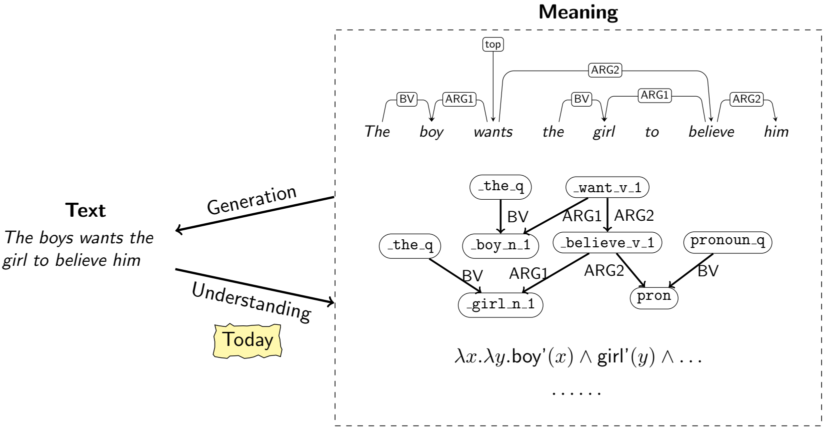
Lecture 5: Graph-Based Meaning Representations

Weiwei Sun

Department of Computer Science and Technology
University of Cambridge

Michaelmas 2024/25



Lecture 5: Graph-Based Meaning Representations

1. From logical forms to semantic graphs
2. String to graph parsing
3. Factorisation-based approach

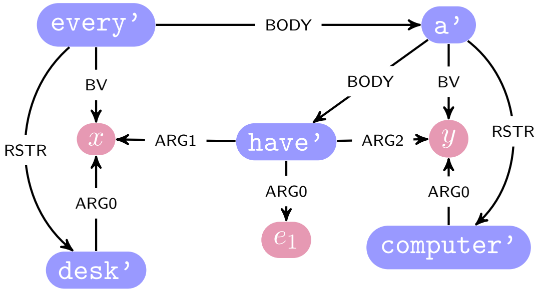
From Logical Forms to Semantic Graphs

Different representations of logical forms

- *Every desk has a computer* more in later lecture on scope
- $\forall x(\text{desk'}(x) \rightarrow (\exists y(\text{computer'}(y) \wedge \text{have'}(e, x, y))))$
- $\text{every'}(x, \text{desk'}(x), \text{a'}(y, \text{computer'}(y), \text{have'}(e, x, y)))$

Different representations of logical forms

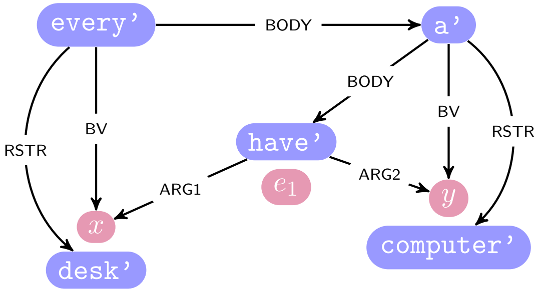
- *Every desk has a computer* more in later lecture on scope
- $\forall x(\text{desk'}(x) \rightarrow (\exists y(\text{computer'}(y) \wedge \text{have'}(e, x, y))))$
- $\text{every'}(x, \text{desk'}(x), \text{a'}(y, \text{computer'}(y), \text{have'}(e, x, y)))$



ARG0 identifies the word that "introduces" a variable, which corresponds to discourse referent.

Different representations of logical forms

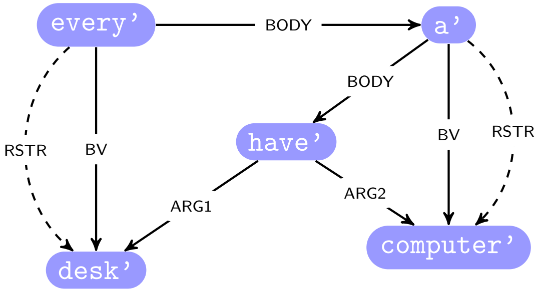
- *Every desk has a computer* more in later lecture on scope
- $\forall x(\text{desk'}(x) \rightarrow (\exists y(\text{computer'}(y) \wedge \text{have'}(e, x, y))))$
- $\text{every'}(x, \text{desk'}(x), \text{a'}(y, \text{computer'}(y), \text{have'}(e, x, y)))$



ARG0 identifies the word that "introduces" a variable, which corresponds to discourse referent.

Different representations of logical forms

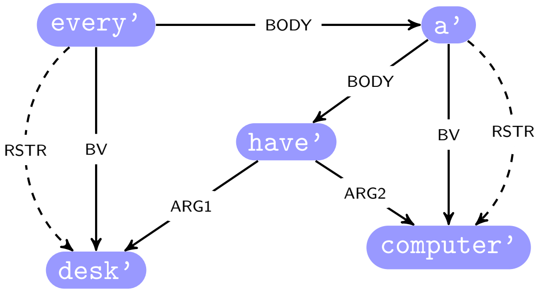
- *Every desk has a computer* more in later lecture on scope
- $\forall x(\text{desk'}(x) \rightarrow (\exists y(\text{computer'}(y) \wedge \text{have'}(e, x, y))))$
- $\text{every'}(x, \text{desk'}(x), \text{a'}(y, \text{computer'}(y), \text{have'}(e, x, y)))$



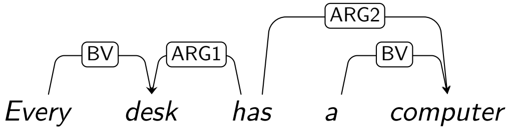
ARG0 identifies the word that "introduces" a variable, which corresponds to discourse referent.

Different representations of logical forms

- *Every desk has a computer* more in later lecture on scope
- $\forall x(\text{desk'}(x) \rightarrow (\exists y(\text{computer'}(y) \wedge \text{have'}(e, x, y))))$
- $\text{every'}(x, \text{desk'}(x), \text{a'}(y, \text{computer'}(y), \text{have'}(e, x, y)))$



ARG0 identifies the word that "introduces" a variable, which corresponds to discourse referent.



Bi-lexical semantic dependency graphs

- Projecting "concept nodes" to "words".
- Relations between "concepts" $\Rightarrow$ bi-lexical semantic dependencies
- Reasonably good though not as expressive as *conceptual* graphs.

Bi-lexical semantic dependency graphs

- Projecting "concept nodes" to "words".
- Relations between "concepts" $\Rightarrow$ bi-lexical semantic dependencies
- Reasonably good though not as expressive as *conceptual* graphs.

Weakness of bi-lexical semantic dependency graphs

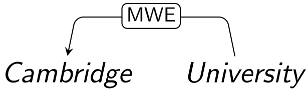
Bi-lexical semantic dependency graphs

- Projecting "concept nodes" to "words".
- Relations between "concepts" $\Rightarrow$ bi-lexical semantic dependencies
- Reasonably good though not as expressive as *conceptual* graphs.

Weakness of bi-lexical semantic dependency graphs

What are the triggers of concepts?

- MWE:



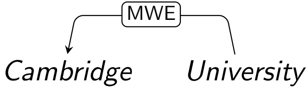
Bi-lexical semantic dependency graphs

- Projecting “concept nodes” to “words”.
- Relations between “concepts” ⇒ bi-lexical semantic dependencies
- Reasonably good though not as expressive as *conceptual* graphs.

Weakness of bi-lexical semantic dependency graphs

What are the triggers of concepts?

- MWE:



Modification

String-to-Graph Parsing

String to Graph parsing

- String to Graph parsing is the task of turning a string into a semantic graph
- We always need to solve two main subtasks:
  - Task 1: Concept Identification
  - Task 2: Relation Extraction
- For some semantic graphs such as EDS we additionally need to do concept-to-word alignment (task 0).

String to Graph parsing: TASKS 1 and 2

- There are two methods for performing tasks 1 and 2.
- One is based on Factorisation
  - Make some global analysis
  - eg. for Task1, use sequence-labelling for concept identification
  - For task 2, use Maximum Subgraph Parsing for relation detection (bilinearity can be applied)
  - This is the classic approach for AMR parsing, but any semantic graph can be parsed this way, even MRS.
- The other is based on Composition
  - local
  - judge quality of each composition step
  - computation is by classification over rules
  - Tasks 1 and 2 are performed in parallel
  - This is the classic approach for MRS.
  - With some limitations, can also be applied to AMR.

the drug was introduced in West Germany this year

the drug was introduced in West Germany this year

_the_q

_introduce_v_to

_year_n_1

_this_q_dem

_drug_n_1

_in_p

loc_nonsp

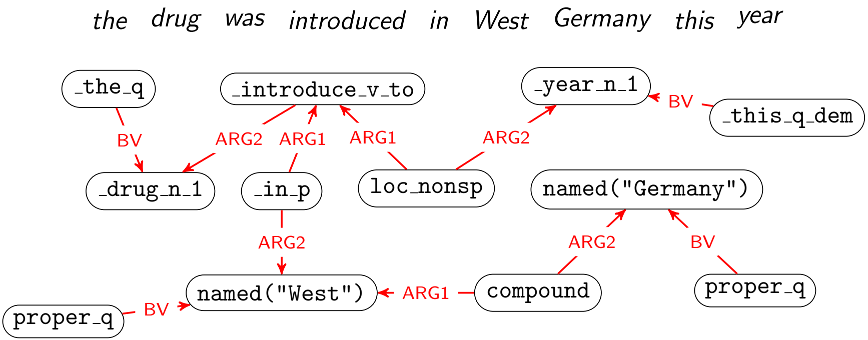
named("Germany")

named("West")

compound

proper_q

proper_q


the drug was introduced in West Germany this year
_the_q
_introduce_v_to
_year_n_1
BV
_this_q_dem
ARG2
ARG1
ARG1
ARG2
BV
_drug_n_1
_in_p
loc_nonsp
named("Germany")
ARG2
ARG2
BV
named("West")
ARG1
compound
proper_q
proper_q
BV

the drug was introduced in West Germany this year

_the_q _introduce_v_to _year_n_1 _this_q_dem

_drug_n_1 _in_p loc_nonsp named("Germany")

proper_q named("West") compound proper_q

Task 1: Concept Identification

the drug was introduced in West Germany this year

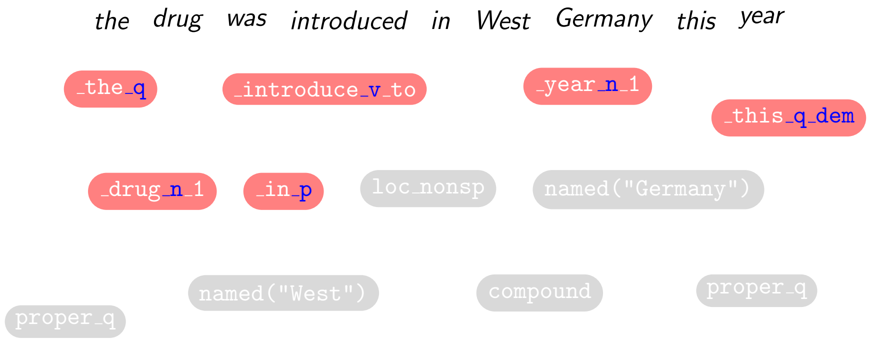
_the_q _introduce_v_to _year_n_1 _this_q_dem

_drug_n_1 _in_p loc_nonsp named("Germany")

named("West") compound proper_q

proper_q

Task 1: Concept Identification



Task 1: Concept Identification

the drug was introduced in West Germany this year

_the_q _introduce_v_to _year_n_1 _this_q_dem

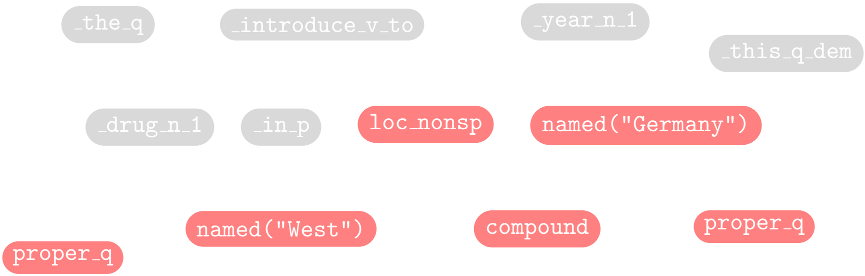
_drug_n_1 _in_p loc_nonsp named("Germany")

named("West") compound proper_q

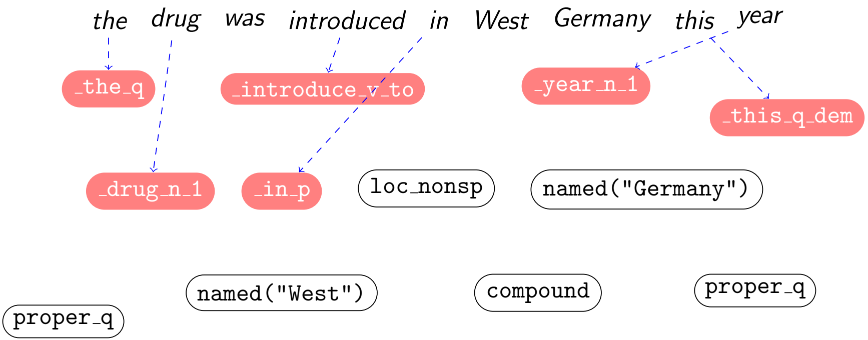
proper_q

Task 1: Concept Identification

the drug was introduced in West Germany this year

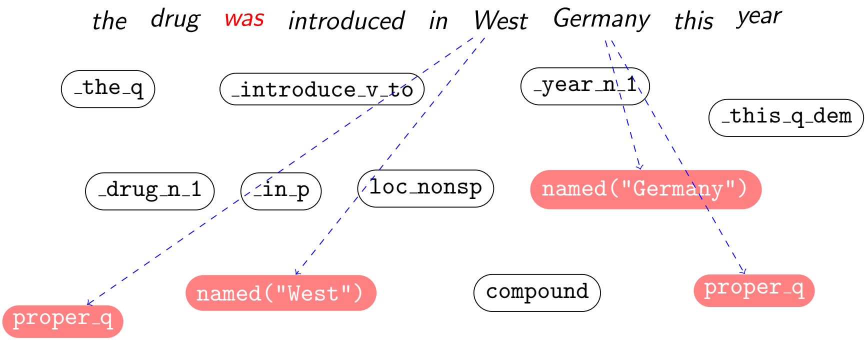


Task 1: Concept Identification



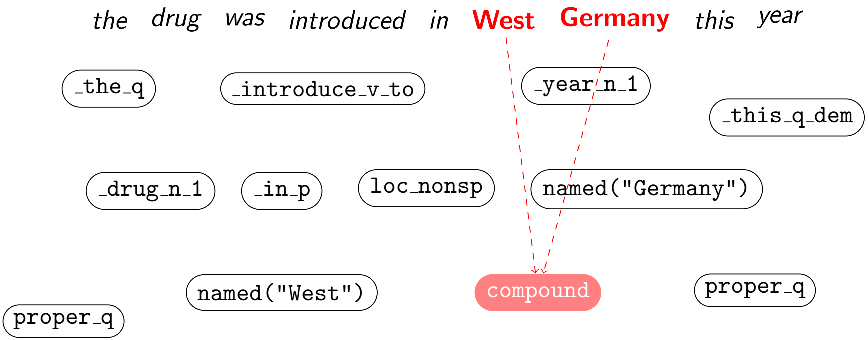
Task 0: Concept-to-word Alignment

Task 1: Concept Identification



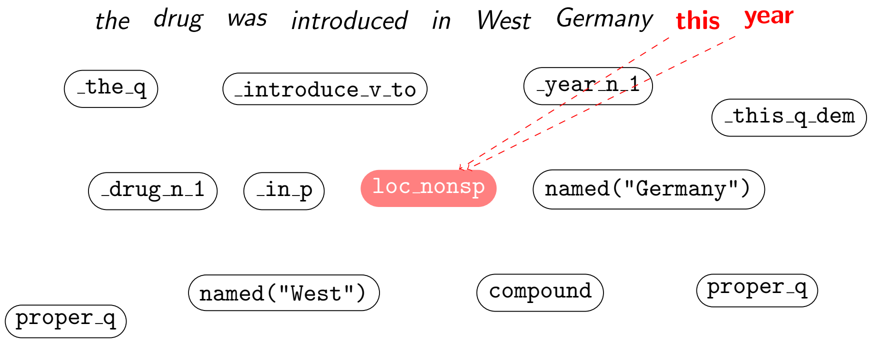
Task 0: Concept-to-word Alignment

Task 1: Concept Identification



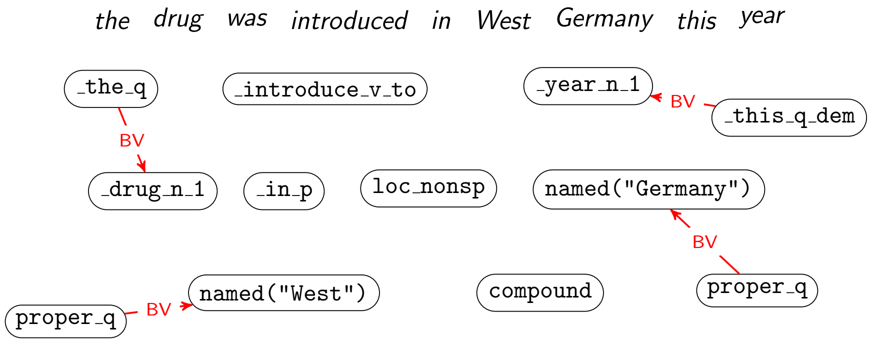
Task 0: Concept-to-word Alignment

Task 1: Concept Identification



Task 0: Concept-to-word Alignment

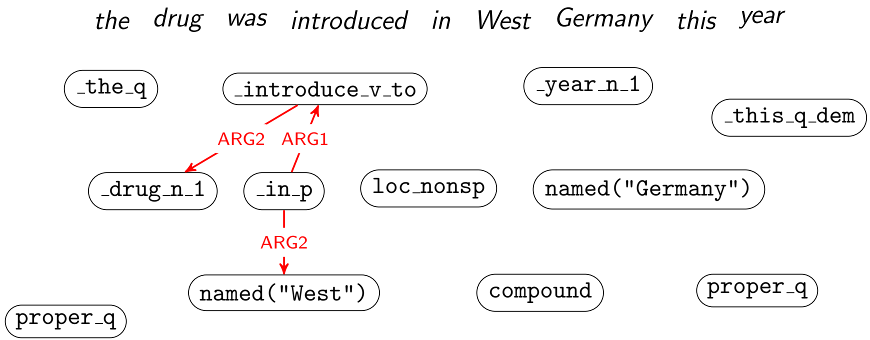
Task 1: Concept Identification



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

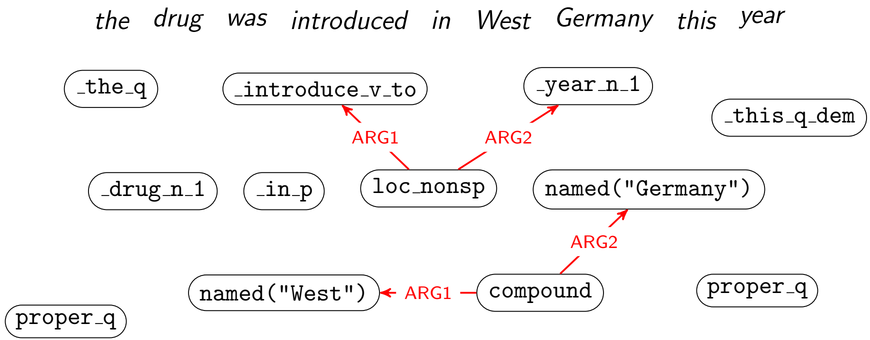
Task 2: Relation Detection



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

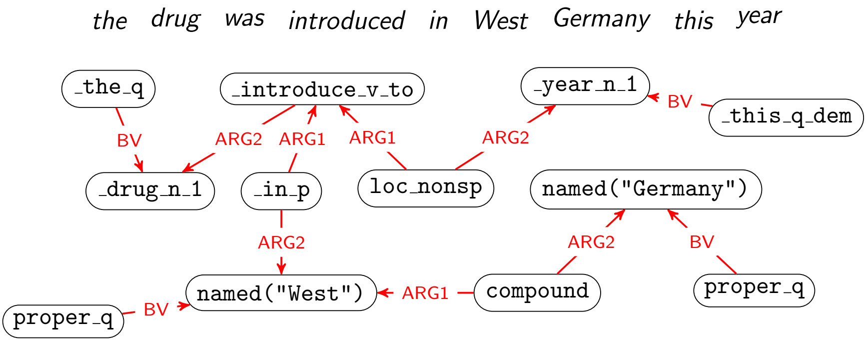
Task 2: Relation Detection



Task 0: Concept-to-word Alignment

Task 1: Concept Identification

Task 2: Relation Detection



Task 0: Concept-to-word Alignment

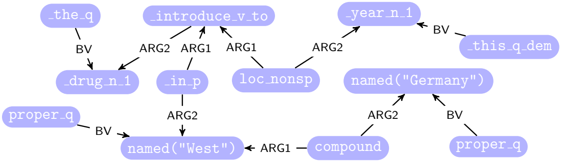
Task 1: Concept Identification

Task 2: Relation Detection

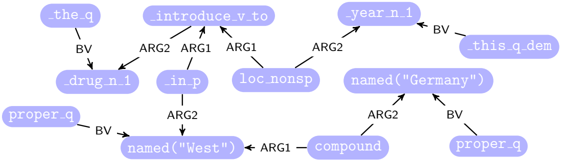
The three sub-tasks should be done ((explicitly or implicitly) and (directly or indirectly))

Factorisation-Based Approach

Concept Identification



Concept Identification



The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	∅	_introduce_v_to	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
					compound		loc_nonsp	

Concept Identification

The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	$\emptyset$	_introduce_v_to	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
					compound		loc_nonsp	

Almost Sequence Labeling

- Some nodes are linked to sub-words.
- Some nodes are linked to multiple words.

Solutions

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging
  - **B**-x: begin of x
  - **I**-x: inside x

Concept Identification

The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	∅	_introduce_v_to	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
					compound			loc_nonsp

Almost Sequence Labeling

- Some nodes are linked to sub-words.
- Some nodes are linked to multiple words.

Solutions

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging
  - **B**-x: begin of x
  - **I**-x: inside x

Concept Identification

The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	∅	_introduce_v_to	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
					B-compound	I-compound	B-loc_nonsp	I-loc_nonsp

Almost Sequence Labeling

- Some nodes are linked to sub-words.
- Some nodes are linked to multiple words.

Solutions

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging
  - **B**-x: begin of x
  - **I**-x: inside x

Concept Identification

The	drug	was	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	∅	_introduce_v_to	_in_p	named("W") proper_q	named("G") proper_q	_this_q_dem	_year_n_1
					compound		loc_nonsp	

Almost Sequence Labeling

- Some nodes are linked to sub-words.
- Some nodes are linked to multiple words.

Solutions

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging
  - **B**-x: begin of x
  - **I**-x: inside x

Neural Tagging

Almost Sequence Labeling

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging

Challenge

Like POS tagging but with thousands of labels.

Delexicalization

The	drug	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	_introduce_v_to	_in_p	named("W") compound proper_q	named("G") proper_q	_this_q_dem	_year_n_1 loc_nonsp

Neural Tagging

Almost Sequence Labeling

- Preprocessing: every node is assigned to a single word
- Chunking: joint segmentation and tagging

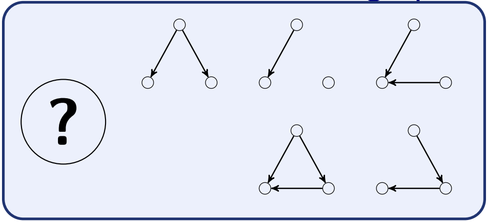
Challenge

Like POS tagging but with thousands of labels.

Delexicalization

The	drug	introduced	in	West	Germany	this	year
_the_q	_drug_n_1	_introduce_v_to	_in_p	named("W") compound proper_q	named("G") proper_q	_this_q_dem	_year_n_1 loc_nonsp
*_q	*_n_1	*_v_to	*_p	named compound proper_q	named proper_q	*_q_dem	*_n_1 loc_nonsp

Relation detection and maximum subgraph



Maximum Subgraph Parsing

- **Start from** a directed graph $G = (V, E)$ that corresponds to the input sentence and a score function that evaluates the *goodness* of a graph.
- **Search for** a *good* subgraph $G' = (V, E' \subseteq E)$:

$$G' = \arg \max_{G^* = (V, E^* \subseteq E)} \textsc{Score}(G^*)$$

First-order factorization

$$G' = \arg \max_{G^* = (V, E^* \subseteq E)} \sum_{e \in E^*} \textsc{ScorePart}(e)$$

Use the inner product space

Bilinearity

- $f(\alpha_1 + \alpha_2, \beta) = f(\alpha_1, \beta) + f(\alpha_2, \beta), \quad f(k\alpha, \beta) = kf(\alpha, \beta)$
- $f(\alpha, \beta_1 + \beta_2) = f(\alpha, \beta_1) + f(\alpha, \beta_2), \quad f(\alpha, k\beta) = kf(\alpha, \beta)$

If $\{e_1, e_2, ...e_n\}$ is a basis, then $f(e_i, e_j)$ ($\forall i, j : 1 \leq i, j \leq n$) identifies f.

Inner product $\langle \alpha, \beta \rangle$

A positive-definite symmetric bilinear function

- positive-definite: $\forall \alpha \neq \mathbf{0} : f(\alpha, \alpha) > 0$
- symmetric: $f(\alpha, \beta) = f(\beta, \alpha)$

Geometric intuitions

Inner product can be viewed as a generalisation of dot product. Inner products allow us to discuss *angles* and *lengths*.

$|\alpha| = \sqrt{\langle \alpha, \alpha \rangle}$ $\qquad$ $\cos(\alpha, \beta) = \frac{\langle \alpha, \beta \rangle}{|\alpha| \cdot |\beta|}$

Use the inner product space

Bilinearity

- $f(\alpha_1 + \alpha_2, \beta) = f(\alpha_1, \beta) + f(\alpha_2, \beta), \quad f(k\alpha, \beta) = kf(\alpha, \beta)$
- $f(\alpha, \beta_1 + \beta_2) = f(\alpha, \beta_1) + f(\alpha, \beta_2), \quad f(\alpha, k\beta) = kf(\alpha, \beta)$

If $\{e_1, e_2, ...e_n\}$ is a basis, then $f(e_i, e_j)$ ($\forall i, j : 1 \leq i, j \leq n$) identifies f.

Inner product $\langle \alpha, \beta \rangle$

A positive-definite symmetric bilinear function

- positive-definite: $\forall \alpha \neq \mathbf{0} : f(\alpha, \alpha) > 0$
- symmetric: $f(\alpha, \beta) = f(\beta, \alpha)$

Geometric intuitions

Inner product can be viewed as a generalisation of dot product. Inner products allow us to discuss *angles* and *lengths*.

$|\alpha| = \sqrt{\langle \alpha, \alpha \rangle}$ $\qquad\qquad$ $\cos(\alpha, \beta) = \frac{\langle \alpha, \beta \rangle}{|\alpha| \cdot |\beta|}$

Biaffine parsing

on whiteboard

Readings

- Bender, E.M., Flickinger, D., Oepen, S., Packard, W. and Copestake, A. Layers of interpretation: On grammar and compositionality. ICWS 2015.
- T. Dozat and C. Manning. Deep Biaffine Attention for Neural Dependency Parsing.
- S. Oepen, A. Koller and W. Sun. ACL Tutorial on Graph-Based Meaning Representations: Design and Processing.