Making Choices
Statistical Microplanning

Claire Gardent
CNRS/LORIA and Université de Lorraine, Nancy

XRCE, October 2015
Grenoble, France
Generating from Knowledge-Bases NLG Approaches A Grammar-Based Statistical Approach for Microplanning

Joint Work with

Laura Perez-Beltrachini

Funded by the French ANR Project WebNLG
http://talc1.loria.fr/webnlg/stories/about.html
Generating from Knowledge-Bases NLG Approaches A Grammar-Based Statistical Approach for Microplanning

Writing/Producing a Text = Making Choices

What is talked about? (Content Selection)

Structuring the selected data into a text plan (Document planning)

Producing fluent text (Microplanning)

- Describing entities (Generating Referring Expressions)
- Choosing lexical items and syntactic structures (Lexicalisation, Surface Realisation, Aggregation, Sentence Segmentation)
Outline

1. Generating from Knowledge-Bases

2. NLG Approaches

3. A Grammar-Based Statistical Approach for Microplanning
Semantic Web and Knowledge-Bases

Ontologies

- Biomedical domain: SNOMED, GO, BioPAX, the Foundational Model of Anatomy and the U.S. National Cancer Institute Thesaurus
- Ontologies for e.g., geography, geology, agriculture and defence

Large scale RDF datasets

- DBPedia, Geonames, US Census, EuroStat, MusicBrainz, BBC Programmes, Flickr, DBLP, PubMed, UniProt, FOAF, SIOC, OpenCyc, UMBEL, Yagoo ...
Generating from Knowledge-Bases

Many applications could benefit from KB-to-Text generation.

Quelo Natural Language Interface

(a) Natural Language Interfaces for KB
Generating from Knowledge-Bases

Many applications could benefit from KB-to-Text generation.

![Diagram of the MIAKT Generator]

The MIAKT System: Generating Patient Report from RDF data

(b) Natural Language Descriptions of KB entities/concepts
Many applications could benefit from KB-to-Text generation.

<table>
<thead>
<tr>
<th>Class label</th>
<th>OWL axioms (Manchester syntax)</th>
<th>Natural Language Definition Extracted</th>
</tr>
</thead>
<tbody>
<tr>
<td>22rv1</td>
<td>bearer_of some 'prostate carcinoma' derives_from some 'Homo sapiens' derives_from some prostate</td>
<td>A 22rv1 is a cell line. A 22rv1 is all of the following: something that is bearer of a prostate carcinoma, something that derives from a homo sapiens, and something that derives from a prostate.</td>
</tr>
<tr>
<td>HeLa</td>
<td>bearer_of some 'cervical carcinoma' derives_from some 'Homo sapiens' derives_from some cervix derives_from some 'epithelial cell'</td>
<td>A he la is a cell line. A he la is all of the following: something that is bearer of a cervical carcinoma, something that derives from a homo sapiens, something that derives from an epithelial cell, and something that derives from a cervix.</td>
</tr>
<tr>
<td>Ara-C-resistant murine leukemia</td>
<td>has subclass b117h* has subclass b140h*</td>
<td>An ara c resistant murine leukemia is a cell line. A b117h, and a b140h are kinds of ara c resistant murine leukemias.</td>
</tr>
<tr>
<td>GM18507</td>
<td>derives_from some 'Homo sapiens' derives_from some lymphoblast has_quality some male</td>
<td>A gm18507 is all of the following: something that has as quality a male, something that derives from a homo sapiens, and something that derives from a lymphoblast.</td>
</tr>
</tbody>
</table>

The SWAT System: Verbalising KB Content

(c) Natural Language Presentation of KBs
Natural Language Generation

Manually or Automatically Acquired Templates (Duma et al. 2010, Blake et al. 2013, Schilder et al. 2013)

Align Text and Data

Create Template

Name (b.birthPlace, birthDate, d.deathPlace, deathDate) was a shortDescription.

Picture from Duma and Klein 2010
Natural Language Generation


Parallel Corpus

Learn Mapping

- Probabilistic CFG mapping DB to Text
- Cascaded Discriminative models
- Statistical Machine Translation

Picture from Konstas and Lapata 2013
Natural Language Generation


The NaturalOwl System: Describing individuals or classes of owl ontologies
Parallel Data-to-Text corpus is hard to get
Parallel Data-to-Text corpus is hard to get

Manually crafted grammars, lexicons and text plans are costly to develop
Parallel Data-to-Text corpus is hard to get

Manually crafted grammars, lexicons and text plans are costly to develop

Microplanning problem: grammatical ≠ fluent
KB to Text Generation

Parallel Data-to-Text corpus is hard to get.

Manually crafted grammars, lexicons and text plans are costly to develop.

Microplanning problem: grammatical ≠ fluent

? I am looking for a flight. Its departure date should be November 5th. The arrival date of the flight should be November 6th. The destination of the flight should be Paris.

✓ I am looking for a flight whose departure date should be November 5th, whose arrival date should be November 6th and whose destination should be Paris.
A Statistical Grammar-Based Approach

Input = KB Query
A Statistical Grammar-Based Approach

Input = KB Query
Segment Input, lexicalise KB symbols, aggregate and realise
A Statistical Grammar-Based Approach

Input = KB Query
Segment Input, lexicalise KB symbols, aggregate and realise

Professor ∩ Researcher ∩ ∃teach.LogicCourse
∩ ∃worksAt.AlicanteUniversity
A Statistical Grammar-Based Approach

Input = KB Query
Segment Input, lexicalise KB symbols, aggregate and realise

Professor □ Researcher □ ∃ teach.LogicCourse
□ ∃ worksAt.AlicanteUniversity

I am looking for a professor who is a researcher and teaches a course on logic.
He should work for Alicante University.
A Statistical Grammar-Based Approach

Combines a grammar, a lexicon with a surface realisation algorithm integrating a hypertagger, a beam search and a ranker
A Statistical Grammar-Based Approach

Combines a grammar, a lexicon with a surface realisation algorithm integrating a hypertagger, a beam search and a ranker

The grammar

- Defines the space of possible realisations
- Enforces hard constraints (grammaticality)

The Statistical Modules (Hypertagger, Beam Search, Ranker)

- Allow for efficiency (speed)
- Enforce soft constraints (fluency)
The Generation Algorithm

- Hypertagging: Selects the n-best sequences of grammar rules (TAG trees) given the input semantics
- Lexical Selection: retrieves TAG trees whose semantic subsumes the input and which are compatible with the hypertagger decisions
- Surface Realisation: Combines TAG trees to produce Sentences
- Ranking: Select n best outputs using Language Model
Grammar Based Generation

Input = KB Query

Professor ⊓ Researcher ⊓ ∃ teach.LogicCourse
⊓ ∃ worksAt.AlicanteUniversity
Grammar Based Generation

Input = KB Query

Professor ⊓ Researcher ⊓ ∃teach.LogicCourse
⊓ ∃worksAt.AlicanteUniversity

Professor(p) Researcher(p) teach(p c) LogicCourse(c) worksAt(p u)
AlicanteUniversity(u)
Grammar Based Generation

S

NP\downarrow^X \quad \text{NP}^T

V

should run

run(X,Y)

PP

P

on

NP\downarrow^Y \quad \text{NP}^T

Diesel
diesel(T)

The car
car(T)
Grammar Based Generation

\[ S \]
\[ NP \downarrow^c \]
\[ VP \]
\[ V \]
\[ PP \]
\[ \text{should run} \]
\[ \text{on} \]
\[ NP \downarrow^d \]
\[ \text{Diesel} \]
\[ car(c), run(c,d), diesel(d) \]
Grammar Based Generation

The car

\[ \text{car}(c) \]

should run

\[ \text{run}(c,d) \]

on

\[ \text{diesel}(d) \]

\[ \text{car}(c), \text{run}(c,d), \text{diesel}(d) \]
Grammar Based Generation

S

NP
The car
  car(c)

VP
  V
    should run
    run(c,d)

  PP
    P
      on
      diesel
      diesel(d)

\[car(c), \text{run}(c,d), \text{diesel}(d)\]
Grammar Based Generation

The car should run on diesel
The lexicalised grammar is very big (n * number of words) (tractability)
Grammar-Based Generation

The lexicalised grammar is very big \((n \times \text{number of words})\) (tractability)
\[\rightarrow\] Separate Grammar from Lexicon
Grammar and Lexicon

S
  NP↓X  VP
  |
  V  PP
  |
should run runOn(X, Y)
  |
  P  NP↓Y
  |
  on
Grammar and Lexicon

S
 NP↓X
 VP
 V
 PP
 R(X, Y)
 NP↓Y
 P

Semantics
runOn
Tree
nx0Vpnx1
Anchor
should run
Co-Anchor
P → on

Claire Gardent
Making Choices
Statistical Microplanning
Grammar and Lexicon

**S**

**NP**\( \downarrow X \)

**VP**

**V**

**PP**

\( R(X, Y) \)

\( \downarrow P \)

\( NP \downarrow \)

**Semantics**

**runOn**

**Tree**

\( nx0Vpnx1 \)

**Anchor**

should run

**Co-Anchor**

P \( \rightarrow \) on

**Semantics**

**assistWith**

**Tree**

\( nx0Vpnx1 \)

**Anchor**

should assist

**Co-Anchor**

P \( \rightarrow \) with
Grammar and Lexicon

The lexicon

- relates KB Symbols, Natural Language Expressions and Syntax (Grammar rules). It is *domain specific*.
- is acquired automatically
Grammar and Lexicon

The lexicon

- relates KB Symbols, Natural Language Expressions and Syntax (Grammar rules). It is domain specific.
- is acquired automatically

The grammar

- specifies the various syntactic realisations of words. It is generic.
- is a small, manually specified Tree Adjoining Grammar
Automatic Lexicon Induction

The lexicon is automatically derived from KB symbols (Trevisan 2010)
Automatic Lexicon Induction

The lexicon is automatically derived from KB symbols (Trevisan 2010)

Step 1: Tokenize and PoS Tag

\[ \text{runsOn} \rightarrow \text{runs/VBD on/IN} \]
Automatic Lexicon Induction

The lexicon is automatically derived from KB symbols (Trevisan 2010)

Step 1: Tokenize and PoS Tag

runsOn $\rightarrow$ runs/VBD on/IN

Step 2: The result sequence is mapped to one or more Lexical Entries
Automatic Lexicon Induction

The lexicon is automatically derived from KB symbols (Trevisan 2010)

**Step 1: Tokenize and PoS Tag**

\[ \text{runsOn} \rightarrow \text{runs/VBD on/IN} \]

**Step 2: The result sequence is mapped to one or more Lexical Entries**

\[ \text{runs/VBD on/IN} \rightarrow \]

- Semantics: \( \text{runsOn} \)
- Tree: \( \text{nx0Vpnx1} \)
- Anchor: should run
- Co-Anchor: P \( \rightarrow \) on
Generic Grammar

A small (100 trees), hand-written generic grammar models subcategorisation and syntactic variation.

Syntactic Variations

NP₀ should be equipped with NP₁

and NP₀ should be equipped with NP₁

NP₀ which should be equipped with NP₁

NP₀ (...) and which should be equipped with NP₁

NP₀ (...), which should be equipped with NP₁

NP₀ equipped with NP₁

NP₀ (...) and equipped with NP₁

NP₀ (...), equipped with NP₁

NP₁ with which NP₀ should be equipped

NP₀ (equipped with X) and with NP₁

NP₀ (equipped with X), with NP₁

Canonical

S-Coordination

SubjRel

SubjRelPU

SubjRelPU

PpartOrGerund

SharedSubj

SharedSubj

PObjRel

Ellipsis

Ellipsis
Generic Grammar

A small (100 trees), hand-written generic grammar models subcategorisation and syntactic variation.

Valency/Subcategorisation Variations

- $NP_0$ should generate $NP_1$  
  $nx0VV\mathit{V}n\times1$  
  Canonical

- $NP_0$ should run on $NP_1$  
  $nx0VV\mathit{V}p\times1$  
  Canonical

- $NP_0$ should be equipped with $NP_1$  
  $nx0VV\mathit{V}Vp\times1$  
  Canonical

- $NP_0$ should be the equipment of $NP_1$  
  $nx0VV\mathit{V}D\mathit{V}D\mathit{N}p\times1$  
  Canonical

- $NP_0$ should have access to $NP_1$  
  $nx0VV\mathit{V}Np\times1$  
  Canonical

- $NP_0$ should be relevant to $NP_1$  
  $nx0VV\mathit{V}A\mathit{p}p\times1$  
  Canonical

- $NP_0$ should be an $N_1$ product  
  $nx0VV\mathit{V}D\mathit{N}n\times1$  
  Canonical

- $NP_0$ with $NP_1$  
  $\beta\times0P\times1$  
  Canonical
Making Choices (Hypertagging)

The hypertagger

- Filters the initial search space (efficiency)
- Is trained to eliminate sequences of grammar trees that lead to less fluent sentences (fluency)
### Making Choices (Hypertagging)

#### Output of the Lexical Selection

<table>
<thead>
<tr>
<th>CarDealer(X) nx</th>
<th>locatedIn(X,Y) nx0VVVpnx1</th>
<th>City(Y) nx</th>
<th>sell(Y,Z) nx0VVVn×1</th>
<th>Car(Z) nx</th>
<th>runOn(Z,W) nx0VVpnx1</th>
<th>Diesel nx</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRO0VVVpnx1</td>
<td>sCONJn×0VVVVpnx1</td>
<td>sCONJn×0VVVVn×1</td>
<td>PRO0VVVn×1</td>
<td>sCONJn×0VVVVn×1</td>
<td>sCONJn×0VVVVn×1</td>
<td>...</td>
</tr>
<tr>
<td>ANDWHn×0VVVVpnx1</td>
<td>COMMAWHn×0VVVVpnx1</td>
<td>COMMAWHn×0VVVVn×1</td>
<td>ANDWHn×0VVVVn×1</td>
<td>ANDWHn×0VVVVn×1</td>
<td>ANDWHn×0VVVVn×1</td>
<td>...</td>
</tr>
<tr>
<td>betan×0VPpnx1</td>
<td>betan×0VPpnx1</td>
<td>betan×0VPpnx1</td>
<td>betan×0VPpnx1</td>
<td>betan×0VPpnx1</td>
<td>betan×0VPpnx1</td>
<td>...</td>
</tr>
<tr>
<td>betan×0ANDVPpnx1</td>
<td>betan×0ANDVPpnx1</td>
<td>betan×0ANDVPpnx1</td>
<td>betan×0ANDVPpnx1</td>
<td>betan×0ANDVPpnx1</td>
<td>betan×0ANDVPpnx1</td>
<td>...</td>
</tr>
<tr>
<td>W1pn×1nx0VV</td>
<td>W1pn×1nx0VV</td>
<td>W1pn×1nx0VV</td>
<td>W1pn×1nx0VV</td>
<td>W1pn×1nx0VV</td>
<td>W1pn×1nx0VV</td>
<td>...</td>
</tr>
<tr>
<td>betav×0ANDVVpnx1</td>
<td>betav×0ANDVVpnx1</td>
<td>betav×0ANDVVpnx1</td>
<td>betav×0ANDVVpnx1</td>
<td>betav×0ANDVVpnx1</td>
<td>betav×0ANDVVpnx1</td>
<td>...</td>
</tr>
<tr>
<td>betav×0COMMAVVVpnx1</td>
<td>betav×0COMMAVVVpnx1</td>
<td>betav×0COMMAVVVpnx1</td>
<td>betav×0COMMAVVVpnx1</td>
<td>betav×0COMMAVVVpnx1</td>
<td>betav×0COMMAVVVpnx1</td>
<td>...</td>
</tr>
</tbody>
</table>

**I am looking for a car dealer located in a city who should sell cars. The car should run on diesel.**
The **hypertagger** prunes the initial search space and favours Tree/Syntactic Classes sequences which yield fluent sentences.

\[
\text{CarDealer} \sqcap \exists \text{locatedIn.City} \sqcap \exists \text{sell.Car} \sqcap \exists \text{runOn.Diesel}
\]
Making Choices (Inversed Parsing and Ranking)

The **hypertagger** prunes the initial search space and favours Tree/Syntactic Classes sequences which yield fluent sentences.

\[
\text{CarDealer} \sqcap \exists \text{locatedIn.City} \sqcap \exists \text{sell.Car} \sqcap \exists \text{runOn.Diesel}
\]

\[
\text{Tbetanx0VPpnx1 TANDWHnx0VVnx1 Tnx0VVpnx1 Tnx}
\]

*I am looking for a car dealer located in a city and who should sell a car. The car should run on diesel.*

\[
\text{Tnx0VPpnx1 Tnx0VVVnx1 Tnx0VVVpnx1}
\]

*I am looking for a car dealer. He should be located in a city. He should sell a car. The car should run on diesel.*
## Hypertagging

A linear-chain Conditional Random Field model is used to define the posterior probability of labels (TAG trees, syntactic classes) $y = \{y_1, \ldots, y_n \}$ given features informed by the input semantics $x = \{x_1, \ldots, x_k \}$:

$$
P(y \mid x) = \frac{\exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \theta_j f_j(x, i, l_i, l_{i-1})]}{\sum_{l'} \exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \theta_j f_j(x, i, l'_i, l'_{i-1})]}
$$

Given a set of candidate hypertags (TAG trees) associated with each literal, the hypertagger finds the optimal hypertag sequence $y^*$ for a given input semantics $x$:

$$
y^* = \text{argmax}_y P(y \mid x)
$$
Experimental Setup

Grammar and Lexicon

- Grammar: 69 trees, 10 syntactic classes
- Lexicon: 13 KB, 10K entries, 1296 concepts and elations, average lexical ambiguity: 7.73.

Evaluation Metrics

- Hypertagging Accuracy
- Coverage and Speed
- Output quality (Human Evaluation)
- Qualitative Analysis (Microplanning)

Comparison Models

- Template-Based Model
- Symbolic Grammar-Based Model
Data

Training Data for the CRF

- 206 training instances = (KB query, tree sequence) pairs
- From 11 ontologies (Domain Independent)
- Input Length (min:2, max:19, avg: 7.44)
- CRF trained and tested using 10 fold cross validation

Features

- KB Symbol: Shape and content (words) of relation names (unigram and bigrams)
- Lexical features: word overlap between KB symbols, presence/absence of prepositions, etc.
- Entity Chaining Features: distribution of discourse entities in the input query
- Structural features: length of the input, number of predications over the same entity ...
Results: Hypertagging Accuracy

![Graph showing hypertagging accuracy for trees and syntactic classes across One, Five, and Ten samples.](image)

**Claire Gardent**

**Making Choices**

Statistical Microplanning
Results: Coverage

![Bar Chart for Coverage](image)

**Coverage**

- **Trees**
- **Syntactic Classes**

- **Coverage**
- **Coverage-FLS**

- **One**
- **Five**
- **Ten**
Results: Speed

- **Trees**
- **Syntactic Classes**

<table>
<thead>
<tr>
<th>Trees</th>
<th>Syntactic Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>1ms</td>
</tr>
<tr>
<td>Five</td>
<td>1,000ms</td>
</tr>
<tr>
<td>Ten</td>
<td>1,500ms</td>
</tr>
</tbody>
</table>

*Time (in ms)*
Results: Output quality

Human Evaluation

- 48 input queries
- from 13 knowledge bases (2 not used in training corpus)
- 24 raters
- Online evaluation
- Sliding ruler
- Scale 0-50
- Latin Square design
Results: Output quality

- **Clarity**
  - Template: [Score]
  - Hybrid: [Score]

- **Fluency**
  - Template: [Score]
  - Hybrid: [Score]

---

Claire Gardent | Making Choices | Statistical Microplanning
Results: Output quality (BLEU Scores)

![Graph showing BLEU scores for different approaches]

- **Templates**: Blue bars
- **Symbolic**: Green bars
- **Hybrid**: Red bars

**ALL**

- Templates: 0.2
- Symbolic: 0.4
- Hybrid: 0.8

**Generated**

- Templates: 0.6
- Symbolic: 0.8
- Hybrid: 0.8
Example Output: Sentence Segmentation

3 relations, 4 concepts: 1 sentence

I am looking for a used car whose color should be white, which should be located in a France and whose model should be a toyota 4 runner.
Example Output: Sentence Segmentation

3 relations, 4 concepts: 1 sentence

*I am looking for a used car whose color should be white, which should be located in a France and whose model should be a toyota 4 runner.*

4 relations, 5 concepts: 2 sentences

*I am looking for a new car whose exterior color should be beige and whose body style should be a utility vehicle. The new car should run on a natural gas and should be located in a country.*
Example Output: Sentence Segmentation

3 relations, 4 concepts: 1 sentence
I am looking for a used car whose color should be white, which should be located in a France and whose model should be a toyota 4 runner.

4 relations, 5 concepts: 2 sentences
I am looking for a new car whose exterior color should be beige and whose body style should be a utility vehicle. The new car should run on a natural gas and should be located in a country.

3 relations, 5 concepts: 2 sentences
I am looking for a new car whose body style should be a utility vehicle, an off road. The new car should run on a natural gas and should be located in a country.
Example Output: Syntactic Variation

I am looking for a car dealer located in a country and who should sell a car whose make should be a toyota. The car should run on a fuel and should be equipped with a manual gear transmission system. (Participial)

I am looking for a car dealer who should sell a new car whose model should be a toyota. It should be located in a country. (VP with pronominal subject)

I am looking for a new car, an off road whose body style should be a utility vehicle. The new car should run on a natural gas and should be located in a country. (Coordinated VP)

I am looking for a car produced by a car make. The car make should be the make of a toyota. The car make should be located in a city and should produce a land rover freelander. (Canonical Declarative Sentence)
Example Output: Aggregation

VP Coordination

I am looking for a new car (...) that should run on natural gas and should be located in a country.

N1 (V1 N1 and V2 N2)
Example Output: Aggregation

**VP Coordination**

NewCar (...) □ ∃runOn.NaturalGas □ ∃locatedInCountry.Country

I am looking for a new car (...). This new car (should run on natural gas and should be located in a country)_{VP}.

**Relative Clause Coordination**

CommunicationDevice □ ∃assistsWith.Understanding
□ ∃assistsWith.Hearing Disability

I am looking for a communication device (which should assist with a understanding and which should assist with a hearing disability)_{RelCl}. 
Example Output: Aggregation

**NP Coordination**

CarDealer ⊓ ∃_sell_.CrashCar ⊓ ∃_sell_.NewCar

I am looking for a car dealer who should sell (a crash car and a new car)_{NP}. 

Claire Gardent  
**Making Choices** Statistical Microplanning
Example Output: Aggregation

NP Coordination

CarDealer ⊓ ∃sell.CrashCar ⊓ ∃sell.NewCar

I am looking for a car dealer who should sell (a crash car and a new car)_{NP}.

N-Ary NP Coordination

Car ⊓ ∃equippedWith.ManualGearTransmission


∪ ∃equippedWith.AirBagSystem

I am looking for a car equipped with (a manual gear transmission system, an alarm system, a navigation system and an air bag system)_{NP}.
Summary

- Generating from RDF Data (DBPedia, Robot tour)
- Lexicalisation (multi-triple relations, Domain-Range)
- N-ary relations
- Discourse
Summary

- Generating from RDF Data (DBPedia, Robot tour)
- Lexicalisation (multi-triple relations, Domain-Range)
- N-ary relations
- Discourse

THANKS!
WebNLG is looking for a Postdoc/Research Assistant/Engineer

- Machine Learning, Deep learning
- Natural Language Generation