CS 784: Computational Linguistics
Lecture 15: Semantics

Freda Shi

School of Computer Science, University of Waterloo
fhs@uwaterloo.ca

March 11, 2025

[Slides adapted from Weiwei Sun.]

What is Semantics?

Semantics from PIE root ***dheie- "to see, look"**.

PIE = Proto Indo European, a hypothetical, reconstructed ancestor of Indo-European languages.

What is Semantics?

Semantics from PIE root ***dheie- "to see, look"**.

Meaning from PIE ***meino- "opinion, intent"**, perhaps from root ***men- "to think"**.

PIE = Proto Indo European, a hypothetical, reconstructed ancestor of Indo-European languages.

What is Semantics?

Semantics from PIE root ***dheie- "to see, look"**.

Meaning from PIE ***meino- "opinion, intent"**, perhaps from root ***men- "to think"**.

PIE = Proto Indo European, a hypothetical, reconstructed ancestor of Indo-European languages.

- Semantics is a subdiscipline of linguistics, which deals with the meaning of words and sentences.

What is Semantics?

Semantics from PIE root ***dheie- "to see, look"**.

Meaning from PIE ***meino- "opinion, intent"**, perhaps from root ***men- "to think"**.

PIE = Proto Indo European, a hypothetical, reconstructed ancestor of Indo-European languages.

- Semantics is a subdiscipline of linguistics, which deals with the meaning of words and sentences.
- Its object of study is a specialized kind of linguistic meaning:
  - Tied to the language signal;
  - Precisely definable;
  - Reasonably objective (shared by speakers).

What is Semantics?

Semantics from PIE root ***dheie- "to see, look"**.

Meaning from PIE ***meino- "opinion, intent"**, perhaps from root ***men- "to think"**.

PIE = Proto Indo European, a hypothetical, reconstructed ancestor of Indo-European languages.

- Semantics is a subdiscipline of linguistics, which deals with the meaning of words and sentences.
- Its object of study is a specialized kind of linguistic meaning:
  - Tied to the language signal;
  - Precisely definable;
  - Reasonably objective (shared by speakers).
- This is opposed to all meaning in the world:
  - Private, modified by live experience;
  - entirely subjective.

A Quick Example

1. Kim promised Jo to do the dishes.
2. Kim wanted to do the dishes.
3. Jo failed to do the dishes.
4. Kim persuaded Jo to do the dishes.
5. Kim managed to do the dishes.

A Quick Example

1. Kim promised Jo to do the dishes.
2. Kim wanted to do the dishes.
3. Jo failed to do the dishes.
4. Kim persuaded Jo to do the dishes.
5. Kim managed to do the dishes.

Questions

- Does a washing up action take place or not?

A Quick Example

1. Kim promised Jo to do the dishes.
2. Kim wanted to do the dishes.
3. Jo failed to do the dishes.
4. Kim persuaded Jo to do the dishes.
5. Kim managed to do the dishes.

Questions

- Does a washing up action take place or not?
- Who is doing the washing up?

A Quick Example

1. Kim promised Jo to do the dishes.
2. Kim wanted to do the dishes.
3. Jo failed to do the dishes.
4. Kim persuaded Jo to do the dishes.
5. Kim managed to do the dishes.

Questions

- Does a washing up action take place or not?
- Who is doing the washing up?
- Do we learn anything else about the washing up?

Some Meta Language

Precise representations need a **language**.

- Natural language, e.g., English.
- Programming language, e.g. Python, SQL.
- Math, e.g. matrix.
- Logic, e.g. λ calculus.
- Automata, e.g. finite-state machines.

Some Meta Language

Precise representations need a **language**.

- Natural language, e.g., English.
- Programming language, e.g. Python, SQL.
- Math, e.g. matrix.
- Logic, e.g. λ calculus.
- Automata, e.g. finite-state machines.

These languages are not necessarily disjoint.

Representing Meaning with Language

Lexicography, e.g. Cambridge Dictionary (`https://dictionary.cambridge.org`).

Representing Meaning with Language

Lexicography, e.g. Cambridge Dictionary (`https://dictionary.cambridge.org`).

Two dictionary entries of a word

- adjective (colour): of the colour of the sky without clouds on a bright day, or a darker or lighter type of this.
- adjective (sad): feeling or showing sadness.

Representing Meaning with Language

Lexicography, e.g. Cambridge Dictionary (`https://dictionary.cambridge.org`).

Two dictionary entries of a word

- adjective (colour): of the colour of the sky without clouds on a bright day, or a darker or lighter type of this.
- adjective (sad): feeling or showing sadness.

Two senses of the word **blue**.

Representing Meaning with Language

Lexicography, e.g. Cambridge Dictionary (`https://dictionary.cambridge.org`).

Two dictionary entries of a word

- adjective (colour): of the colour of the sky without clouds on a bright day, or a darker or lighter type of this.
- adjective (sad): feeling or showing sadness.

Two senses of the word **blue**.

To enable precise reasoning, we usually need more formal languages.

Representing Meaning with Language

Lexicography, e.g. Cambridge Dictionary (`https://dictionary.cambridge.org`).

Two dictionary entries of a word

- adjective (colour): of the colour of the sky without clouds on a bright day, or a darker or lighter type of this.
- adjective (sad): feeling or showing sadness.

Two senses of the word **blue**.

To enable precise reasoning, we usually need more formal languages.

This lecture will mostly cover sentence-level semantics.

Event in an Expression

The explosive **eruption** of *Hunga-Tonga Hunga-Ha'apai* **sent** *a shockwave* *around the world*.

The **event** literally touched every corner of the globe as the pressure wave spread out in all directions to complete a full circumnavigation.

Event in an Expression

The explosive **eruption** of *Hunga-Tonga Hunga-Ha'apai* **sent** *a shockwave* *around the world*.

The **event** literally touched every corner of the globe as the pressure wave spread out in all directions to complete a full circumnavigation.

[Source: BBC News]

Computational Lexicography

Cambridge Dictionary *send* (verb)

- To cause something to go from one place to another, especially by post or email.
- To cause or order someone to go and do something ...

Computational Lexicography

Cambridge Dictionary *send* (verb)

- To cause something to go from one place to another, especially by post or email.
- To cause or order someone to go and do something ...

FrameNet *send*.v Sending

A Sender plans the Path (along with Source and Goal) of a Theme and places it in circumstances such that it travels along this Path under the power of some entity other than the Sender.

The explosive eruption of Hunga-Tonga Hunga-Ha'apai SENT a shockwave around the world.

Frame Semantics

- **Assumption**: To understand the meanings of the words in a language we must first have knowledge of the **semantic frames**.

Frame Semantics

- **Assumption**: To understand the meanings of the words in a language we must first have knowledge of the **semantic frames**.
- A **semantic frame** is a schematic representation of an **event**, object, situation, or relation providing the background structure against which words are understood.

[Source: `https://framenet2.icsi.berkeley.edu/docs/allslides2.pdf`]

Frame Semantics

- **Assumption**: To understand the meanings of the words in a language we must first have knowledge of the **semantic frames**.
- A **semantic frame** is a schematic representation of an **event**, object, situation, or relation providing the background structure against which words are understood.

[Source: https://framenet2.icsi.berkeley.edu/docs/allslides2.pdf]

Charles J. Fillmore (1929-2014)

List verbs for eating and drinking that you know in English

breakfast.v, *consume.v*, *devour.v*, *dine.v*, *down.v*, *drink.v*, *eat.v*,

List verbs for eating and drinking that you know in English

breakfast.v, *consume.v*, *devour.v*, *dine.v*, *down.v*, *drink.v*, *eat.v*, *feast.v*, *feed.v*, *gobble.v*, *gulp.n*, *gulp.v*, *guzzle.v*, *have.v*, *imbibe.v*, *ingest.v*, *ingestion.n*, *lap.v*, *lunch.v*, *munch.v*, *nibble.v*, *nosh.v*, *nurse.v*, *put away.v*, *put back.v*, *quaff.v*, *sip.n*, *sip.v*, *slurp.n*, *slurp.v*, *snack.v*, *sup.v*, *swig.n*, *swig.v*, *swill.v*, *tuck.v*

List verbs for eating and drinking that you know in English

breakfast.v, *consume.v*, *devour.v*, *dine.v*, *down.v*, *drink.v*, *eat.v*, *feast.v*, *feed.v*, *gobble.v*, *gulp.n*, *gulp.v*, *guzzle.v*, *have.v*, *imbibe.v*, *ingest.v*, *ingestion.n*, *lap.v*, *lunch.v*, *munch.v*, *nibble.v*, *nosh.v*, *nurse.v*, *put away.v*, *put back.v*, *quaff.v*, *sip.n*, *sip.v*, *slurp.n*, *slurp.v*, *snack.v*, *sup.v*, *swig.n*, *swig.v*, *swill.v*, *tuck.v*

The nouns here are nominalizations of the verbs.

List verbs for eating and drinking that you know in English

breakfast.v, consume.v, devour.v, dine.v, down.v, drink.v, eat.v, feast.v, feed.v, gobble.v, gulp.n, gulp.v, guzzle.v, have.v, imbibe.v, ingest.v, ingestion.n, lap.v, lunch.v, munch.v, nibble.v, nosh.v, nurse.v, put away.v, put back.v, quaff.v, sip.n, sip.v, slurp.n, slurp.v, snack.v, sup.v, swig.n, swig.v, swill.v, tuck.v

The nouns here are nominalizations of the verbs.

Frame: Ingestion

An Ingestor consumes food or drink (Ingestibles), which entails putting the Ingestibles in the mouth for delivery to the digestive system. This may include the use of an Instrument. Sentences that describe the provision of food to others are NOT included in this frame.

The wolves DEVOURED the carcass completely.

FrameNet

A computational Lexicography project based on the principles of Frame Semantics.

- 1,224 frames
- 13,640 lexical units
- 10,542 frame elements
- 1,876 frame-to-frame relations
- **202,229 annotated sentences**
- **14% "full-text" annotation**

[Source: `https://framenet2.icsi.berkeley.edu/docs/allslides2.pdf`]

A Formal Attempt to Evaluate Semantic Equivalence

Provides a “shallow” semantic analysis (no modality and scope).

A Formal Attempt to Evaluate Semantic Equivalence

Provides a "shallow" semantic analysis (no modality and scope).

Generalizes well across **some** languages.

A Formal Attempt to Evaluate Semantic Equivalence

Provides a "shallow" semantic analysis (no modality and scope).

Generalizes well across **some** languages.

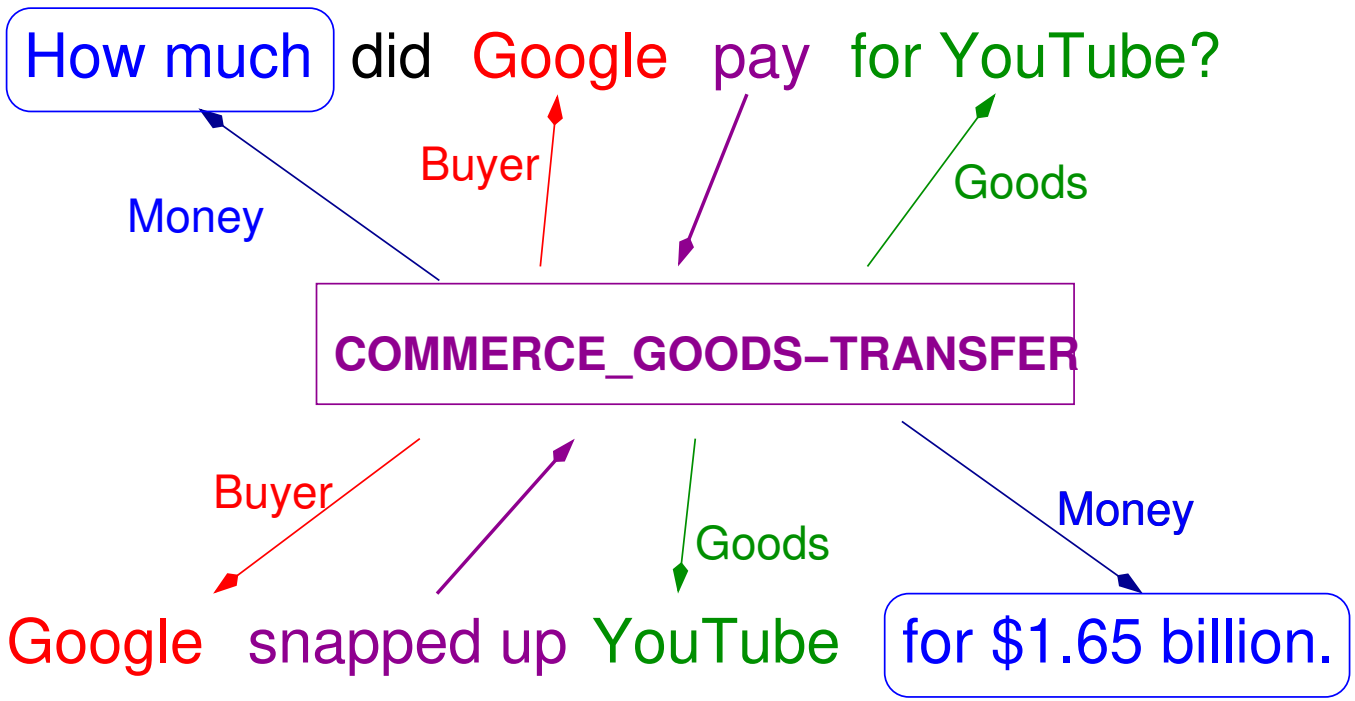
Benefit(ted) various NLP tasks, and interpretability!

A Formal Attempt to Evaluate Semantic Equivalence

Provides a "shallow" semantic analysis (no modality and scope).

Generalizes well across **some** languages.

Benefit(ted) various NLP tasks, and interpretability!



Subcategorization

Fact 1: a fixed configuration of **required** participants in the actions they denote.

Subcategorization

Fact 1: a fixed configuration of **required** participants in the actions they denote.

- *I baked a cake*.
- *I bet you five dollars I can get 100 marks in assignment 2*.
- *I was sleeping*.

Subcategorization

Fact 1: a fixed configuration of **required** participants in the actions they denote.

- *I baked a cake*.
- *I bet you five dollars I can get 100 marks in assignment 2*.
- *I was sleeping*.

- John ate the steak.
- John devoured the steak.
- *John devoured.

Subcategorization

Fact 1: a fixed configuration of **required** participants in the actions they denote.

- *I baked a cake*.
- *I bet you five dollars I can get 100 marks in assignment 2*.
- *I was sleeping*.

- John ate the steak.
- John devoured the steak.
- *John devoured.

- I dined.
- *I dined pizza.

Argument vs. Adjuncts

There are also some **optional** participants, which can sometimes look surprisingly similar to arguments:

- I *waited* *for hours*.
- I *waited* *for the bus*.
- I *waited* *for the bus* *for hours*.

Argument vs. Adjuncts

There are also some **optional** participants, which can sometimes look surprisingly similar to arguments:

- I *waited* *for hours*.
- I *waited* *for the bus*.
- I *waited* *for the bus* *for hours*.

Repetition of Adjuncts and Arguments

- I was waiting for the bus in July in the afternoon for a long time between church and tea at a bus station in the cold wind.
- *I gave the cake to Kim to Sandy.

Argument vs. Adjuncts

There are also some **optional** participants, which can sometimes look surprisingly similar to arguments:

- I *waited* *for hours*.
- I *waited* *for the bus*.
- I *waited* *for the bus* *for hours*.

Repetition of Adjuncts and Arguments

- I was waiting for the bus in July in the afternoon for a long time between church and tea at a bus station in the cold wind.
- *I gave the cake to Kim to Sandy.

Arguments of the same kind can be applied exactly once in a clause, whereas adjuncts of the same kind can be repeatedly applied in the same clause.

Arguments vs. Adjuncts: Some Tests

- Adjuncts can be **iterated** and **reordered**. (Arguments can't).

Arguments vs. Adjuncts: Some Tests

- Adjuncts can be **iterated** and **reordered**. (Arguments can't).
- Arguments are located **next to the head** (in their canonical form); they can't be put anywhere else.

Arguments vs. Adjuncts: Some Tests

- Adjuncts can be **iterated** and **reordered**. (Arguments can't).
- Arguments are located **next to the head** (in their canonical form); they can't be put anywhere else.
- When we want to use coordination on arguments and adjuncts, we can **coordinate arguments with arguments** and **adjuncts with adjuncts**, but we cannot mix the two.
  - I *waited* *for the bus and the car*.
  - *I *waited* *for the bus and hours*.

Semantic Role Labeling

Semantic role labeling (SRL)

- identifies arguments as well as adjuncts (harder)
- label their semantic roles (easier)

Supervised learning provides popular solutions.

Semantic Role Labeling

Semantic role labeling (SRL)

- identifies arguments as well as adjuncts (harder)
- label their semantic roles (easier)

Supervised learning provides popular solutions.

Constituency-based SRL

- CoNLL shared task 2004 and 2005

Semantic Role Labeling

Semantic role labeling (SRL)

- identifies arguments as well as adjuncts (harder)
- label their semantic roles (easier)

Supervised learning provides popular solutions.

Constituency-based SRL

- CoNLL shared task 2004 and 2005

Dependency-based SRL

- CoNLL shared task 2008 and 2009

VerbNet and Unified Verb Index for English

https://uvi.colorado.edu/

VerbNet and Unified Verb Index for English

`https://uvi.colorado.edu/`

VerbNet: An extension of Levin (1993).

- Actor
- Agent
- Beneficiary
- Theme
- etc.

VerbNet and Unified Verb Index for English

`https://uvi.colorado.edu/`

VerbNet: An extension of Levin (1993).

- Actor
- Agent
- Beneficiary
- Theme
- etc.

PropBank: Annotations of semantic roles, based on the Penn Treebank.

- Arg0/A0: proto-Agent
- Arg1/A1: proto-Patient
- Arg2–6: verb-specific roles
- ArgM-Manner: adjuncts
- ArgM-...

Semantic Role Labeling as Sequence Labeling

The B-I-O annotation scheme:

- B-*: Beginning of a span
- I-*: Inside of a span
- O: Outside of any span

Semantic Role Labeling as Sequence Labeling

The B-I-O annotation scheme:

- B-*: Beginning of a span
- I-*: Inside of a span
- O: Outside of any span

The	explosive	eruption	of	Hunga-Tonga	Hunga-Ha'apai
B-A0	I-A0	I-A0	I-A0		I-A0

sent	a	shockwave	around	the	world.
V.1	B-A1	I-A1	B-ArgM	I-ArgM	I-ArgM

Semantic Role Labeling as Sequence Labeling

The B-I-O annotation scheme:

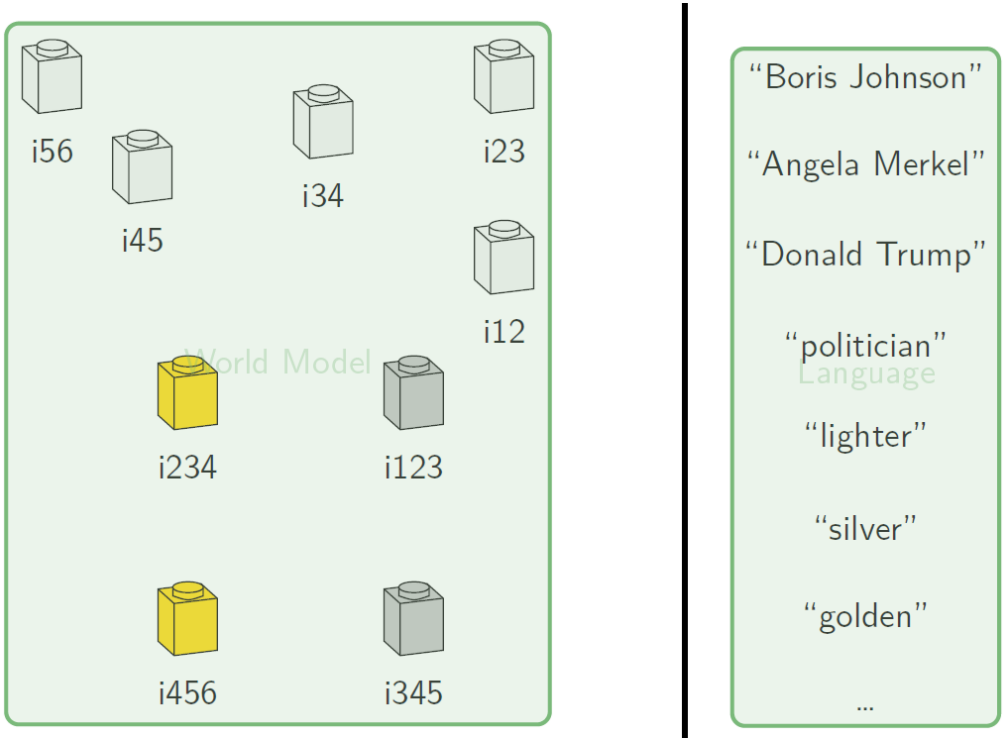
- B-*: Beginning of a span
- I-*: Inside of a span
- O: Outside of any span

The	explosive	eruption	of	Hunga-Tonga	Hunga-Ha'apai
B-A0	I-A0	I-A0	I-A0		I-A0

sent	a	shockwave	around	the	world.
V.1	B-A1	I-A1	B-ArgM	I-ArgM	I-ArgM

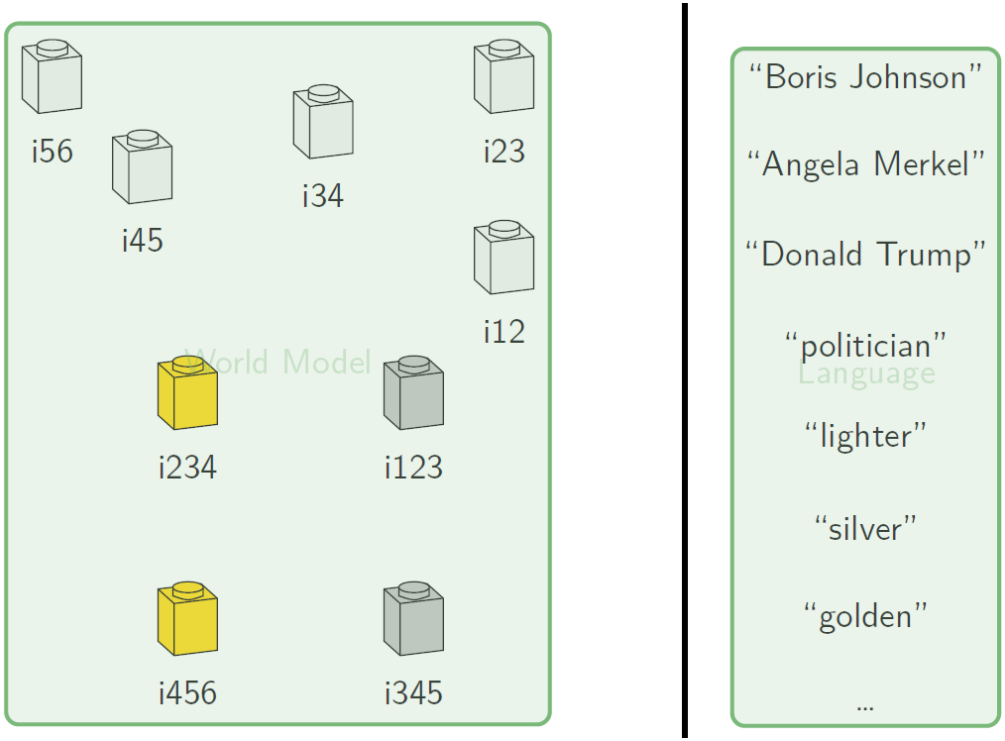
Any sequence labeling models we learned before can be applied here!

Set-Theoretic semantics



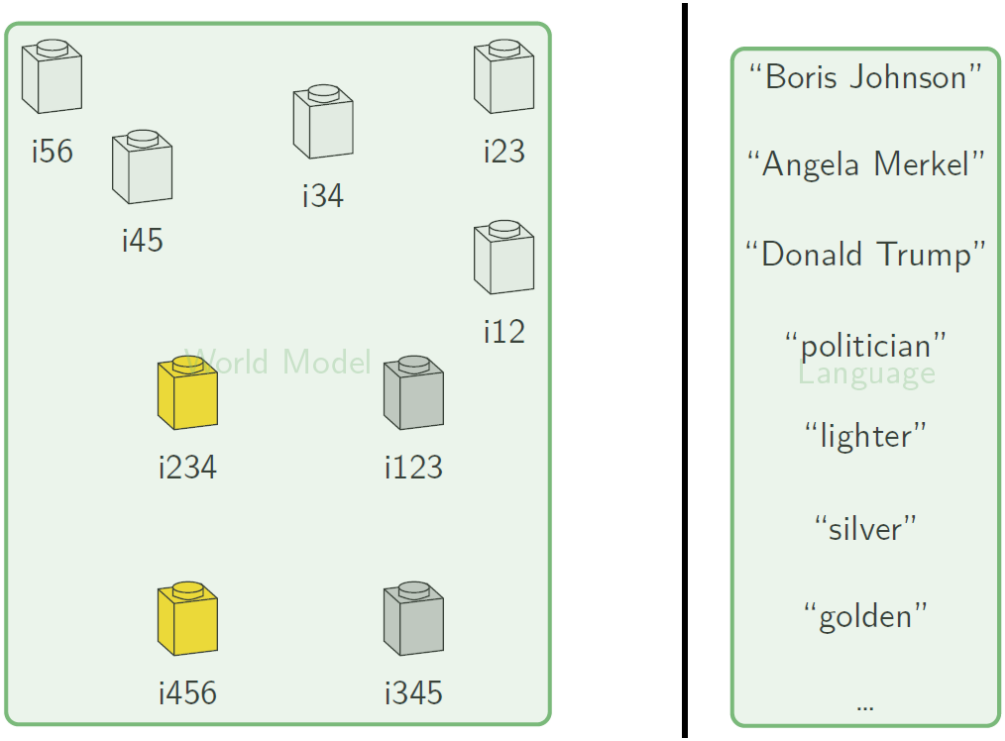
- The semantics is defined in terms of Set Theory.

Set-Theoretic semantics



- The semantics is defined in terms of Set Theory.
- $[\![\text{Trump}]\!] = i12$, where $[\![\cdot]\!]$ is called the **evaluation function** that maps words to their meanings.

Set-Theoretic semantics



- The semantics is defined in terms of Set Theory.
- $[\![\text{Trump}]\!] = i12$, where $[\![\cdot]\!]$ is called the **evaluation function** that maps words to their meanings.
- How can we easily and precisely describe sets as well as operations over sets?

Predicates are functions and sets

Q: What is the meaning of *politician*? **A**: `politician`

Predicates are functions and sets

Q: What is the meaning of *politician*? **A**: `politician`

`politician` is a set (of all politician entities).

Predicates are functions and sets

Q: What is the meaning of *politician*? **A**: `politician`

`politician` is a set (of all politician entities).

It's can be also viewed as a "membership" function, where each entity is mapped to a boolean value (true means the entity is a politician).

$$\texttt{politician}(i_{23}) = \mathit{true}$$
$$\texttt{politician}(i_{34}) = \mathit{false}$$

Predicates are functions and sets

Q: What is the meaning of *politician*? **A**: `politician`

`politician` is a set (of all politician entities).

It's can be also viewed as a "membership" function, where each entity is mapped to a boolean value (true means the entity is a politician).

$$\texttt{politician}(i_{23}) = \mathit{true}$$
$$\texttt{politician}(i_{34}) = \mathit{false}$$

λ-calculus offers a minimal programming language to describe such functions.

λ-calculus

- β-reduction/function application:

$$[\lambda x.M](N) \longrightarrow M[x := N]$$

  Apply a λ-term to an argument $x = N$, and get a value (not limited to true or false).

[Source: `https://plato.stanford.edu/entries/lambda-calculus/`]

λ-calculus allows us to build functions in a very convenient way.

λ-calculus

- β-reduction/function application:

$$[\lambda x.M](N) \longrightarrow M[x := N]$$

Apply a λ-term to an argument $x = N$, and get a value (not limited to true or false).

[Source: https://plato.stanford.edu/entries/lambda-calculus/]

λ-calculus allows us to build functions in a very convenient way.

- $f(x) = x^2 \longleftrightarrow [\lambda x.[x^2]]$

λ-calculus

- β-reduction/function application:

$$[\lambda x.M](N) \longrightarrow M[x := N]$$

  Apply a λ-term to an argument $x = N$, and get a value (not limited to true or false).

[Source: https://plato.stanford.edu/entries/lambda-calculus/]

λ-calculus allows us to build functions in a very convenient way.

- $f(x) = x^2 \longleftrightarrow [\lambda x.[x^2]]$
- $f(5) = 25 \longleftrightarrow$

λ-calculus

- β-reduction/function application:

$$[\lambda x.M](N) \longrightarrow M[x := N]$$

Apply a λ-term to an argument $x = N$, and get a value (not limited to true or false).

[Source: `https://plato.stanford.edu/entries/lambda-calculus/`]

λ-calculus allows us to build functions in a very convenient way.

- $f(x) = x^2 \longleftrightarrow [\lambda x.[x^2]]$
- $f(5) = 25 \longleftrightarrow [\lambda x.[x^2]](5) = 25$

λ-calculus

- β-reduction/function application:

$$[\lambda x.M](N) \longrightarrow M[x := N]$$

Apply a λ-term to an argument $x = N$, and get a value (not limited to true or false).

[Source: https://plato.stanford.edu/entries/lambda-calculus/]

λ-calculus allows us to build functions in a very convenient way.

- $f(x) = x^2 \longleftrightarrow [\lambda x.[x^2]]$
- $f(5) = 25 \longleftrightarrow [\lambda x.[x^2]](5) = 25$
- $g(x, y) = x^2 + y^2 \longleftrightarrow [\lambda x.[\lambda y.[x^2 + y^2]]]$

λ-calculus

- β-reduction/function application:

$$[\lambda x.M](N) \longrightarrow M[x := N]$$

Apply a λ-term to an argument $x = N$, and get a value (not limited to true or false).

[Source: https://plato.stanford.edu/entries/lambda-calculus/]

λ-calculus allows us to build functions in a very convenient way.

- $f(x) = x^2 \longleftrightarrow [\lambda x.[x^2]]$
- $f(5) = 25 \longleftrightarrow [\lambda x.[x^2]](5) = 25$
- $g(x, y) = x^2 + y^2 \longleftrightarrow [\lambda x.[\lambda y.[x^2 + y^2]]]$
- $g(2, 1) = 5 \longleftrightarrow [\lambda x.[\lambda y.[x^2 + y^2]]](2)(1) = 5$

λ-calculus

- β-reduction/function application:

$$[\lambda x.M](N) \longrightarrow M[x := N]$$

Apply a λ-term to an argument $x = N$, and get a value (not limited to true or false).

[Source: https://plato.stanford.edu/entries/lambda-calculus/]

λ-calculus allows us to build functions in a very convenient way.

- $f(x) = x^2 \longleftrightarrow [\lambda x.[x^2]]$
- $f(5) = 25 \longleftrightarrow [\lambda x.[x^2]](5) = 25$
- $g(x, y) = x^2 + y^2 \longleftrightarrow [\lambda x.[\lambda y.[x^2 + y^2]]]$
- $g(2, 1) = 5 \longleftrightarrow [\lambda x.[\lambda y.[x^2 + y^2]]](2)(1) = 5$

Deep recursive functions $f(f(f(f(f(\ldots)))))$ can be annoying though.

Simple Types

From a nonempty set **BasTyp** of **basic types**, the set **Typ** is the smallest set such that

- **BasTyp** $\subseteq$ **Typ**,
- $\langle \sigma, \tau \rangle \in$ **Typ** if $\sigma, \tau \in$ **Typ**.

A type of form $\langle \sigma, \tau \rangle$ is said to be a **functional type**.

Simple Types

From a nonempty set **BasTyp** of **basic types**, the set **Typ** is the smallest set such that

- **BasTyp** $\subseteq$ **Typ**,
- $\langle \sigma, \tau \rangle \in$ **Typ** if $\sigma, \tau \in$ **Typ**.

A type of form $\langle \sigma, \tau \rangle$ is said to be a **functional type**.

Example

- Assume **e** for individuals and **t** for `true`/`false`,

Simple Types

From a nonempty set **BasTyp** of **basic types**, the set **Typ** is the smallest set such that

- **BasTyp** $\subseteq$ **Typ**,
- $\langle \sigma, \tau \rangle \in$ **Typ** if $\sigma, \tau \in$ **Typ**.

A type of form $\langle \sigma, \tau \rangle$ is said to be a **functional type**.

Example

- Assume **e** for individuals and **t** for `true`/`false`,
- then $\langle \mathbf{e}, \mathbf{t} \rangle$ is the type for unary relations,

Simple Types

From a nonempty set **BasTyp** of **basic types**, the set **Typ** is the smallest set such that

- **BasTyp** $\subseteq$ **Typ**,
- $\langle \sigma, \tau \rangle \in$ **Typ** if $\sigma, \tau \in$ **Typ**.

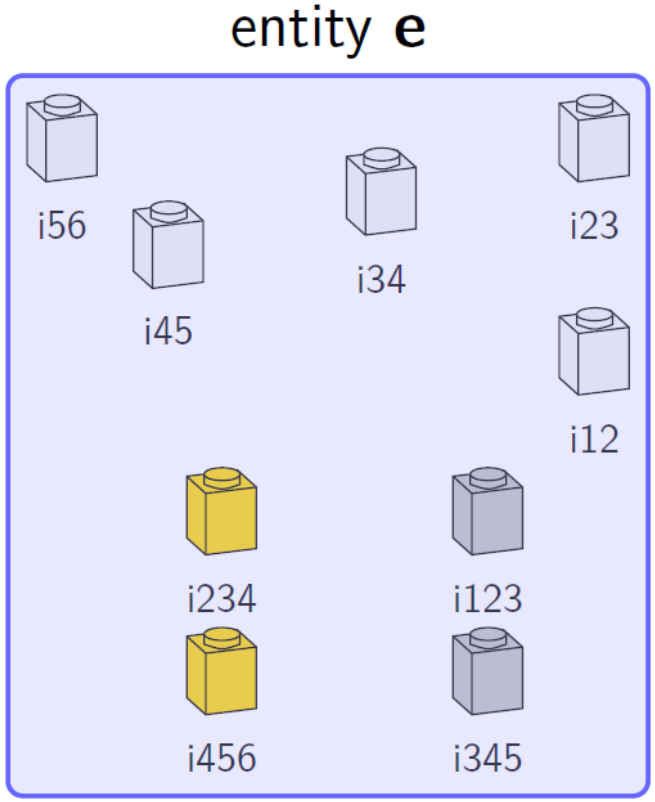
A type of form $\langle \sigma, \tau \rangle$ is said to be a **functional type**.

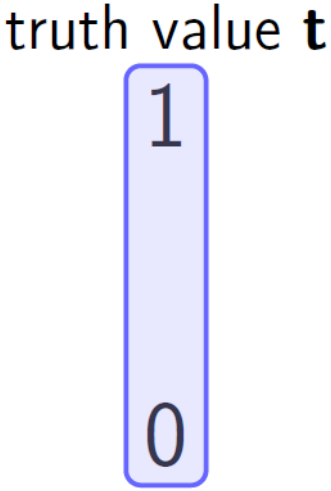
Example

- Assume **e** for individuals and **t** for `true`/`false`,
- then $\langle \mathbf{e}, \mathbf{t} \rangle$ is the type for unary relations,
- and $\langle \langle \mathbf{e}, \mathbf{t} \rangle, \langle \mathbf{e}, \mathbf{t} \rangle \rangle$ is for the type of a function mapping unary relations into unary relations.

e, t, and e to t

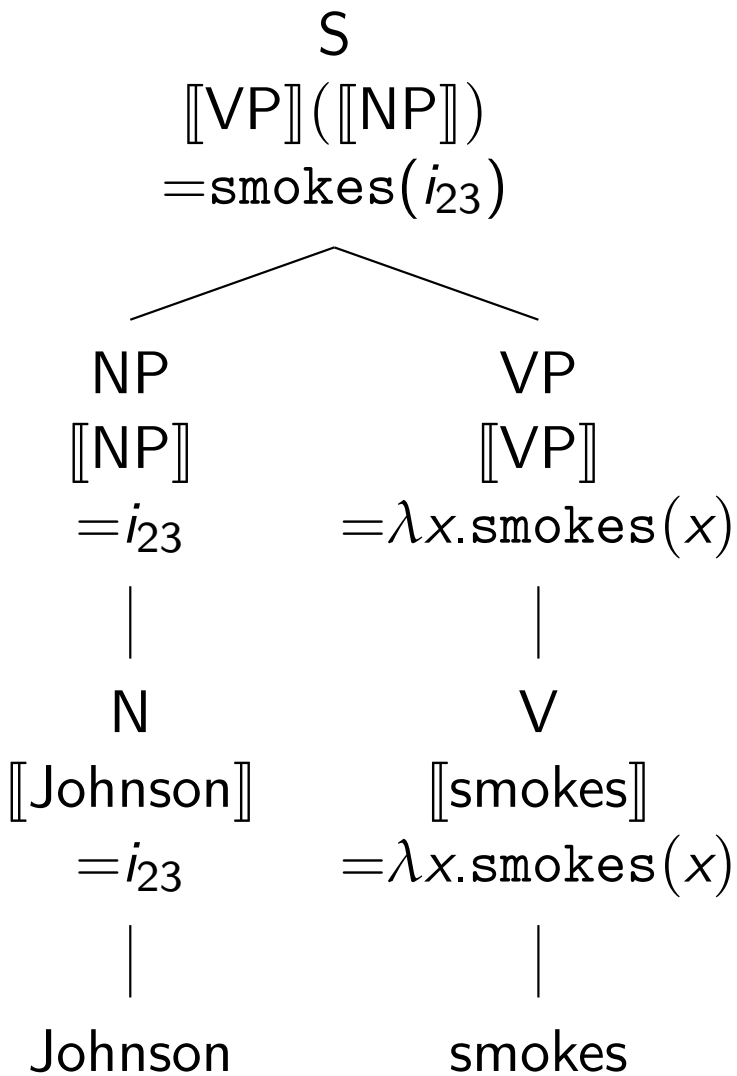
Gottlob Frege: There are only two atomic things, truth values and individuals. All other things are created by function application.





$\lambda x.\texttt{gold}(x)$: $\langle \mathbf{e}, \mathbf{t} \rangle$
$\lambda x.\texttt{silver}(x)$: $\langle \mathbf{e}, \mathbf{t} \rangle$

Syntactico-Semantic Composition



Compositional Semantics

- ⟦Johnson smokes⟧ is not listed in the lexicon.

Compositional Semantics

- ⟦Johnson smokes⟧ is not listed in the lexicon.
- But the interpretation of Johnson smokes can still be derived from its parts along with a syntactic analysis.

Compositional Semantics

- ⟦Johnson smokes⟧ is not listed in the lexicon.
- But the interpretation of Johnson smokes can still be derived from its parts along with a syntactic analysis.
- **Compositional semantics**: make infinite interpretations possible with finite basic elements.

Compositional Semantics

- ⟦Johnson smokes⟧ is not listed in the lexicon.
- But the interpretation of Johnson smokes can still be derived from its parts along with a syntactic analysis.
- **Compositional semantics**: make infinite interpretations possible with finite basic elements.
- ... and note that we have remained precise!

Meanings as Truth Conditions

To know the meaning of a sentence is to know how the world would have to be for the sentence to be true.

—Ludwig Wittgenstein

Meanings as Truth Conditions

> *To know the meaning of a sentence is to know how the world would have to be for the sentence to be true.*
>
> —*Ludwig Wittgenstein*

The meaning of words and sentence parts is their contribution to the truth-conditions of the whole sentence.

Meanings as Truth Conditions

> *To know the meaning of a sentence is to know how the world would have to be for the sentence to be true.*
>
> —*Ludwig Wittgenstein*

The meaning of words and sentence parts is their contribution to the truth-conditions of the whole sentence.

This is where we apply first-order logic, as well as temporal and modal logics.

Natural Language to Code Translation as Semantic Parsing

Geoquery: `https://www.cs.utexas.edu/~ml/geo.html`

Translate natural language questions to executable programs (e.g., SQL query).

Natural Language to Code Translation as Semantic Parsing

Geoquery: `https://www.cs.utexas.edu/~ml/geo.html`

Translate natural language questions to executable programs (e.g., SQL query).

Given that large Transformer-based models have some understanding of language, we can use them for these tasks.

Natural Language to Code Translation as Semantic Parsing

Geoquery: `https://www.cs.utexas.edu/~ml/geo.html`

Translate natural language questions to executable programs (e.g., SQL query).

Given that large Transformer-based models have some understanding of language, we can use them for these tasks.

Execution-aware semantic parsing with large language models: `https://aclanthology.org/2022.emnlp-main.231.pdf`

Next

Grounded semantics