Generating Natural Language from OWL and RDF Data

Grammar-Based, Statistical and Neural Approaches

Claire Gardent

CNRS/LORIA and Université de Lorraine, Nancy

CNL, August 2018
Maynooth, Ireland

Joint Work with

(a) Bikash Gyawali

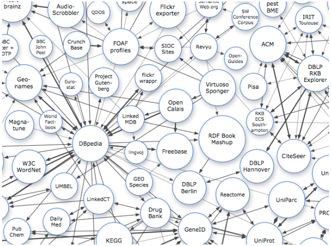
(b) Shashi Narayan

(c) Laura Perez-Beltrachini

(d) Anastasia Shimorina

Funded by the French ANR Project WebNLG
http://webnlg.loria.fr/pages/index.html

Much information is stored in KB and RDF stores

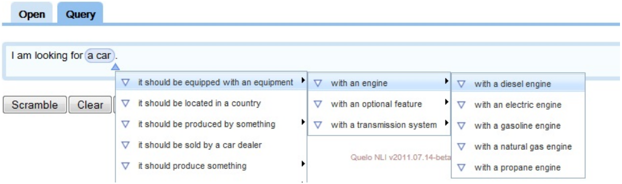


User Survey: 72% of Internet users find it frustrating to get irrelevant information when web searching.
Source: www.internetsociety.org/survey

Natural Language Generation makes this data accessible

QUERYING

Quelo: NLG allows the user to query a Knowledge Base in English



Natural Language Generation makes this data accessible

SUMMARISING

Miakt: NLG generates a patient report from an RDF data store.

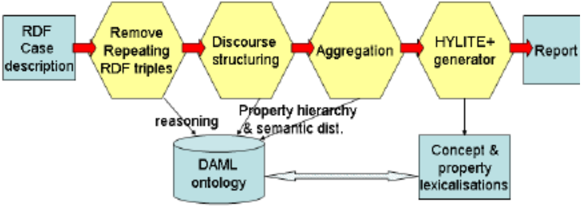


Fig. 1. The MIAKT Generator

Natural Language Generation makes this data accessible

VERBALISING

SWAT: NLG translates the content of an OWL Knowledge Base into English

Class label	OWL axioms (Manchester syntax)	Natural Language Definition Extracted
22rv1	bearer_of some 'prostate carcinoma' derives_from some 'Homo sapiens' derives_from some prostate	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.
HeLa	bearer_of some 'cervical carcinoma' derives_from some 'Homo sapiens' derives_from some cervix derives_from some 'epithelial cell'	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.
Ara-C-resistant murine leukemia	has subclass b117h* has subclass b140h*	A ara c resistant murine leukemia is a cell line. A b117h, and a b140h are kinds of ara c resistant murine leukemias.
GM18507	derives_from some 'Homo sapiens' derives_from some lymphoblast has_quality some male	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.

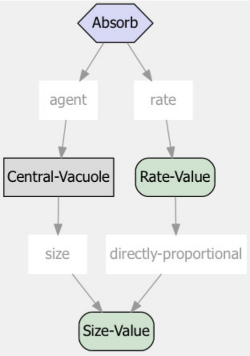
Outline

1. Using NLG to Verbalise a Knowledge Base
   - The KBGen Challenge
   - Grammar-Based NLG
2. Using NLG to query a KB
   - The Quelo NL Interface
   - Statistical NLG
3. Using NLG to verbalise RDF Data
   - The WebNLG Challenge
   - Neural NLG

Verbalising a Knowledge Base

KBGen 2012: an international shared task

Given a set of relations selected from the AURA knowledge base, generate a sentence that is grammatical and fluent in English.



The rate of absorption of a central vacuole is directly proportional to the size of the vacuole.

The KBGen Shared Task

Small Training Corpus: 207 training instances (data/text pairs)

3 Participants:

- UDEL: Hand Written Rule Based System (U. Delaware)
- IMS: Statistical System using a probabilistic grammar induced from the training data (U. Stuttgart)
- LOR-KBGEN: Grammar induced from the training data (Lorraine U.)

LOR-KBGen: A Grammar-Based Approach

A Tree Adjoining Grammar (TAG) is automatically induced from the training corpus

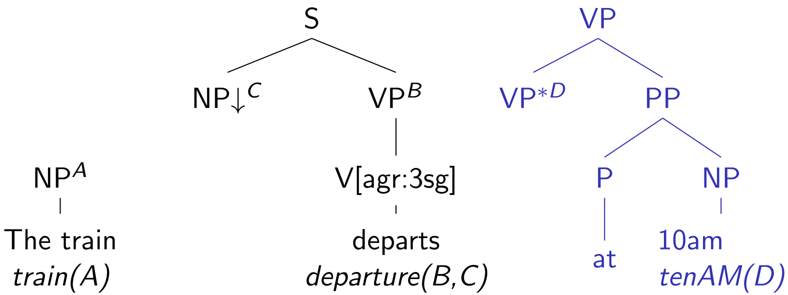
Each grammar rule

- captures a semantically coherent unit
  Semantic Principle
- groups syntactic functors with their dependents
  Extended Domain of Locality

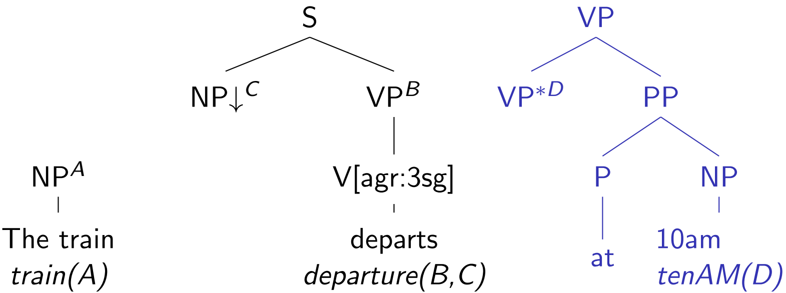
B. Gyawali and C. Gardent
Surface Realisation from Knowledge-Bases.
ACL 2014. Baltimore, USA.

Grammar-Based Generation

Grammar-Based Generation

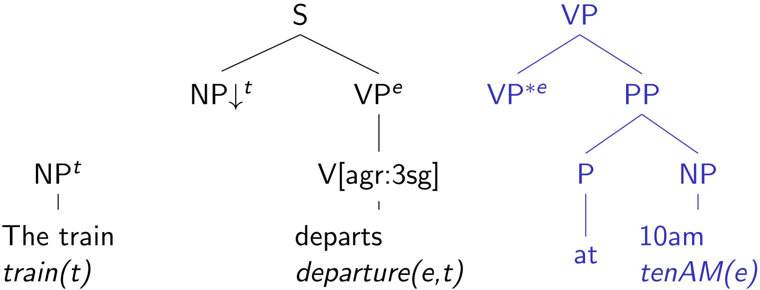


Grammar-Based Generation



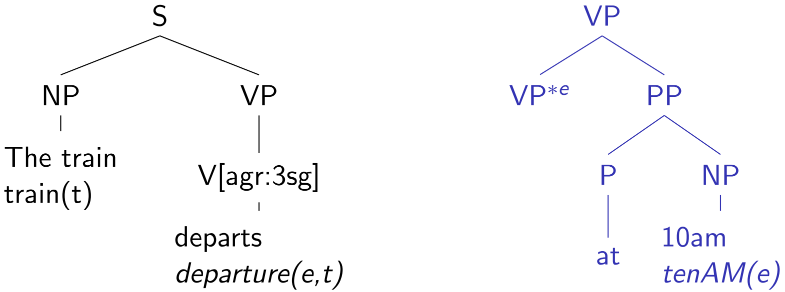
`Input data:` *train(t), departure(e,t), tenAM(e)*

Grammar-Based Generation

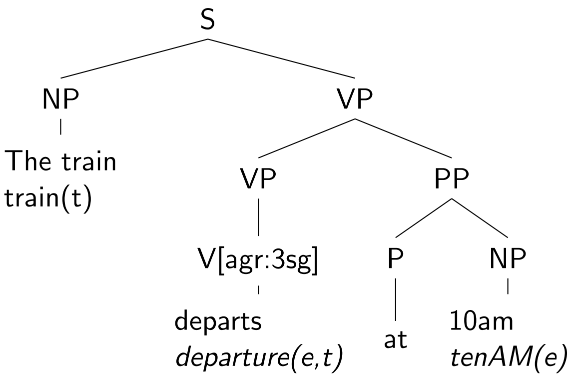


`Input data:` *train(t), departure(e,t), tenAM(e)*

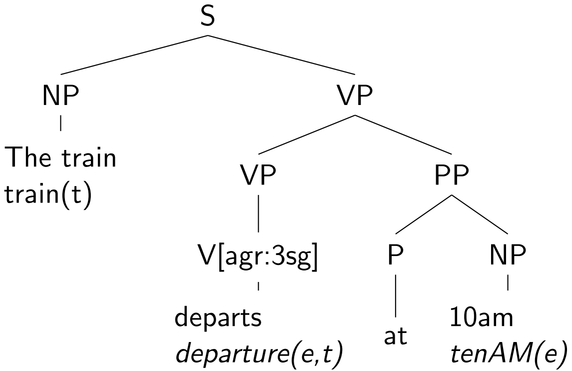
Grammar-Based Generation



Grammar-Based Generation



Grammar-Based Generation



`The train departs at 10am`

Separating Grammar from Lexicon

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

Separating Grammar from Lexicon

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

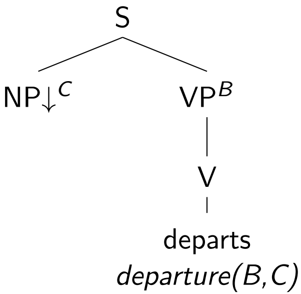
- abstract over lexical items in the grammar

Separating Grammar from Lexicon

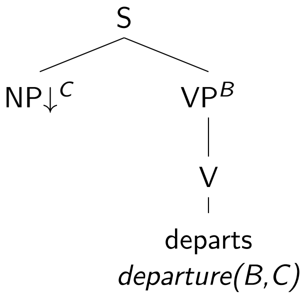
Since each tree is lexicalised, the resulting grammar can be very large. In practice, we therefore

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

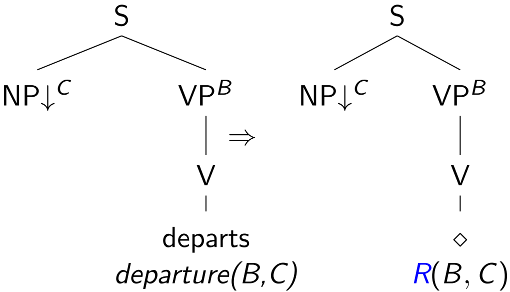
Separating Grammar from Lexicon



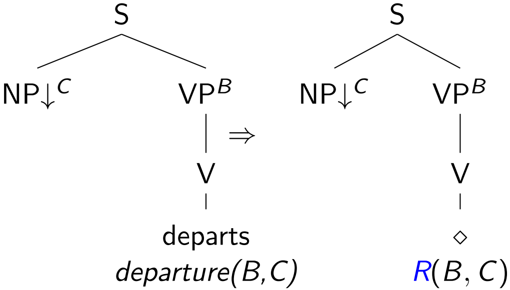
Separating Grammar from Lexicon



Separating Grammar from Lexicon



Separating Grammar from Lexicon



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

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

...

Inducing a Grammar from the KBGen Data

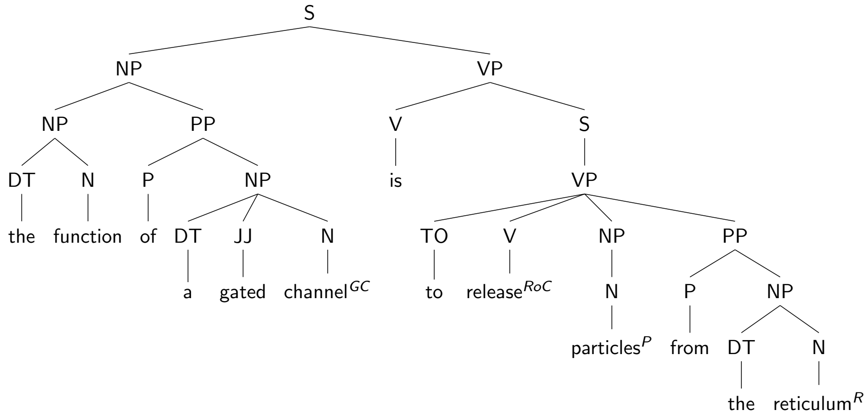
For each (data,sentence) pair in the input:

- Parse and Align semantic variables with words
- Project variables up the parse tree
- Extract subtrees (create a grammar)
- Split trees (generalise)

Example KBGen Input

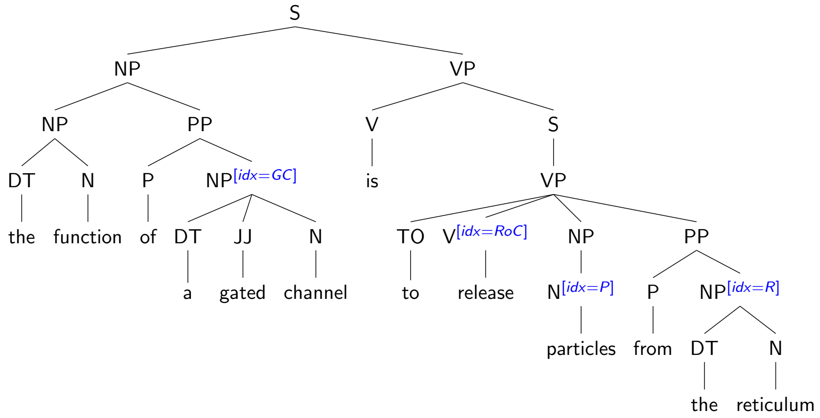
Data	Release-Of-Calcium(RoC) Gated-Channel(GC) Particle-In-Motion(PM) Endoplasmic-Reticulum(ER) agent(RoC, GC) object(RoC, PM) base(RoC, ER) has-function(GC, RoC)
Sentence	*The function of a gated channel is to release particles from the endoplasmic reticulum*

Step 1: Parsing and Variable/Word Alignement



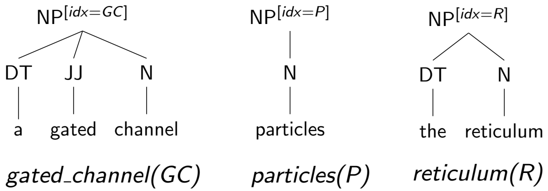
gated_channel(GC)
release_of_calcium(RoC)
particles(P)
reticulum(R)

Step 2: Variable Projection

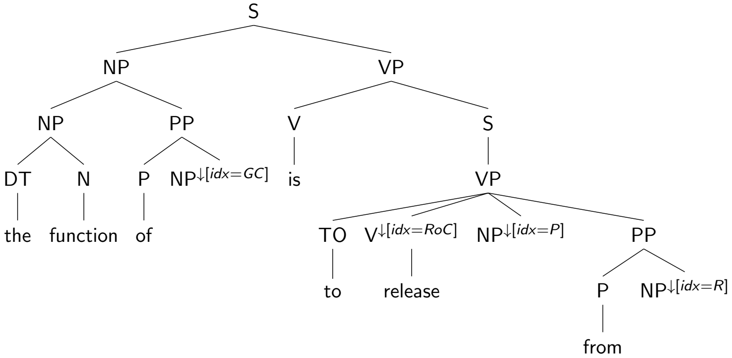


gated_channel(GC)
release_of_calcium(RoC)
particles(P)
reticulum(R)

Step 3: Tree Extraction (Entities))



Step 3: Tree Extraction (Events)



Release_Of _Calcium(RoC)

object(RoC,P)

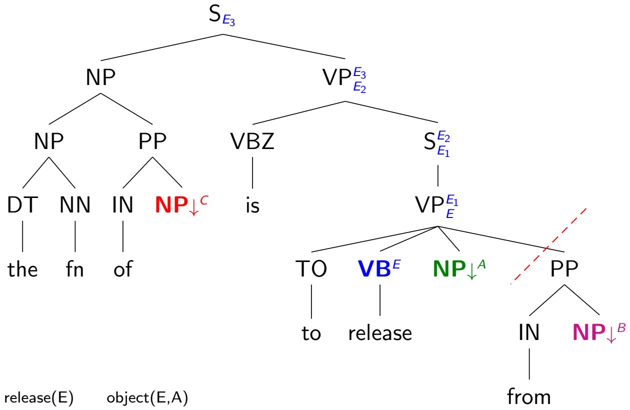
base(RoC,R)

agent(RoC,GC)

has_function(GC,RoC)

Step 4: Grammar Expansion

We further extract from each Event tree, subtrees corresponding to Subject-Verb-Object structure and optional modifiers.

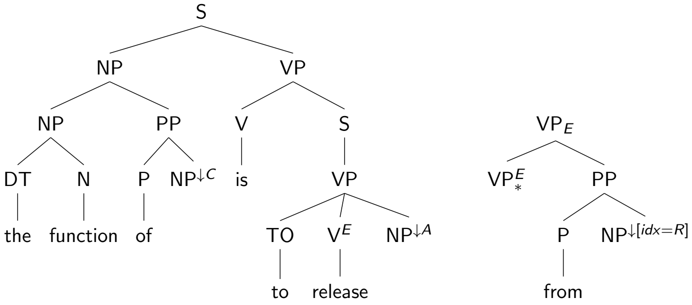


release(E) object(E,A)

agent(E,C) has-function(C,E)

$\boxed{\texttt{base(E,B)}}$

Step 4: Splitting Trees

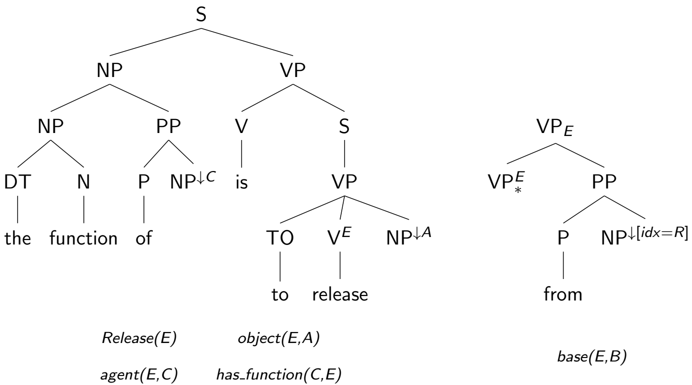


Release(E) *object(E,A)*

agent(E,C) *has_function(C,E)*

base(E,B)

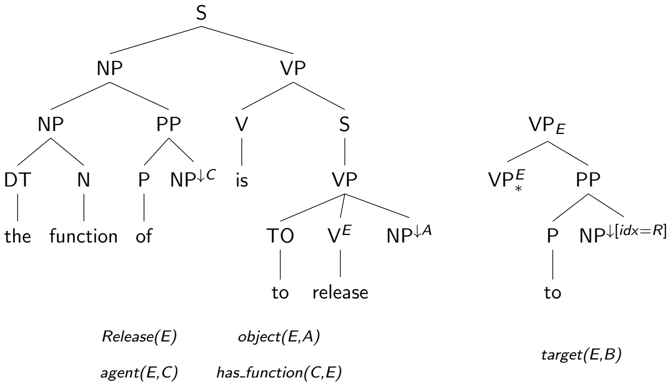
Step 4: Splitting Trees



The function of C is to release A from B

The function of C is to release A

Step 4: Splitting Trees



The function of C is to release A from B

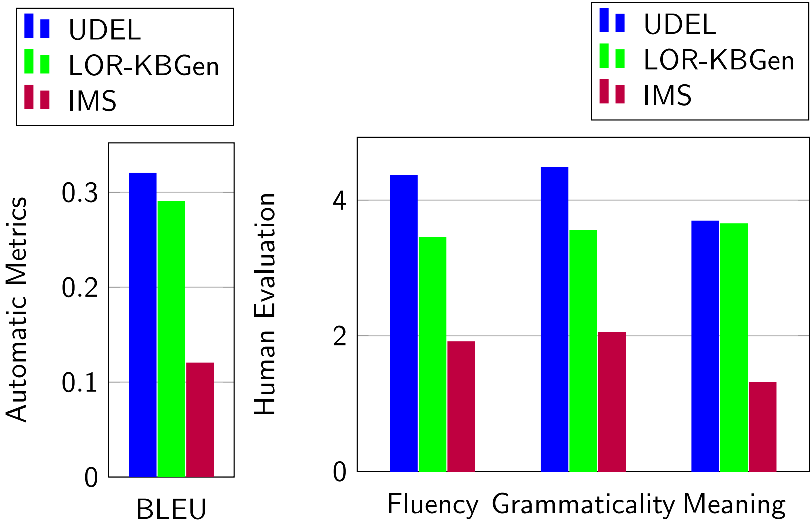
The function of C is to release A

The function of C is to release A to B

Evaluation and Results

- 72 inputs from KBGEN
- Automatic Evaluation: BLEU
- Human-Based Evaluation
  - 12 participants were asked to rate sentences along three dimensions:
    - **fluency**: Is the text easy to read?
    - **grammaticality**: Is the text grammatical ?
    - **adequacy**: Does the meaning conveyed by the generated sentence correspond to the meaning conveyed by the reference sentence?
  - Online evaluation (LG-Eval toolkit)
  - Subjects used a sliding scale (1 to 5)
  - Latin Square Experimental Design was used to ensure that each evaluator sees the same number of output from each system and for each test set item.

Results


UDEL
LOR-KBGen
IMS
Automatic Metrics
0.3
0.2
0.1
0
BLEU
UDEL
LOR-KBGen
IMS
Human Evaluation
4
2
0
Fluency
Grammaticality
Meaning

Conclusion

Linguistically guided grammar induction:

- permits a fully automated approach (unlike the UDEL system)
- yields output sentences whose quality is close to those produced by a hand written system (unlike the IMS system)

Using NLG to query a KB

Interactive refinement of the user query

- Possible (consistent with KB) extensions of the current user query are computed by an automated reasoner ⇒ Set of DL formulae (F)
- Each formal extension ($f \in F$) is then verbalised using NLG
- N.B. The user may revise (substitute, delete, add) the current query

L. Perez-Beltrachini and C. Gardent
Incremental Query Generation
EACL 2014. Gothenburg, Sweden, April 2014.

C. Gardent and L. Perez-Beltrachini
A Statistical, Grammar-Based Approach to Micro-Planning
Computational Linguistics, 43:1, March 2017.

A Statistical Grammar-Based Approach

Input = KB Query

$\texttt{Professor} \sqcap \texttt{Researcher} \sqcap \exists\texttt{teach.LogicCourse}$
$\sqcap \exists\texttt{worksAt.AlicanteUniversity}$

I am looking for a professor who is a researcher and teaches a course on logic. He should work for Alicante University.

Microplanning Task: Segment, lexicalise, aggregate and realise

A Statistical Grammar-Based Approach

The grammar

- Enforces grammaticality
- Accounts for language variability (paraphrasing)

The Statistical Module (Hypertagger)

- Enforces microplanning choices (fluency)
- Enhances efficiency (speed)

The Generation Algorithm

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

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)

Step 1: Tokenize and PoS Tag

`runsOn` → runs/VBD on/IN

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

runs/VBD on/IN →

Semantics	*runsOn*
Tree	nx0Vpnx1
Anchor	should run
Co-Anchor	P → on

Generic Grammar

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

Valency/Subcategorisation Variations

NP_0 should generate NP_1	nx0VVnx1	Canonical
NP_0 should run on NP_1	nx0VVpnx1	Canonical
NP_0 should be equipped with NP_1	nx0VVVpnx1	Canonical
NP_0 should be the equipment of NP_1	nx0VVDNpnx1	Canonical
NP_0 should have access to NP_1	nx0VVNpnx1	Canonical
NP_0 should be relevant to NP_1	nx0VVApnx1	Canonical
NP_0 should be an N_1 product	nx0VVDNnx1	Canonical
NP_0 with NP_1	betanx0Pnx1	Canonical

Generic Grammar

Syntactic Variations

NP_0 should be equipped with NP_1	Canonical
and NP_0 should be equipped with NP_1	S-Coordination
NP_0 which should be equipped with NP_1	SubjRel
NP_0 (...) and which should be equipped with NP_1	SubjRelPU
NP_0 (...), which should be equipped with NP_1	SubjRelPU
NP_0 equipped with NP_1	PpartOrGerund
NP_0 (...) and equipped with NP_1	SharedSubj
NP_0 (...), equipped with NP_1	SharedSubj
NP_1 with which NP_0 should be equipped	PObjRel
NP_0 (equipped with X) and with NP_1	Ellipsis
NP_0 (equipped with X), with NP_1	Ellipsis

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

Accounting for Syntactic Variations (Lexical Selection)

For a given KB symbol, the grammar models multiple syntactic realisations of that symbol

CarDealer(X)	locatedIn(X,Y)	City(Y)	sell(Y,Z)	Car(Z)	runOn(Z,W)	Diesel
nx	nx0VVVpnx1	nx	nx0VVVnx1	nx	nx0VVpnx1	nx
	PRO0VVVpnx1		PRO0VVVnx1		...	
	sCONJnx0VVVpnx1		sCONJnx0VVVnx1		...	
	sCONJPRO0VVVpnx1		sCONJPRO0VVVnx1		...	
	W0nx0VVVpnx1		W0nx0VVVnx1		...	
	ANDWHnx0VVVpnx1		ANDWHnx0VVVnx1		...	
	COMMAWHnx0VVVpnx1		COMMAWHnx0VVVnx1		...	
	betanx0VPpnx1		betanx0VPpnx1		...	
	betanx0ANDVPpnx1		betanx0ANDVPpnx1		...	
	betanx0COMMAVPpnx1		betanx0COMMAVPpnx1		...	
	W1pnx1nx0VV		W1pnx1nx0VV		...	
	betavx0ANDVVVpnx1		betavx0ANDVVVnx1		...	
	betavx0COMMAVVVpnx1		betavx0COMMAVVVnx1		...	

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

Making Choices

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

$\texttt{CarDealer} \sqcap \exists\texttt{locatedIn.City} \sqcap \exists\texttt{sell.Car} \sqcap \exists\texttt{runOn.Diesel}$

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.

~~Tnx0VPpnx1 Tnx0VVnx1 Tnx0VVpnx1~~
~~*I am looking for a car dealer. The car dealer should be located in a city. The car dealer should sell a car. The car should run on diesel.*~~

Making Choices (Hypertagging)

CRF Hypertagging Model

We learn a linear-chain CRF model to predict the mapping between observed input features and hidden syntactic labels $y=\{y_1, \ldots, y_L\}$.

$$P(y \mid x) = \frac{1}{Z(x)} \prod_{l=1}^{L} exp \sum_{k=1}^{K} \theta_k \Phi_k(y_{l-1}, x) \tag{1}$$

The hypertagger finds the optimal hypertag sequence y^* for a given input semantics x:

$$y^* = argmax_y P(y \mid x)$$

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 ...

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.

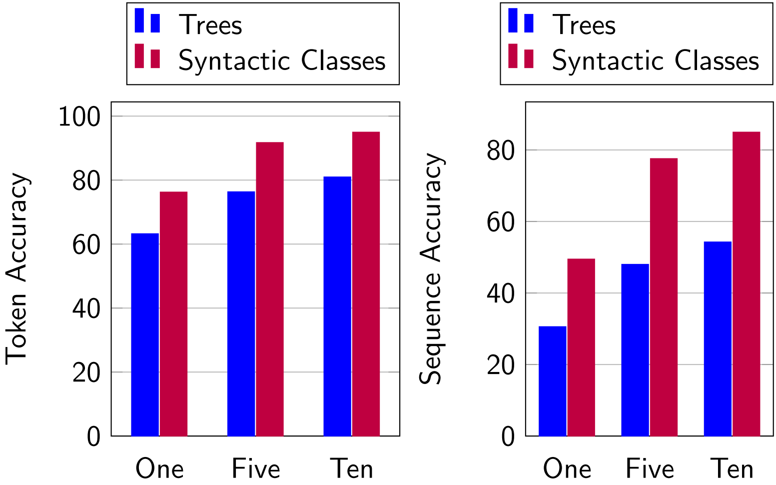
Evaluation Metrics

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

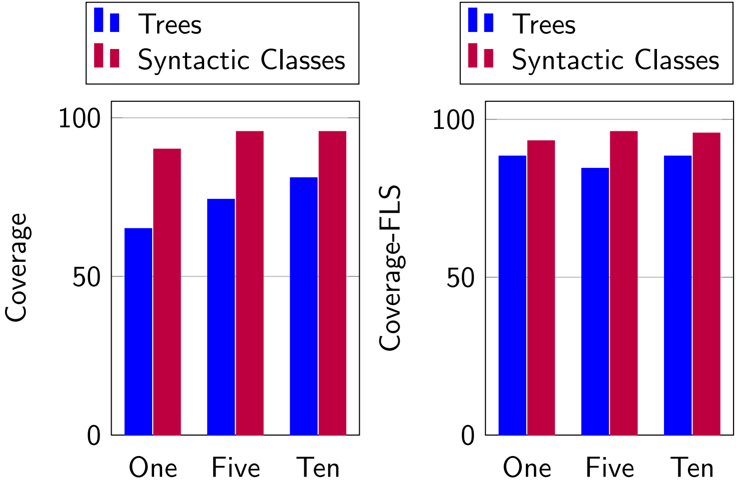
Comparison Models

- Template-Based Model
- Symbolic Grammar-Based Model

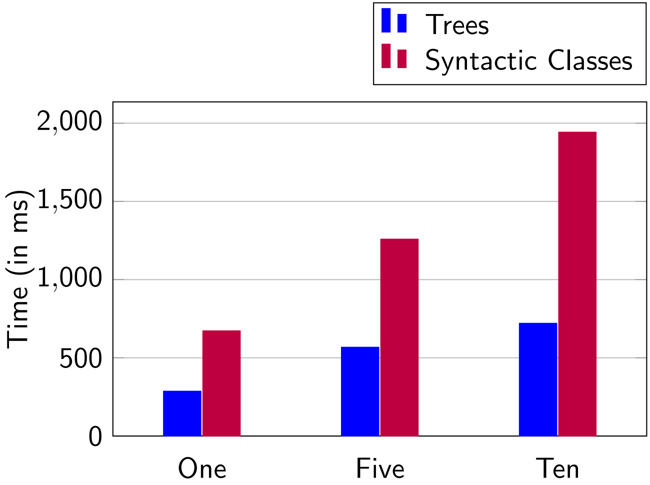
Results: Hypertagging Accuracy



Results: Coverage



Results: Speed

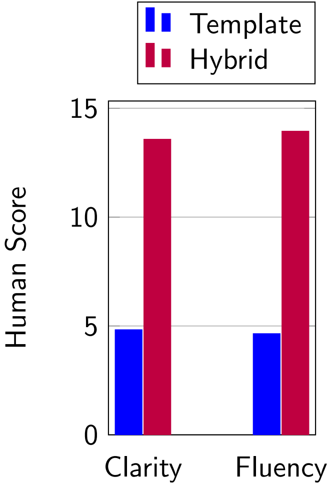


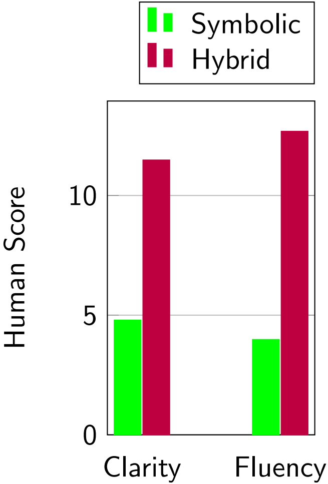
Results: Output quality

Human Evaluation

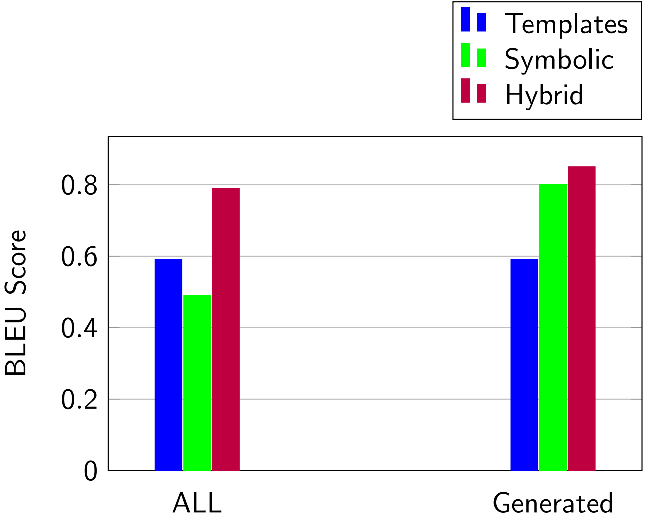
- 48 input queries
- from 13 knowledge bases (2 not used in training corpus)
- 24 raters
- Online evaluation
- Sliding ruler
- Scale 0-50
- Latin Square design

Results: Output quality





Results: Output quality (BLEU Scores)



Example Ouput: 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 sells a car whose model is a toyota. **It should be located in a country***.* *(Sentence 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

NewCar (...) $\sqcap$ $\exists$runOn.NaturalGas $\sqcap$ $\exists$locatedInCountry.Country

I am looking for a new car (...). This new car (should run on natural gas and should be located in a country)$_{VP}$. *N1 (V1 N1 and V2 N2)*

Relative Clause Coordination

CommunicationDevice $\sqcap$ $\exists$assistsWith.Understanding

$\sqcap$ $\exists$assistsWith.HearingDisability

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 $\sqcap$ $\exists$sell.CrashCar $\sqcap$ $\exists$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 $\sqcap$ $\exists$equippedWith.ManualGearTransmission
$\sqcap$ $\exists$equippedWith.AlarmSystem $\sqcap$ $\exists$equippedWith.NavigationSystem
$\sqcap$ $\exists$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

Ambiguous Grammar = High Expressivity, Large Search Space

Hypertagging = Making Choices

1. Using NLG to Verbalise a Knowledge Base
   - The KBGen Challenge
   - Grammar-Based NLG
2. Using NLG to query a KB
   - The Quelo NL Interface
   - Statistical NLG
3. Using NLG to verbalise RDF Data
   - The WebNLG Challenge
   - Neural NLG

WebNLG: Goals

[NLG]
Provide a benchmark on which to train, evaluate and compare microplanners for data-to-text generation.

[Semantic Web]
Train, evaluate and compare verbalisers for RDF Data

WebNLG: A Microplanning Task

Data ⇒ Text

```
(John_E_Blaha birthDate 1942_08_26)
(John_E_Blaha birthPlace San_Antonio)
(John_E_Blaha occupation Fighter_pilot)
```

John E Blaha, born in San Antonio on 1942-08-26, worked as a fighter pilot

- Generating Referring Expressions: Describing entities
- Lexicalisation: Choosing lexical items
- Surface Realisation: Choosing syntactic structures
- Aggregation: Avoiding repetition
- Sentence segmentation: Segmenting the content into sentence size chunks

Creating the WebNLG Dataset

- RDF KB (DBPedia) – $\boxed{\text{Content Selection}}$ → Data
- Text produced by crowdworkers

	WebNLG
# data-text pairs	40,049
# distinct inputs	15,095
# DBPedia Categories	15

Laura Perez-Beltrachini, Rania Mohammed Sayed and Claire Gardent
Building RDF Content for Data-to-Text Generation
COLING, 2016.

Claire Gardent, Anastasia Shimorina, Shashi Narayan and Laura Perez-Beltrachini
Creating Training Corpora for NLG Micro-Planning
ACL, 2017.

Training and Testing Data

- Train/Dev/Test split: 80/10/10
- 10 seen categories: Astronaut, University, Monument, Building, ComicsCharacter, Food, Airport, SportsTeam, City and WrittenWork
- 5 unseen categories: Athlete, Artist, MeanOfTransportation, CelestialBody, Politician

	Train+Dev	Test Seen	Test Unseen	All
Entries	7,812	971	891	9,674
Data/text pairs	20,370	2,495	2,433	25,298

The Participants

61 downloads, 6 participants, 8 systems

3 Pipeline Systems
TILB-PIPELINE, UIT-VNU and UPF-FORGE

1 SMT-Based System
TILB-SMT

5 Neural-Based Systems
ADAPT, MELBOURNE, PKUWRITER, TILB-NMT and BASELINE

Pipeline Systems

	Order Triples	Aggr.	Templ.	REG	Gr.	re-ranking
TILB-PIPELINE	+	-	Induced	+	-	+
UIT-VNU	-	-	Induced	-	-	-
UPF-FORGE	+	+	Manual	-	+	-

Seq2seq Systems

	Pre-processing	Word Repr	Add. Module
TILB-NMT	Delex		REG Module
PKUWRITER			Rerank
MELBOURNE	Delex and Sem Typing	Glove vectors	
ADAPT	Tokenize RDF	Subwords	

Global Results

BLEU	
Melbourne	45.13
TiLB-SMT	44.28
PKUWriter	39.88
UPF-FORGe	38.65
TiLB-Pipeline	35.29
TiLB-NMT	34.60
Baseline	33.24
ADAPT	31.06
UIT-VNU	7.07

METEOR	
UPF-FORGe	0.39
TiLB-SMT	0.38
Melbourne	0.37
TiLB-NMT	0.34
ADAPT	0.31
PKUWriter	0.31
TiLB-Pipeline	0.30
Baseline	0.23
UIT-VNU	0.09

TER	
Melbourne	0.47
TiLB-SMT	0.53
PKUWriter	0.55
UPF-FORGe	0.55
TiLB-Pipeline	0.56
TiLB-NMT	0.60
Baseline	0.61
UIT-VNU	0.82
ADAPT	0.84

- 6 systems above the baseline (4 well above it)
- Neural NLG
  - Glove vectors and semantic typing of entities help (Melbourne)
  - Relexicalisation works better than subwords (ADAPT)

Results for Seen Categories

BLEU	
ADAPT	60.59
MELBOURNE	54.52
TILB-SMT	54.29
BASELINE	52.39
PKUWRITER	51.23
TILB-NMT	44.34
TILB-PIPELINE	43.28
UPF-FORGE	40.88
UIT-VNU	19.87

METEOR	
ADAPT	0.44
TILB-SMT	0.42
MELBOURNE	0.41
UPF-FORGE	0.40
TILB-NMT	0.38
TILB-PIPELINE	0.38
PKUWRITER	0.37
BASELINE	0.37
UIT-VNU	0.15

TER	
ADAPT	0.37
MELBOURNE	0.40
BASELINE	0.44
PKUWRITER	0.45
TILB-SMT	0.47
TILB-PIPELINE	0.48
TILB-NMT	0.51
UPF-FORGE	0.55
UIT-VNU	0.78

- Neural and SMT systems are better are “reproducing” seen data
- Rule based systems (UPF-FORGE, TILB-PIPELINE) seems to produce text that is more different from references than learned systems (higher METEOR and TER)

Results on Unseen Categories

BLEU	
UPF-FORGE	35.70
MELBOURNE	33.27
TILB-SMT	29.88
PKUWRITER	25.36
TILB-NMT	25.12
TILB-PIPELINE	20.65
ADAPT	10.53
BASELINE	06.13
UIT-VNU	0.11

METEOR	
UPF-FORGE	0.37
TILB-SMT	0.33
MELBOURNE	0.33
TILB-NMT	0.31
PKUWRITER	0.24
TILB-PIPELINE	0.21
ADAPT	0.19
BASELINE	0.07
UIT-VNU	0.03

TER	
UPF-FORGE	0.55
MELBOURNE	0.55
TILB-SMT	0.61
TILB-PIPELINE	0.65
PKUWRITER	0.67
TILB-NMT	0.72
BASELINE	0.80
UIT-VNU	0.87
ADAPT	1.4

- UPF-FORGE performs well on unseen data and much better than most neural systems

And also

NLG

- Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang and Wei Wang
  GTR-LSTM: A Triple Encoder for Sentence Generation from RDF Data.
  ACL, 2018.
- Emiel Krahmer, Thiago Castro Ferreira, Sander Wubben, Ákos Kádár and Diego Moussallem
  NeuralREG: An end-to-end approach to referring expression generation.
  ACL, 2018.
- Emilie Colin and Claire Gardent.
  Generating Syntactic Paraphrases.
  EMNLP, 2018.

Sentence Simplification

- Shashi Narayan, Claire Gardent, Shay Cohen and Anastasia Shimorina
  Split and Rephase
  EMNLP, 2017.
- Roee Aharoni and Yoav Goldberg
  Split and Rephrase: Better Evaluation and a Stronger Baseline
  ACL, 2018.

Relation Extraction

- Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu and Jun Zhao
  Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism
  ACL, 2018.

What next ?

- Better NLG models
- Other text types and communication goals
- Multilingual Generation

THANKS!