WP2 - Semantic Parsing and Generation of Documents and Document Components

Claire Gardent

CNRS/LORIA, Nancy, France

ModelWriter, Izmir, January 2015

WP2: From Text to Data and Back

Expected Results
A reversible semantic processor which maps text to data (models) and data (models) to text.

Subtasks

- T2.1 – Data Collection (Text + Data Corpus)
- T2.2 – Hybrid approaches to semantic parsing (Text $\rightarrow$ Data)
- T2.3 – Hybrid approaches to Natural Language Generation (Data $\rightarrow$ Text)
- T2.4 – Definition of the target semantic representation language (Data format)
- T2.5 - Development of the semantic parser and of the generator

Example 1: Semantic Parsing

Aircraft Function Dictionary (AFD)

- Text: describes the architectural elements of aircrafts
- Database: aircraft components (e.g., engine), functions (e.g., generating heat) and operational conditions (e.g., landing conditions)
- *Semantic Parsing:* Automatically populates the DB from the text

Text: *This means a minimun clearance of 10 mm shall be ensured after LOAP*

$\downarrow$

DB distance(minimumClearance,10mm,after-LOAP)

Example 2: Semantic Parsing and Reasoning

System Installation Design Principles (SIDP)

- Text: Rules and requirements for system installation
- Model: Manually Formatted Rules
- *Semantic Parsing*: converts text to RDF tuples / DB entries
  - Automates rule conversion (currently converted manually)
    Minimun clearances detailed in table below shall be applied between parallel running bundles ...
    $\Rightarrow$ distance(minimumClearanceYY,tableXX,between-parallelRunningBundles)
  - automates conditions conversion (currently not converted)
    ... when installed in Wing or RFE
    $\Rightarrow$ when-condition(minimumClearanceYY,inWingORrfe)
  - indirectly supports reasoning (WP3) and question answering e.g.,
    What are the locations where a pipe of type T can be installed ?

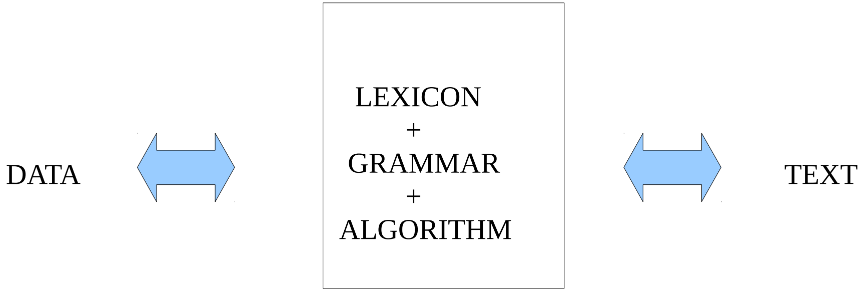
Example 3: Natural Language Generation

hasDepartureDate(Flight June17)
hasArrivalDate(Flight June20)
hasDestination(Flight Paris)
hasCarrier(Flight KLM)
hasTicket(Flight AirTicket)
hasDateOfIssue(AirTicket Today)

$\Rightarrow$

I am looking for a flight whose current departure date should be June 17th whose current arrival date should be June 20th and whose destination should be Paris. The carrier of the flight should be KLM. The ticket of the flight should be an air ticket whose date of issue is today

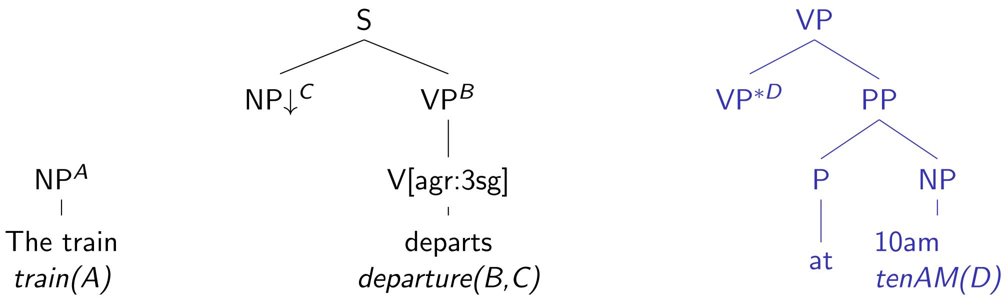
A Reversible Framework for Semantic Parsing and Generation (ANR Project WebNLG, EU Project ALLEGRO)



hasDepartureDate(Flight 2006)	*I am looking for a flight departing on June 17th*
hasArrivalDate(Flight 1706)	*arriving on June 20th*
hasDestination(Flight Paris)	*and whose destination is Paris.*
hasCarrier(Flight KLM)	*The carrier should be KLM*
hasTicket(Flight Ticket)	*The ticket of the flight should be a ticket*
hasDateofIssue(Ticket today)	*issued today.*

Feature-Based Lexicalised Tree Adjoining Grammar

Each grammar rule relates syntax, text and data.

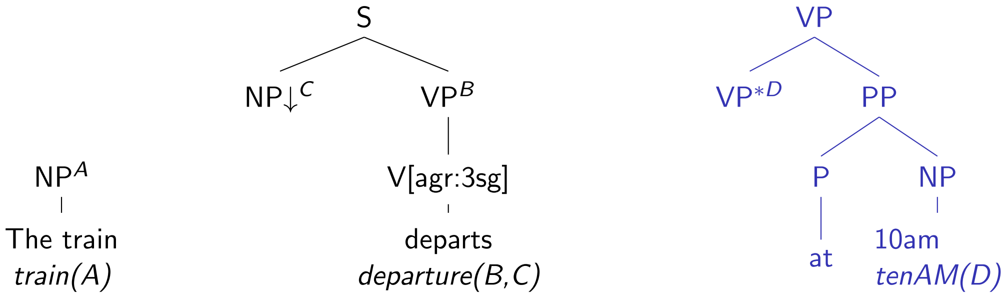


Grammar-Based Processing

`Generation Input:` *train(t), departure(e,t), tenAM(e)*

`Semantic Parsing Input:  The train departs at 10am`

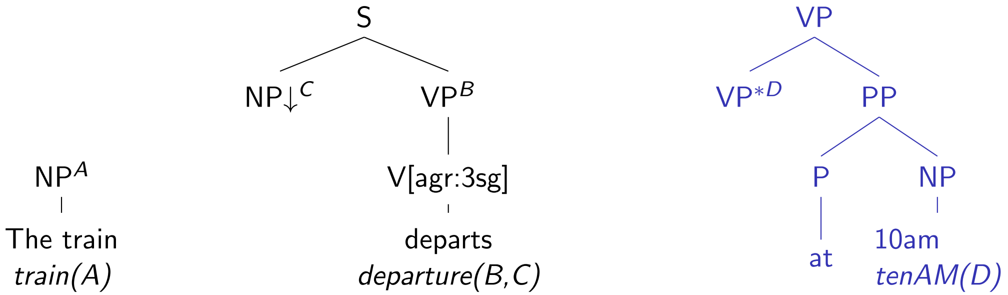
Grammar-Based Processing



`Generation Input:` *train(t), departure(e,t), tenAM(e)*

`Semantic Parsing Input: The train departs at 10am`

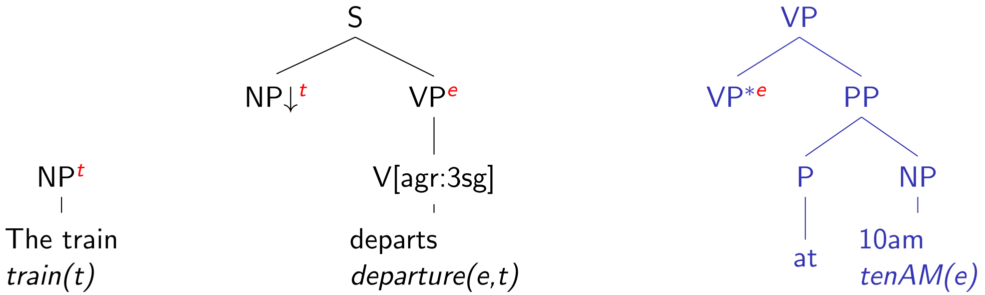
Grammar-Based Processing



`Generation Input:` *train(t), departure(e,t), tenAM(e)*

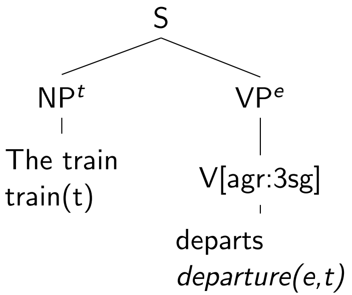
`Semantic Parsing Input:   The train departs at 10am`

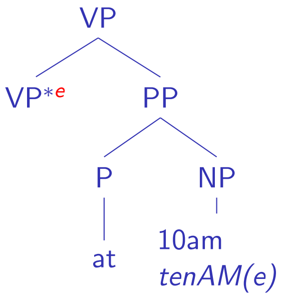
Grammar-Based Processing



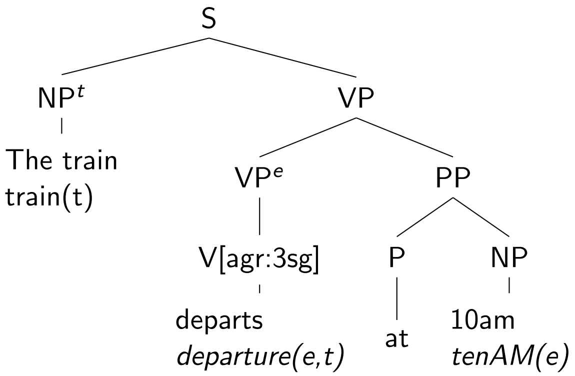
`Generation Input:` *train(t), departure(e,t), tenAM(e)*
`Semantic Parsing Input:` `The train departs at 10am`

Grammar-Based Processing

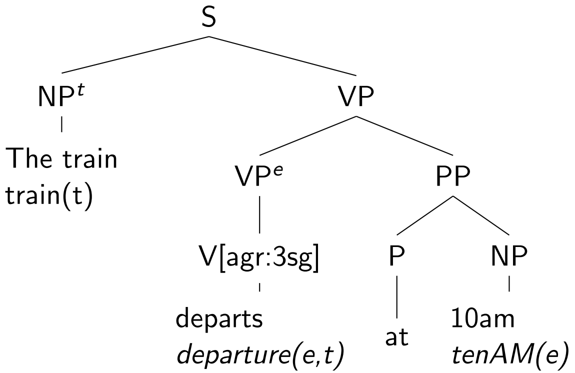




Grammar-Based Processing



Grammar-Based Processing



Generation Output: The train departs at 10am

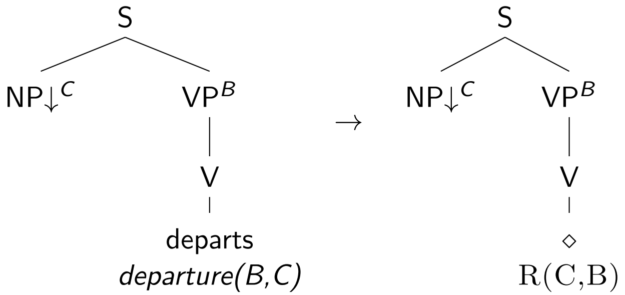
Semantic Parsing Output: *train(t), departure(e,t), tenAM(e)*

Separating Grammar from Lexicon

Since each tree is lexicalised, the resulting grammar can be very large. In practice, we therefore

- abstract over lexical items in the grammar
- use a lexicon to determine which grammar tree is lexicalised/anchored by which lexical items

Separating Grammar from Lexicon



Semantics: *departure*
Tree: nx0V
Anchor: *departs*

Semantics: *arrival*
Tree: nx0V
Anchor: *arrives*

…

Challenges

Data Collection

- Which data? Which text ? Which parsing/generation needs?

Automatic induction of grammar and lexicon

- Automate grammar and lexicon construction
  less time, less expertise required
- Learned mapping between natural language (words) and model vocabulary (e.g., RDF relations)
  increased portability
- Acquire probabilistic/frequency information (to control search)
  better efficiency

Parser and Generator Optimisation

Data Collection

Which data? Which text ? Which parsing/generation needs?

1. MW Usecases
    - Airbus: Synchronisation between regulation documents in natural language and DB rules
    - Others ?
    - We need: text (not pdf or png) and models (again in text format i.e., xml, rdf triples, ...)
2. Summarising/Simplifying Wikipedia pages
    - Semantic Parsing: Convert WKP text into RDF tuples
    - Select "relevant/important" tuples
    - Generation: Generate text from selected tuples

Automatic Induction of Grammar and Lexicon

Use Parsing and Relation Extraction to associate NL with KB relations and with Syntactic Rules

John wrote and directed Reverb

Parse	NP:John-subject-VBN:wrote-object-NP-Reverb	wrote $\Leftrightarrow$ nx0Vnx1
Relation Extraction	(John, authorOf, Reverb)	wrote $\Leftrightarrow$ authorOf

Lexicon	Semantics:	*authorOf*
	Tree:	nx0Vnx1
	Anchor:	*wrote*

Grammar nx0Vnx1 tree $+$ p(t), p(t|r), p(t|w)

Parser and Generator Optimisation

- Hypertagging: filter the initial space
  Train CRF (Conditional Random Field) on parallel corpus of text/data and Grammar Rules (Trees)
  Challenge: Automatic construction of training corpus
- Probabilistic processing: use (conditional) tree probability to control search (beam search)
- Ranking: choose best output
  Train statistical model (CRF,SVM) on parallel corpus of text/data and Grammar Derivations

Summary

Use relation extraction to induce a reversible grammar linking text, syntax and data

Optimise parsing and generation algorithm using statistical techniques to

- filter the initial search space
- guide beam search
- choose best output

Test reversible approach on wikipedia data and MW use cases

Extracting Relations

Text $\Rightarrow$ KB triples (Soderland et al 2010)

Sentence

"Blake threw three touchdown passes as New Orleans opened up a 28-0 halftime lead and cruised to a 31–15 victory over the 49ers."

Domain-independent Tuples

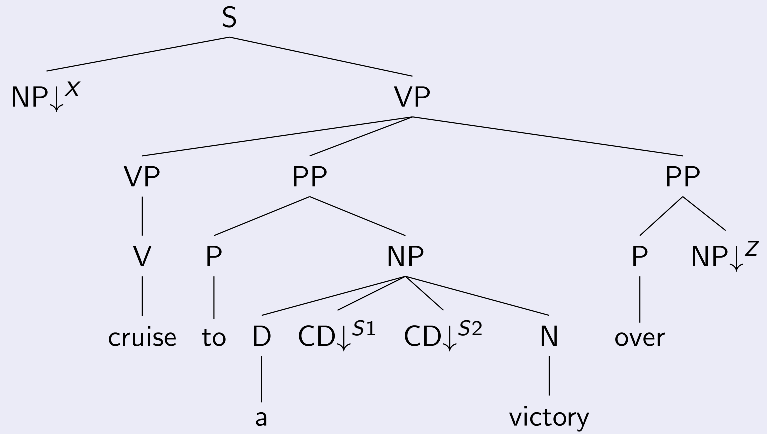
(New Orleans, cruised to, a 31–15 victory)
(a 31–15 victory, over, the 49ers)

Domain-specific Tuples

GameWinner(NFLGame:a 31–15 victory, NFLTeam:New Orleans)
GameLoser(NFLGame:a 31–15 victory, NFLTeam:the 49ers)
TeamScoringAll(NFLTeam:New Orleans, FinalScore:31)
TeamScoringAll(NFLTeam:the 49ers, FinalScore:15)

Grammar induction

Syntax (From Sentence and triples)



Semantics (From the Domain-specific Relations)

GameWinner(NFLGame:a S1–S2 victory, NFLTeam:X) GameLoser(NFLGame:a S1-S2 victory, NFLTeam:Z) TeamScoringAll(NFLTeam:X, FinalScore:S1) TeamScoringAll(NFLTeam:Z,

Open Issues

Which data/models and which texts ?

- Text type impacts semantic processing
  E.g., statistical parsers work well on “standard text”, less well e.g., on cooking recipe, instructions type text
- The delta between text and model signature/structure impacts semantic processing
  Mapping to RDF tuples is easier than mapping to code

Which text format ? Texte, xml, html.

- Pdf, doc, giff .. requires a conversion step which negatively impacts text quality and therefore semantic processing

Which representation language for data/models (RDF, OWL, UML, JAVA code, MDD4CCA ...)?

- Developing NLP tools to map text to different formats is costly

Which use cases ?

- Clearly identify use cases which involves semantic processing, semantic parsing and text generation

Thanks!