(h_2 \rightarrow h_3) \\
l_2: & man(x) \\
l_3: & love(e, x, y) \\
l_4: & \exists y(h_4 \wedge h_5) \\
l_5: & woman(y) \\
h_2 =_q l_2, h_4 =_q l_5 &
\end{array}
$$

- All hs must equal a (unique) l; no free variables
- So there are two solutions:

  ∃ outscopes ∀: $h_2 = l_2, h_4 = l_5, h_3 = l_3 h_5 = l_1$ Specify tree
  ∀ outscopes ∃: $h_2 = l_2, h_4 = l_5, h_3 = l_1, h_5 = l_3$
- LF construction via the grammar must now λ-abstract labels, as well as predicates, arguments to predicates etc.

Rules Forming FOL:

Exist e: verb

“A”: exist x

Relation: e.g. ^instrument (e,x); ^time(e,x); ^with(e,style)

If verb is unique: put verb(e) in front of for all, er(e,x) ^ ee(e,y) inside

Alternative grammar: exist e, Use ^ to connect verb(e), subject(e,x), object(e,y)

Intran: no need (sam sees)
transitive verb: need subject

8.	Vi→walks	$\lambda x.\exists e.walking(e) \wedge walker(e,x)$
9.	Vi→sees	$\lambda x.\exists e.seeing(e) \wedge seer(e,x)$
10.	Vt→walks	$\lambda P \lambda x.P(\lambda y.\exists e.walking(e) \wedge walker(e,x) \wedge walkee(e,y))$
11.	Vt→sees	$\lambda P \lambda x.P(\lambda y.\exists e.seeing(e) \wedge seer(e,x) \wedge seen(e,y))$

c) $\exists e.x.eating(e) \wedge sandwich(x) \wedge eater(e, Liang) \wedge eaten(e,x)$ A sandwich

If didn't specify, e.g. John, bone, just replace x with it
If "A BONE", use: exist x. bone(x);
If "THE BONE", : exist! X. bone(x)

When facing "some", "a": (scope ambiguity)

Rules to give FOL:

e.g. "some student ate every apple with a fork"

Use ^ connects all (since there are Det)

Put "One specific" out

Put "not specific" in

(b) $\exists x(Student(x) \wedge \forall y(Apple(y) \rightarrow \exists e \exists z(Fork(z) \wedge Eats(e,x,y) \wedge e \prec n \wedge With(e,z)))$

(c) $\forall x(Apple(x) \rightarrow \exists e \exists y \exists z(Student(y) \wedge Fork(z) \wedge Eats(e,y,x) \wedge e \prec n \wedge With(e,z))$

Specific verb:

All the students (separately) lift Marie ("lift" not specific)

$$\forall x.student(x) \Rightarrow \exists e.lifting(e) \wedge lifter(e,x) \wedge liftee(e, Marie)$$

The students all lift Marie together ("lift" specific)

$$\exists e.lifting(e) \wedge \forall x.student(x) \Rightarrow lifter(e,x) \wedge liftee(e, Marie)$$

Lecture 16 - Semantic Role Labelling and Argument Structure

Semantic (Thematic) Roles

- Instead of focusing on syntax, consider **semantic/thematic roles** defined by each event

Also a feature

Semantic Roles:

OPENER – an initiator/doer in the event [Who?]

OPENED - an affected entity [to Whom / to What?]

Different sentence (syntax) might have same semantic

- The idea of semantic roles can be combined with other aspects of meaning (beyond this course)

- Commonly used thematic roles
  - **Agent** - *The boy* kicked his toy
  - **Theme** - The boy kicked *his toy* (The thing)
  - **Experiencer** - *The boy* felt sad
  - **Result** - The girl built *a shelf* with power tools
  - **Instrument** - The girl built a shelf *with power tools*
  - **Source** - She came *from home*
  - Etc.

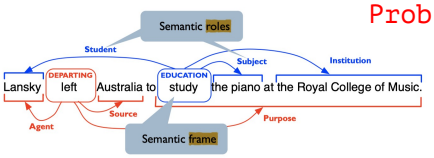
Beneficiary

Patient

Destination

- **Issues with thematic roles**
  - No universally agreed-upon set of roles
  - Items with the same role (e.g. instrument) may not behave quite the same
    - *Sandy opened the door with a key* vs *The key opened the door*
    - *Sandy ate the salad with a fork* vs *The fork ate the salad*
  - The two main NLP resources for thematic roles avoid these problems by defining very fine-grained roles
    - Specific for individual verbs only (PropBank)
    - Specific to small groups of predicates – not only verbs (FrameNe
- **Semantic role labelling** is identifying which words/phrases play which roles in an event

ProbBank: from verb...

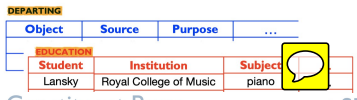


PropBank Frame for *break*:

Frameset **break.01** "break, cause to not be whole":
Arg0: breaker (Proto (first) agent)
Arg1: thing broken (Proto patient)
Arg2: instrument
Arg3: pieces (2-5 variable)

Intuitively, a frame-semantic parser extracts knowledge from text into a relational database

Frames are tables, roles are attributes

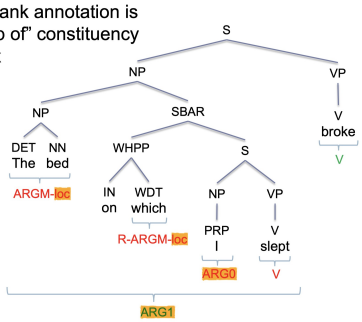


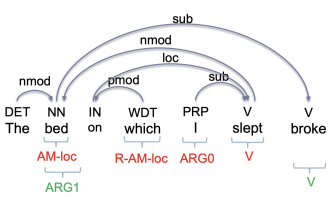
+ functional tags (to modifier / adj of frame verb): TMP, LOC, DIR

SRL on a Constituent Parse

SRL on a Dependency Parse

PropBank annotation is "on top of" constituency syntax



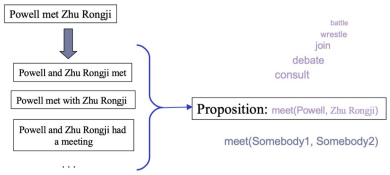


Labeling a node with a role corresponds to labeling the entire dependency subtree rooted at this node with the role

Label bed = label entire subtree on bed

PropBank issues / properties

- Recall our goals:



- Incomplete role consistency across predicates
  - even across synonyms
- Only verbs
  - situations / events are often refereed to with a noun
- Arguably, overly tied to syntax
  - E.g., consider light verb constructions – next slide

Overly tied to syntax:

e.g. light verb construction

Light verb constructions

[New England Electric]$_{Arg0}$ made [an offer of $2 billion]$_{Arg1}$
[to acquire PS of New Hampshire]$_{Arg???}$

Light verb constructions, in current PropBank, are annotated as if make here is a 'normal' verb

Make-01 (*create*):
Arg0: *creator*
Arg1: *creation*
Arg2: *created-from, thing changed*
Arg3: *benefactive*

Offer-01 (*transaction, proposal*):
Arg0: *entity offering*
Arg1: *commodity*
Arg2: *price*
Arg3: *benefactive, or entity offered to*

Different from roles assignment for its paraphras

[New England Electric]$_{Arg0}$ offered [$2 billion]$_{Arg2}$ [for New Hampshire]$_{Arg1}$

FrameNet: Representing events and their participants

FrameNet

- A semantic frame [Fillmore 1968] is a conceptual structure describing a situation, object, or event along with associated properties and participants

Semantic frame = description structure
= a structure describing a thing

- Example: CLOSURE frame
  - Jack opened the lock with a paper clip
  - Semantic Roles (aka Frame Elements
    - OPENER – an initiator/doer in the event [Who?]
    - OPENED - an affected entity [to Whom / to What?]
    - INSTRUMENT – the entity manipulated to accomplish the goal
  - Other roles for CLOSURE/OPENING frame: BENEFICIARY, FASTENER, DEGREE, CIRCUMSTANCES, MANIPULATOR, PORTAL, …

Example: Create_physical_artwo

Definition:

A Creator creates an artefact that is typically an iconic Representation of an actual or imagined entity or event. The Representation may also be evocative of an idea while not based on resemblance.

- Diagrams must be clearly drawn on construction paper.
  I took his picture and told him it came out well.

Frame Elements: (Semantic roles)

Core: ator, representation

Non-Core manner, location_of_representation . . .

FrameNet Issues

Arguably, too small to be useful in applications

- FrameNet r1.7 included 5,093 fully annotated sentences (cf 40,000 for PropBank)

over 1,200 frames, and roles are framespecific

Varying granularity of frames
= gap, sparse

- not designed with natural language understanding applications in mind

Not being tied to syntax, making it hard to predict for ML tools

- arguments do not have to be syntactic constituents (at least in PropBank sense)

FrameNet database

FrameNet is primarily not an annotated corpus, a lexigraphic database which includes an annotated corpus

- Frame definitions
- Relations between frame
- Example realizatio cluding corner cases

Alignment between resources - SemLin

Buy	Sell
Arg0: buyer	Arg0: seller
Arg1: goods	Arg1: goods
Arg2: seller	Arg2: buyer
Arg3: rate	Arg3: rate
Arg4: payment	Arg4: payment

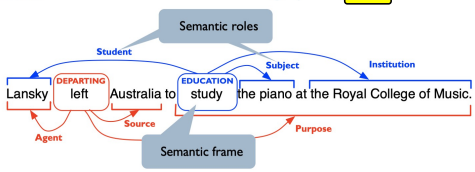
Across PropBank frames and with FrameNet

Example SRL by FrameNet: (similar to prop bank, core & non core)

7. Suggest the semantic role labels you would expect to see in a corpus if the following sentence were annotated using FrameNet: *John carried Mary's bag to the bus for her for a £3 tip yesterday.*

CARRY frame: CARRIER = John, CARRIED = Mary's bag, DESTINATION = to the bus, REWARD = £3, TIME = yesterday
OWNERSHIP frame: OWNER = Mary, OBJECT = bag

Supervised semantic role labeling (SRL)



SRL in supervised setting often in 3 stages:

1. Identify predicates and their senses/frames (multiclass classification)
2. Identify argument spans for the predicate (sequence labeling)
3. Classify the spans into roles (multiclass classification)

But end-to-end approaches now exist as well, and can be as effective

Inductive biases for SRL:

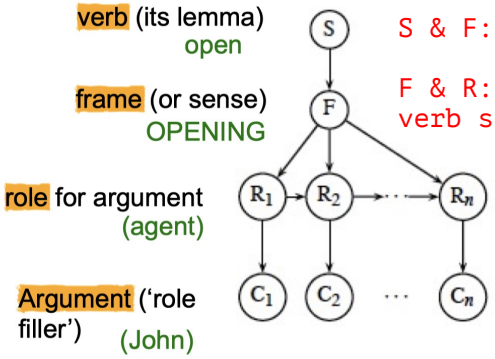
1. Features relying on syntactic structure can be very useful, as syntax is predictive of the semantic representations
2. Constraints on sequence labeling, e.g.,
   - each role appears at most once in a sentence
   - roles should be consistent with predicate (i.e. with the frame definition)
   - constraints from syntax (e.g., PropBank arguments are syntactic constituents)

No annotation: transfer / unsupervised learning:

HMM-like unsupervised SRL models

[Thompson et al, 2003]

$[\text{Jack}]_{\text{Agent}}$ opened $[\text{the lock}]_{\text{Patient}}$ $[\text{with a paper clip}]_{\text{Instrument}}$



S & F: prob of a frame for a verb

F & R: choice of the role depends on the frame/ verb sense

R: transition prob (semantic role)

C: emission prob (seen text)

Lecture 14 - Lexical Semantics: Word Senses, Relations and Classes

Homonymy: two words, same spelling, no related meaning (e.g. bank)
Polysemy: one word, different meaning (but similar)

ontology: A is-a B (includes following:)
hyponym: subset (pigeon is hyponym of bird)
hypernym: superset

meronym: part-whole

sense: different words, different meaning
synonym: same meaning
antonym: opposite meaning

similarity & gradation (gradual change)

WordNet contains:

- **Synsets** ("synonym sets", effectively senses) are the basic unit of organization in WordNet.
  - Each synset is specific to nouns (`.n`), verbs (`.v`), adjectives (`.a`, `.s`), or adverbs (`.r`).
  - Synonymous words belong to the same synset: car^1 (`car.n.01`) = {car,auto,automobile}.
  - Polysemous words belong to multiple synsets: car^1 vs. car^4 = {car,elevator car}. Numbered roughly in descending order of frequency.
- Synsets are organized into a **network** by several kinds of relations, including:
  - **Hypernymy** (Is-A): hyponym {ambulance} is a kind of hypernym car^1
  - **Meronymy** (Part-Whole): meronym {air bag} is a part of holonym car^1

Question Answering

- We would like to build
  - A machine that answers questions in natural language
  - May have access to knowledge bases
  - May have access to vast quantities of English text
- Basically, a smarter Google
- This is typically called **Question Answering**

Semantics

- To build our QA system, we will need to deal with issues in **semantics**, i.e. *meaning*
- **Lexical semantics:** the meanings of individual words
- **Sentential semantics:** how word meanings combine (after that in a sentence)
  - Who did what to whom, when, how, why
- Some examples to highlight problems in lexical semantics:
  - *Plant* (flora) vs *plant* (infrastructure)
    - Words may have different meanings (**senses**)
    - We need to be able to disambiguate between them
  - *Vacation* and *holiday*
    - Words may have the same meaning (**synonyms**)
    - We need to be able to match them
  - *Animals* and *polar bears*
    - Words can refer to a subset (**hyponym**) or superset (**hypernym**) of the concept referred to by another word
    - We need to have a database of such **A is-a B** relationships, called **ontology**
  - *Remove* vs *eliminate*
    - Words may be related in other ways, including **similarity** and **gradation**
    - We need to be able to recognise these to give appropriate responses
  - *Poland* vs *Central Europe*
    - We need to do **inference**
    - A problem for sentential, not lexical, semantics
- Some of these problems can be solved with a good ontology, e.g. WordNet
  - WordNet is a hand-built resource containing 117k **synsets**: sets of synonymous words
  - Synsets are connected by relations such as:
    - hyponym/hypernym (IS-A: chair-furniture)

antonym

homonyms: two words looks the same, diff meaning

Word sense Ambiguity falls into predictable patterns (regular polysemy) :

Pattern	Participating Senses	Example Sentences
Place for an event	Vietnam, Korea, Waterloo, Iraq	It is raining in *Vietnam* / John was shot during *Vietnam*
Place for an institution	White House, Washington, Hollywood, Pentagon, Wall Street*, Supreme Court	The *White House* is being repainted / The *White House* made an announcement

  - Meronym (PART-WHOLE: leg-chair)
  - Antonym (OPPOSITES: good-bad)
- Words are typically semantically ambiguous
  - But there's a lot of regularity (and hence predictability) in the range of *senses* a word can take
  - The senses also influence the word's syntactic behaviour
  - Word senses can be **productive**, making a dictionary model (like WordNet) inadequate
    = a lot
- **Lumping vs Splitting**
  - **Lump** usages of a word into small number of senses
  - **Split** senses to reflect fine-grained distinction

More fine grained (not covered in wordNet):
multiword expressions (including noncom positional expressions, idioms)
Neologisms
Names

- Different translation = different sense
  - Eng. Interest -> German
    - Zins: financial charge paid for load
    - Anteil: stake in company
    - Interesse: all other sense
- **Polysemous** is a word having multiple senses

WordNet chain of hypernyms

(c) relationship, human relationship → relation → abstraction, **abstract entity** → entity
(d) universe, existence, creation, world, cosmos, macrocosm → natural object → whole, unit → object, physical object → **physical entity** → entity

Word Sense Disambiguation (WSD)

- For many applications, we would like to disambiguate senses
  - We may be only interested in one sense
  - Searching for *chemical plant* on the web, we do not want to know about chemical bananas
- Task: given a sense ambiguous word, find the sense in a given *context*
- **WSD as classification**
  - Given a word in context, which sense (class) does it belong to?
  - We can train a supervised classifier, assuming sense-labelled training data
  - Lots of options available:
    - Naive Bayes, Maximum Entropy
    - Decision Lists
    - Decision Trees

e.g. Naive bayes: need prior & posterior prob

+ Use what features?
Neighbor / content words
Syntactically related words / Syntactic role
Topic, PoS Tag

- **Issues with WSD**

Issue 1
  - Not always clear how fine-grained the gold-standard should be
  - Difficult/expensive to annotate corpora with fine-grained senses

Issue 2
  - Classifiers must be trained separately for each word
    - Hard to learn anything for infrequent or unseen words
    - Requires new annotations for each new word

+ Evaluation:
Extrinsic
Intrinsic: acc / F1 + baseline

Not a solution to issue 1: Relying on dictionary senses has limitations in granularity and coverage

Semantic Classes

- Other approaches, such as **named entity recognition** and **supersense tagging** define coarse-grained semantic categories like *person*, *location*, *artifact*
- Like sense, can disambiguate: *Apple* as *organisation* vs *food*
- Unlike senses, which are *refinements* or particular words, classes are typically larger groupings
- Unlike senses, classes can be applied to words/names not listed in a lexicon

Sense: meaning of particular words

OOV

Named Entity Recognition (NER)

- Recognising and classifying **proper names** in text is important for many applications; a kind of **information extraction**
- Different inventories of classes:
  - **Smaller:** *person, organisation, location, miscellaneous*
  - **Larger:** also *product, work_of_art, historical_event, etc.* as well as numeric value types (*time, money*, etc.)

NER Use feature based sequence tagging (example feature: capitalization, gazetteers (list of known names))

Supersense Tagging

- Supersense tagging does beyond NER to cover all nouns and *verbs*

N:ANIMAL	V:COMMUNICATION
N:ARTIFACT	V:COMPETITION
N:ATTRIBUTE	V:CONSUMPTION
N:BODY	V:CONTACT
	V:CREATION
	V:EMOTION

Features (e.g. used in Naive Bayes for WSD)

Simple features ***

- Directly neighboring words (and/or their lemmas)
  - **interest** paid
  - rising **interest**
  - lifelong **interest**
  - **interest** rate
  - **interest** piqued
- Any content words in a 50 word window
  - pastime
  - financial
  - lobbied
  - pursued

More features

- Syntactically related words
- Syntactic role in sense
- Topic of the text
- Part-of-speech tag, surrounding part-of-speech tags

Of course, with NB we have the usual problem with correlated features. MaxEnt doesn't assume they are independent. Independent assumption

+ constituent tags

Dependency tags & words

Extracting frame in frameNet

Also MaxEnt features

Open set words (or lemmas) in window

Word embeddings

Domain knowledge & common sense

+ cross entropy for LM

Bracket score for tree (still F1)

BLeU not used anymore

Upper bond accuracy = IAA

PMI (e.g. enjoy VPing NP)

+ significance test:

Parametric: T-Test;Z-Test
Non-Parametric: McMemar test

Parse tree: Stochastic / pemutation

Evaluation

EVALUATION:

- Extrinsic: test as part of IR, QA, or MT system
- Intrinsic: evaluate classification accuracy or precision/recall against gold-standard senses F1
- Baseline: choose the most frequent sense (sometimes hard to beat)

Lecture 15 - Distributional Semantics

Word Similarity

- How to know if words have similar meanings?
- Can we just use a thesaurus? = Similar words dictionary
  - May not have a thesaurus in every language
  - Even if we do, many words and phases will be missing
- Let's try to compute similarity automatically
- Meaning from context(s)

  A bottle of *raki* is on the table.
  Everybody likes *raki*.
  Raki makes you drunk.
  We make *raki* out of grapes.

Distributional Hypothesis

- Perhaps we can infer meaning just by looking at contexts a word occurs in
- Perhaps meaning *is* the context a word occurs in (Wittgenstein)
- Either way, similar contexts imply similar meanings
  - This idea is known as the **distributional hypothesis**
- Represent each word w_i as a vector of its contexts
  - Distributional semantic models also called **vector-space models**
- Each dimension is a context word
  - = 1 if it co-occurs with w_i
  - = 0 otherwise
- For example:

	pet	bone	fur	run	brown	screen	mouse	fetch
w_1	1	1	1	1	1	0	0	1
w_2	1	0	1	0	1	0	1	0
w_3	0	0	0	1	0	1	1	0

  - Real vectors would be far more sparse!

The Context

- Questions
  - What defines 'context'?
    - What are the dimensions?
    - What counts as co-occurrence?
  - How to weigh the context words (boolean? Counts? other?)
  - How to measure *similarity between vectors*?

Defining the context

- There are **two kinds of co-occurrence** between two words:
  - **First-Order Co-Occurrence (syntagmatic association)** Using PMI
    - Typically nearby each other, *wrote* is a first-order associate of *book*
  - **Second-Order Co-Occurrence (paradigmatic association)** Vector similarity
    - Have similar neighbours *wrote* is second-order associate of *said* and *remarked*

Lemma of Open se

- Usually ignore **stopwords** (function words and other very frequent/uninformative words)
- Usually use a large window around the target word (e.g. 100 words, maybe even whole documen
- But smaller windows allow for relations other than co-occurrence
  - E.g. dependency relation from parser
- All of these for semantic similarity
  - For **syntactic** similarity, use a small window (1-3 words) and track *only* frequent words

Weighing the context words

- Binary indicators are not very informative = one hot
- Presumably more frequent co-occurrences matter more
- Is frequency good enough?
  - Frequent (overall) words are expected to have high counts in the context-vector
    - Regardless of whether they occur more often with this word than with others
- We want to know which words occur *unusually* often in the context of w: more than we'd expect by chance?
  More than independent
  - E.g. 'New' and 'York'
- Put it another way, what **collocations** include w?

Pointwise Mutual Information (PMI)
Solve first order co-occurance

$$PMI(x,y) = log_2 \frac{P(x,y)}{P(x)P(y)} \qquad PMI(x,y) = \log_2 \frac{N \cdot C(x,y)}{C(x)C(y)}$$

- Where
  - $P(x,y)$ is the **actual** probability of seeing words x and y together
  - $P(x)P(y)$ is the **predicted** probability of seeing words x and y together, **IF** x and y are independent!
- PMI tells us how much more/less likely the co-occurrence is than if the words were independent
- **Problems**
  - In practise, PMI is computed with counts (using MLE)
  - Therefore, it is over-sensitive to the chance co-occurrence of infrequent words

Alternatives to PMI for finding collocations

- There are a lot of alternatives!
  - Student *t*-test
  - Pearson's χ^2 statistic
  - Dice coefficient
  - Likelihood ratio test (Dunning, 1993)
  - Lin association measure (Lin, 1998)
  - Etc.
- Of those listed above, Dunning's Likelihood Ratio Test is probably the most reliable one for low counts
  - However, which works best may depend on the downstream too

Improving PMI

- Use **Positive PMI** (PPMI)
  - Change all negative PMI values to 0
  - Because of infrequent words, not enough data to accurately determine negative PMI values
- Introduce *smoothing* in PMI computation

Similarity Solve second order

- Assume you have context vectors for two words $\overline{v}$ and $\overline{w}$
  - Vectors in high-dimensional space
  - Containing PMI (or PPMI) values for all context words
- Vectors seem to capture both syntactic and semantic information
- So the question is, how to measure the 'distance' between two vectors?
  - Euclidean Distance $\left(\sum_i (v_i - w_i)^2\right)^{1/2}$ 💬
    - Doesn't work well even if one dimension has an extreme value
  - Dot Product
    - Vectors are longer if they have higher values in each dimension
    - So more frequent words have higher dot products
      - Dot product is generally larger for longer vectors, regardless of similarity
  - Normalised Dot Product = cosine
    - Normalise through dividing by the length of each vector

$$distance_{NDP} = \frac{\overline{v} \cdot \overline{w}}{|\overline{v}|\,|\overline{w}|}$$

      - The normalised dot product is just the cosine of the angle between vectors
        - Ranges from -1 (vectors pointing to opposite directions) to 1 (same direction)

Alternatives:

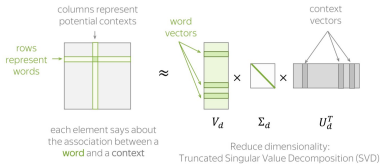
- Again, there are many other similarity measures
  - Jaccard measure
  - Dice measure
  - Jenson-Shannon divergence KL divergence
  - Etc.
  - Again, depends on the downstream too

Evaluation of similarity computations

- Intrinsic evaluation is often a comparison to *psycholinguistic* data
  - Relatedness judgements Being asked:
    - E.g. on a scale of 1-10, how related are the following concepts:
      - Lemon and Truth = 1/10
      - Lemon and Orange = 9/10
    - Still a funny task
    - Answers depend a lot on *how* the question is asked (e.g. *related* vs *similar*)
  - Word association
    - Upon seeing or hearing a word, say the first word that comes to mind
    - Data collected from lots of people provides probabilities of each answer
    - For example, for *Lemon*,
      - 0.16 Orange
      - 0.11 Sour
      - 0.09 Tree
- Benchmarking
  - Human judgements provide a ranked list of related words/associations for each word w
  - Computer system provides a ranked list of most similar words to w
  - Compute the Spearman Rank Correlation between the lists (how well do the rankings match)

Compact space Similar to one hot

- So far our vectors have length V, the size of the vocabulary
- Do we really need this many dimensions?
- Can we represent words in a smaller dimensional space that preserves the similarity relationships of larger space?



Latent Semantic Analysis (LSA) Reduce word vector dimensio

- One of the earliest methods for reducing dimensions while preserving similarity
- Like Principal Component Analysis (PCA) except that we do not subtract off the means
- LSA representations usually work better than originals for many tasks

Neural Network Methods

- Recent methods for learning reduced-dimensional representations (now often called **embeddings**)
- Train a neural network to predict context words based on input word
  - Use hidden layer(s) as the input word's vector representation
- Deep mathematical similarities to LSA, but can be faster to train

Compositionality

- One definition of collocations: **non-compositional** phrases Could not be teared apart = idiomatic expression
  - *White House* is not just a house that is white
- But a lot of language *is* compositiona
  - Red barn
  - Wooden plank
- Can we capture compositionality in a vector space model?
  - More formally, compositionality implies some operator $\oplus$ such that

$$\text{meaning}(w_1 w_2) = \text{meaning}(w_1) \oplus \text{meaning}(w_2) \quad 2)$$

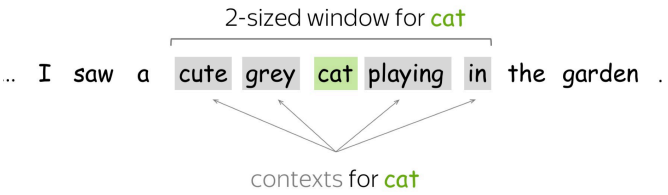
-
  - Possible operators
    - vector addition (doesn't work very well)
    - tensor product
    - nonlinear operations learned by neural networks (current trend) vorks
- One problem: words like 'not'
  - More like operators than points in space

Lecture 22 Neural Embeddings & Neural Classifiers

Since could use embedding dimension, no need to represent word (token) as number

One hot embedding (embedding dim = vocal size); Long & No similarity

Latent semantic Analysis



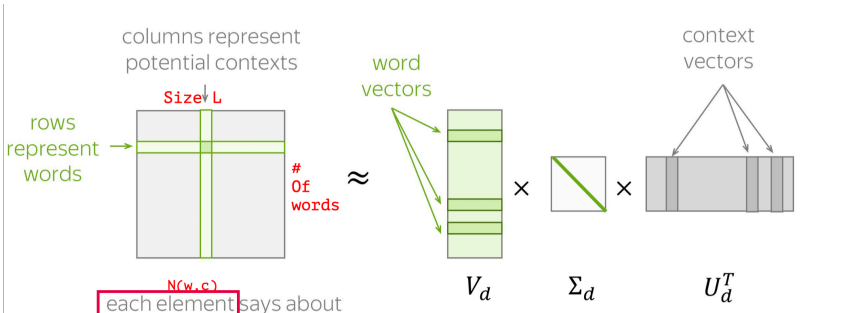
Context:

- surrounding words in a L-sized window

Matrix element:

- N(w, c) – number of times word w appears in context c

Recall: to make it work reasonably well, you need something more sophisticated (e.g., PMI)

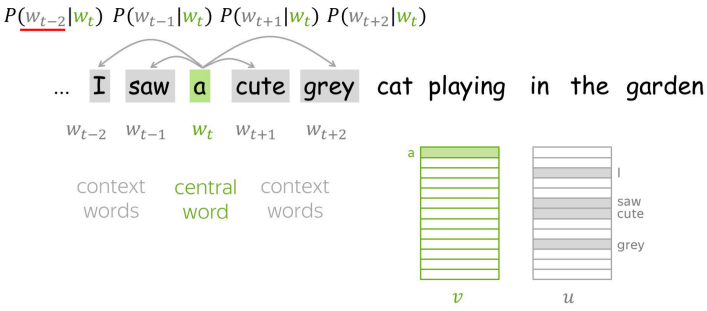


each element says about the association between a word and a context

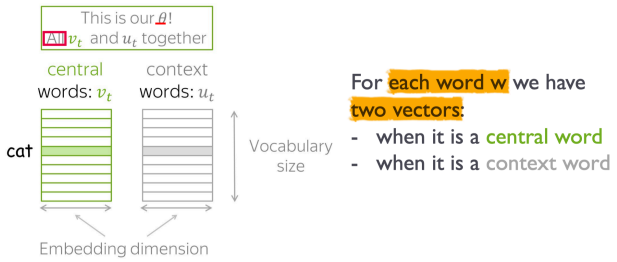
Reduce dimensionality: Truncated Singular Value Decomposition (SVD)

Hard to scale, any alternative?

Prediction-based (aka neural) methods (**Mikolov's Skipgram**):



How do we calculate the probabilities $P(w_{t+j}|w_t,\theta)$?



For each word w we have two vectors:

- when it is a central word
- when it is a context word

The probability of the context word o given the central word c is

The Formula for Expected Value (EV) Is:

$EV = \sum P(X_i) \times X_i$

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

Dot product: measures similarity of o and c. Larger dot product = larger probability

Normalize over entire vocabulary to get probability distribution

This is the softmax function

$$\text{Loss} = J(\theta) = -\frac{1}{T}\log L(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m \le j \le m, \\ j \neq 0}} \log P(w_{t+j}|w_t,\theta)$$

agrees with our plan above ↦ go over text ↦ with a sliding window ↦ compute probability of the context word given the central

Optimized by gradient descent $\theta^{new} = \theta^{old} - \alpha \nabla_\theta J(\theta)$

Stochastic (optimize one word at a time):

$$-\frac{1}{T}\sum_{t=1}^{T}\sum_{-m \le j \le m, j \neq 0} \log P(w_{t+j}|w_t,\theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-m \le j \le m, j \neq 0} J_{t,j}(\theta)$$

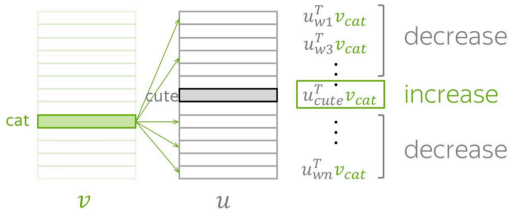
e.g.

I saw a cute grey cat playing in the garden Relation to LSA

$$J_{t,j}(\theta) = -\log P(cute|cat) = -\log \frac{\exp u_{cute}^T v_{cat}}{\sum_{w \in Voc} \exp u_w^T v_{cat}}$$

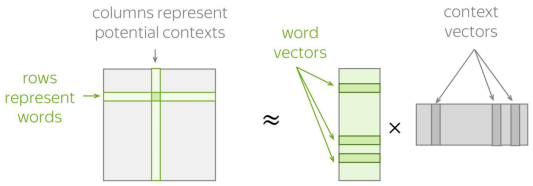
It is possible to show that optimizing the skipgram objective (with the negative sampling modification) corresponds to factorizing PMI matrix (Levy & Goldberg 2014)

A gradient step results in:



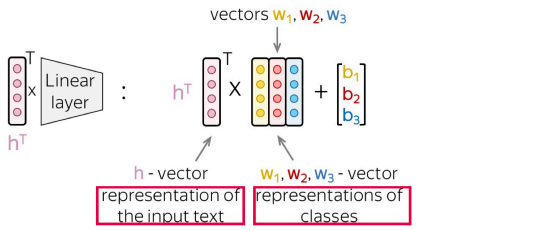
In practice, a slight modification of this objective (negative sampling) is used for faster training

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m, j \neq 0} P(w_{t+j}|w_t, \theta)$$

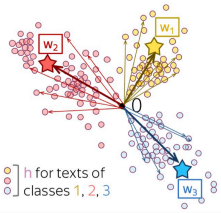


Classification by logistic regression / NN (with softmax)

Representation of the document

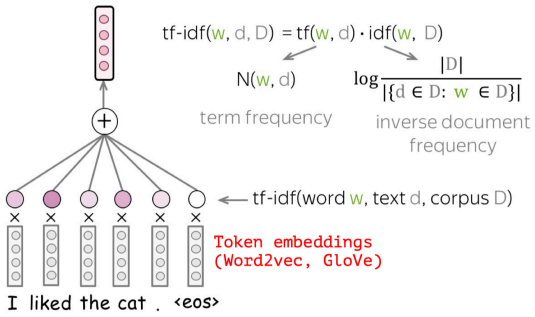


Intuition: the representation of the document points in the direction of the class representation



Basic models: bags of words (= Embeddings)

Weighted sum of embeddings
(e.g., using tf-idf weights)



Lecture 23 Neural Embeddings & Neural Classifiers

RNN, multi layer RNN, bidirectional RNN

RNN language model (predict next token, get word embedding)

Seq2Seq, Encoder Decoder, Conditional LM, Beam search decoding

Problem: this is a bottleneck!

Problem: Fixed source representation is suboptimal:

- for the encoder, it is hard to compress the sentence;
- for the decoder, different information may be relevant at different steps.

Solution: modeling "attention"

Lecture 17 - Discourse Coherenc

continuation in language (sentences)

Pragmatics:

the study of how utterances convey meaning that's linguistically implicit

Discourse Coherence

- Making sense of **verbal actions**
  - We assume action choice isn't arbitrary (choice is informed by the context)
  - So we infer more than we see
  - And may change these inferences as we see more
- **Representation**
  - How should discourse coherence be represented *formally* and *computationally*?
- **Construction**
  - What inference processes, and what knowledge sources, are used when *identifying coherence relations*?
- Examples:
  - "John can open Bill's safe." "He knows the combination"
    - If "He" is John: we infer *explanation* ("because")
    - If "He" is Bill: we infer (at best) *continuation* ("and") with a very vague topic
  - "John can open Bill's safe." "He should change the combination."
    - If "He" is Bill: we infer *result* ("so")
    - If "He" is John, we infer a weaker *result* (?)
    - Subjects are more likely antecedents, but not here!
    - Pronouns shall be interpreted in a way that *maximises* coherence, even if this conflicts with predictions from other knowledge sources!
- **Word Meaning**
  - "A: Did you buy the apartment?" "B: Yes, but we rented it / No, but we rented it"
    - "Yes, but we rented it" signifies that B is the landlord and is renting out the flat
    - "No, but we rented it" signifies that B is renting the flat
- **Bridging**
  - "John took an engine from Avon to Dansville" "He picked up a boxcar / He also took a boxcar"
    - "He picked up a boxcar" signifies that the boxcar was in Dansville
    - "He also took a boxcar" signifies that the boxcar was in Avon
- **Implicit Agreement**
  - M (to K and S): "Karen and I are having a fight"
    M (to K and S): "after she went out with Keith not me"
    K (to M and S): "Well Mark, you never asked me out"
    - "Well" entails implicit agreement

Explicit relationship:
"because", "what's more"
"Then", "and"

- **Dishonesty**
  - P: "Do you have any bank accounts in Swiss banks, Mr. Bronston?" B: "No, sir."
    P: "Have you ever?" B: "The company had an account there for about six months, in Zurich."
    - The last sentence is interpreted as an indirect answer, implying *no* (he did not have a personal bank account in Swiss banks ever)
    - His answer is *literally true*, but the *negative answer* is *false*
      - In fact, Bronston had once had a large personal bank account in Switzerland, where over a five year period he had deposited more than $180,000
    - Supreme Court overrules conviction for perjury (= false proof)
      - Different ruling is probable if Bronston had said "*Only* the company had an account there for about six months, in Zurich."

- Gesture
  - Coherence relations connect speech and gesture and sequences of gestures
    - Speech *so that* gesture
    - Speech *by* gesture
    - Speech *and moreover* gesture

- Discourse coherence drives a lot of pragmatic infere
  - Resolution of pronouns
    Temporal and spatial inference
    Agreement (and disagreement)
    Gesture
    Plausible Deniability

Segmented Discourse Representation Theory

SDRT: The Logical Form (LF) of Monologue

- Logical form consists of A, F, El(π1,π2), connected subsegs, inferred, relationship
  - Set A of **labels** $\pi_1, \pi_2, \ldots$
    - Each label stands for a **segment** of discourse
  - A mapping F from each label to a formula representing its content
  - Vocabulary includes coherence relations
    - E.g. $Elaboration(\pi_1, \pi_2)$
- Logical Forms and Coherence
  - Coherent discourse is a single segment of rhetorically connected subsegments
  - More formally:
    - The **partial order** over A induced by F has a **unique root**
- Example

π_1: John can open Bill's safe.
π_2: He knows the combination.

π_0 : *Explanation*(π_1, π_2)
π_1 : ιx(*safe*(x) & *possess*$(x, bill)$ & *can*$(open(e_1, john, x))$
π_2 : ιy(*combination*(y) & *of*(y, x) & *knows*$(john, y)$

Truth condition:

[$\mathcal{F}(\pi_0)$]iff [$Explanation(\pi_1, \pi_2)$]
iff $\mathcal{F}(\pi_1) \wedge \mathcal{F}(\pi_2) \wedge \varphi_{Expl}(\pi_1, \pi_2)$
iff ιx(*safe*$(x) \wedge$ *possess*$(x, bill) \wedge$ *can*$(open(e_1, john, x)) \wedge$
ιy(*combination*(y) & *of*$(y, x) \wedge$ *knows*$(john, y)) \wedge$
$\wedge$ *cause*(e_{π_2}, e_{π_1})
= expl (pi1, pi2)

- Bits in red are specific values that go beyond content that's revealed by linguistic form.
-
- They are inferred via commonsense reasoning that's used to construct a maximally coherent interpretation.

SDRT: Logical form of dialogue

- LF tracks all current public commitments for each agent, including commitments to coherence relations.

(1) a. M (to K and S): Karen 'n' I're having a fight,
b. M (to K and S): after she went out with Keith and not me.
c. K (to M and S): Well Mark, you never asked me out.

Turn	M	K
1	π_{1M} : *Explanation*(a, b)	$\emptyset$
2	π_{1M} : *Explanation*(a, b)	π_{2K} : *Explanation*$(a, b) \wedge$ *Explanation*(b, c)

Dishonesty

Asher and Lascarides (2011)

(2) a. P: Do you have any bank accounts in Swiss banks?
b. B: No, sir.
c. P: Have you ever?
d. B: The company had an account there for 6 months.

n	Prosecutor	Bronston
	$a : \mathcal{F}(a)$	$\emptyset$
2	$a : \mathcal{F}(a)$	π_{2B} : *Answer*(a, b)
3	π_{3P} : *Continuation*(a, c)	π_{2B} : *Answer*(a, b)
4	π_{3P} : *Continuation*(a, c)	π_{4B} : *Answer*$(a, b) \wedge$ *Continuation*(b, d)

Without ∧ Indirect-Answer(c, d)

1. Plausible Deniability: Must test rigorously whether it's safe to treat the implied answer as a matter of public record.

2. Neologism proof equilibria: distinguishes (2)d vs. "only".

Computational methods for constructing formal semantic representations of discourse

Symbolic approaches to constructing LF

- Draw on rich information sources:
  - linguistic content, world knowledge, mental states...
- Deploy reasoning that supports inference with partial information
  Unlike classical logic, this requires consistency tests.
- Typically, construct LF and evaluate it in the same logic,
  making constructing LF undecidable.
  = No automatic inference

Since hand-crafted rules, it is only feasible for very small domains

So Want to learn a discourse parser, but lack of annotated corpus

State of the art: Supervised Learning for SDRT (77% F score)

Avoiding Annotation

Sporleder and La

- Coherence relations can be overtly signalled:
  - *because* signals EXPLANATION; *but* signals CONTRAST
    = self supervised, get coherence relationship from data
- So produce a training set *automatically*:
  - Max fell because John pushed him
    ⇒
    EXPLANATION(*Max fell*, *John pushed him*).

BUT

- Combined training set of manual and automatically labelled examples doesn't improve accuracy.

So you're better off manually labelling a small set of examples!

Why?

Contrast to *Elaboration* Example: relation could not be auto produced

Although the electronics industry has changed greatly, possibly the greatest change is that very little component level manufacture is done in this country.

Summary

- Computing logical form should be decidable; modularity is key to this.
- Data-driven approaches are a major challenge.
- Linking rich models of discourse semantics to models of human behaviour and decision making is also a major challenge, but essential for tackling dialogues where the agents' goals conflict.