Machine Translation

13: Syntax

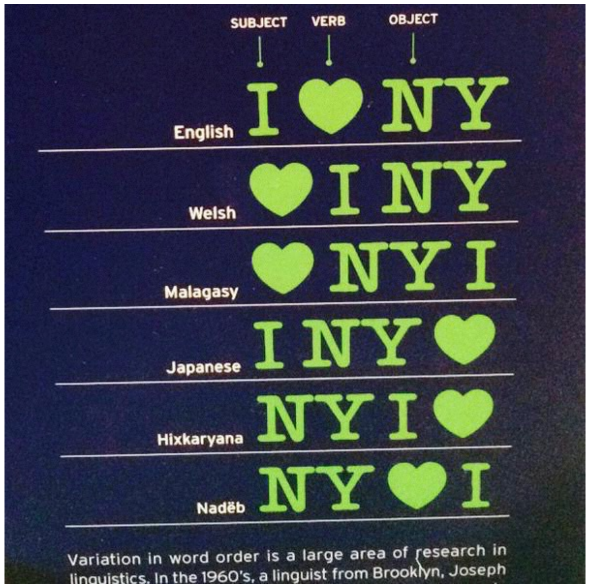
Rico Sennrich

University of Edinburgh

Overview

today

- some syntactic phenomena that make (machine) translation difficult
- some solutions from MT research



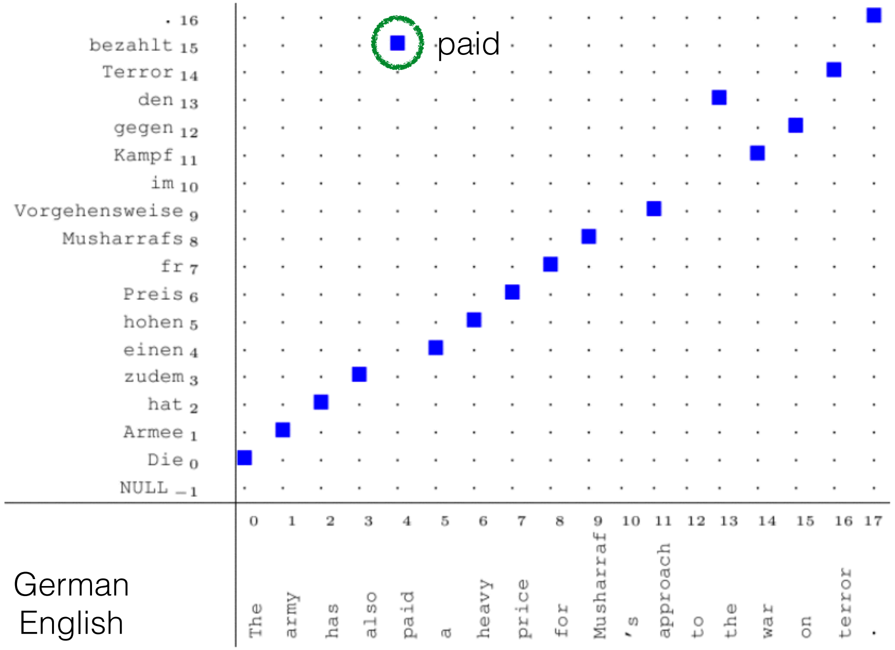
Word Order

- languages differ in word order:
  - SVO: English, French, Mandarin, Russian, ...
  - SOV: Hindi, Latin, Japanese, Korean, ...
  - VSO, VOS, OSV, OSV exist, but less common
  - German is V2 in main clause, SOV in subordinate clause
  - word order more flexible when function is morphologically marked

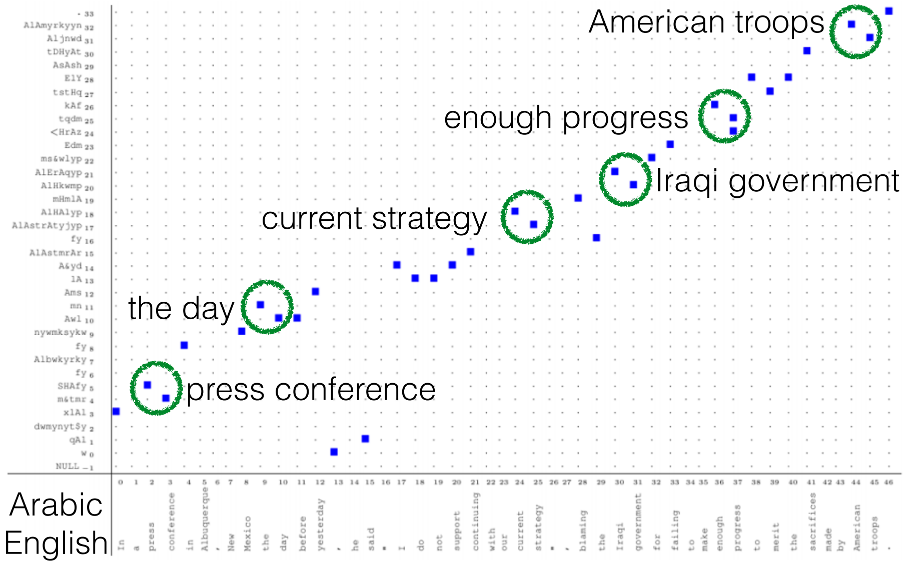
example: German–English

der Mann , der den letzten Marathon gewonnen hat

the man who won the last marathon



Features	*Indo-European* English	German	French	*Afro-Asiatic* Arabic	*Altaic* Turkish	*Japanese* Japanese	*Sino-Tibet.* Chinese
Phrase-level Noun,Adpositions [with] [a stick]	Adp-N	Adp-N	Adp-N	Adp-N	N-Adp	N-Adp	N-Adp/ Adp-N
Noun,Genitive [Tom's] [stick]	N-Gen/ Gen-N	N-Gen	N-Gen	N-Gen	Gen-N	Gen-N	Gen-N
Noun,Adjective [hungry] [Tom]	A-N	A-N	N-A	N-A	A-N	A-N	A-N
Noun,Demonstrative [this] [stick]	Dem-N	Dem-N	Dem-N	Dem-N	Dem-N	Dem-N	Dem-N
Noun,Numeral [two] [sticks]	Num-N	Num-N	Num-N	Num-N	Num-N	Num-N	Num-N
Adjective,DegreeW. [very] [hungry]	Deg-A	Deg-A	Deg-A	A-Deg	Deg-A	Deg-A	Deg-A
Feature	**English**	**German**	**French**	**Arabic**	**Turkish**	**Japanese**	**Chinese**



Word Order

translation units can be discontinuous

example: German separable verb prefixes are clause-final

he **proposes** a trade
er **schlägt** einen Handel **vor**

Word Order

"The Awful German Language" by Mark Twain

The Germans have another kind of parenthesis, which they make by splitting a verb in two and putting half of it at the beginning of an exciting chapter and the other half at the end of it. Can any one conceive of anything more confusing than that? These things are called "separable verbs." The German grammar is blistered all over with separable verbs; and the wider the two portions of one of them are spread apart, the better the author of the crime is pleased with his performance.

Word Order

source side pre-reordering

- preprocess the source text to better match target language word order
- standard for some language pairs for phrase-based SMT
- various approaches based on syntactic analysis of source sentence:
  - hand-written rules [Nießen and Ney, 2000, Collins et al., 2005]
  - automatically learned rules [Xia and McCord, 2004, Genzel, 2010]
  - neural pre-reordering [Miceli Barone and Attardi, 2015]
- negative results for neural MT [Du and Way, 2017]

Word Order

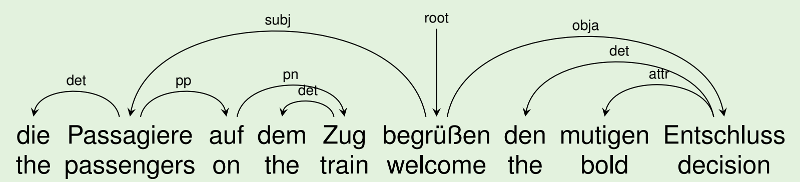
target side pre-reordering?

- in principle, we can reorder target side before training
- need second step to restore original target language word order
  $\rightarrow$ this is hard
- some research, but never became standard

Syntactic N-grams

- n-grams may not be meaningful if word order is flexible
- syntactic n-grams for evaluation: head-word-chain metric (HWCM) [Liu and Gildea, 2005]

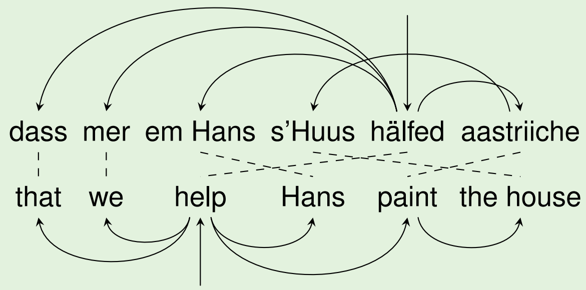
syntactic n-grams



<s> begrüßen	begrüßen Passagiere	Passagiere die
Passagiere auf	auf Zug	Zug dem
begrüßen Entschluss	Entschluss den	Entschluss mutigen

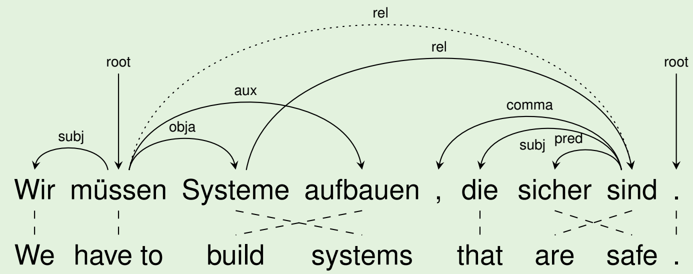
Non-projective Structures

classical example of context-sensitive structures in Swiss German (Zurich dialect)



Non-projective Structures

non-projective German dependency tree



- most syntax-based SMT systems are context-free
- either they can't produce non-projective structures, or we use pseudo-projective arcs (dotted line)

Subcategorization

- words occur with specific syntactic arguments
- complex mapping between syntactic arguments and meaning

example

- *remember* can have direct object or clausal object
- semantic role: content of the memory
  - *he remembers his medical appointment.*
  - *he remembers that he has a medical appointment.*
- *remind* can have direct object, and prep. or clausal object
  - direct object: recipient of information
  - prep. or clausal object: information
  - *I remind him of his medical appointment.*
  - *I remind him that he has a medical appointment.*
- ungrammatical (or semantically nonsensical):
  - *he remembers of his medical appointment.
  - *he reminds his medical appointment.

Subcategorization

subcategorization rules often differ between languages

- *he remembers the medical appointment.*
- **er erinnert den Arzttermin.*

- *er erinnert sich an den Arzttermin.*
- **he remembers himself to the medical appointment.*

for some translations, syntactic arguments swap semantic roles

- *he misses the cat*
- *die Katze fehlt ihm*
  (the cat is missing to him)

Subcategorization

different meanings of words occur with different subcategories:

- *she applies for a job.*
  prep. object *for*: submit oneself as a candidate (German: “bewerben”)
- *this rule applies to everyone.*
  intransitive: be relevant (German: “gelten”)
- *he applies the wrong test.*
  transitive: use (German: “anwenden”)

Syntax and Neural MT

attentional encoder–decoder is less limited than previous approaches

- reordering can be learned by attention model
- consequence of subcategorization constraints:
  you should not translate syntactic arguments independently
- recurrent model can handle discontiguous and non-projective structures

recent research

does neural MT benefit from syntactic structure/information?

Linguistic Input Features

guide reordering with syntactic information

source	*Gefährlich$_{pred}$ ist die Route$_{subj}$ aber dennoch.*
reference	*However the route is dangerous.*
baseline NMT	**Dangerous is the route, however.*

Linguistic Input Features

disambiguate words by POS

English	German
close$_{verb}$	schließen
close$_{adj}$	nah
close$_{noun}$	Ende

source	*We thought a win like this might be close$_{adj}$.*
reference	*Wir dachten, dass ein solcher Sieg nah sein könnte.*
baseline NMT	**Wir dachten, ein Sieg wie dieser könnte schließen.*

Neural Machine Translation: Multiple Input Features

Use separate embeddings for each feature, then concatenate
(same method as for inclusion of lemma)

baseline: only word feature

$$E(close) = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.3 \\ 0.1 \end{bmatrix}$$

$|F|$ input features

$$E_1(close) = \begin{bmatrix} 0.4 \\ 0.1 \\ 0.2 \end{bmatrix} \quad E_2(adj) = \begin{bmatrix} 0.1 \end{bmatrix} \quad E_1(close) \parallel E_2(adj) = \begin{bmatrix} 0.4 \\ 0.1 \\ 0.2 \\ 0.1 \end{bmatrix}$$

Experiments

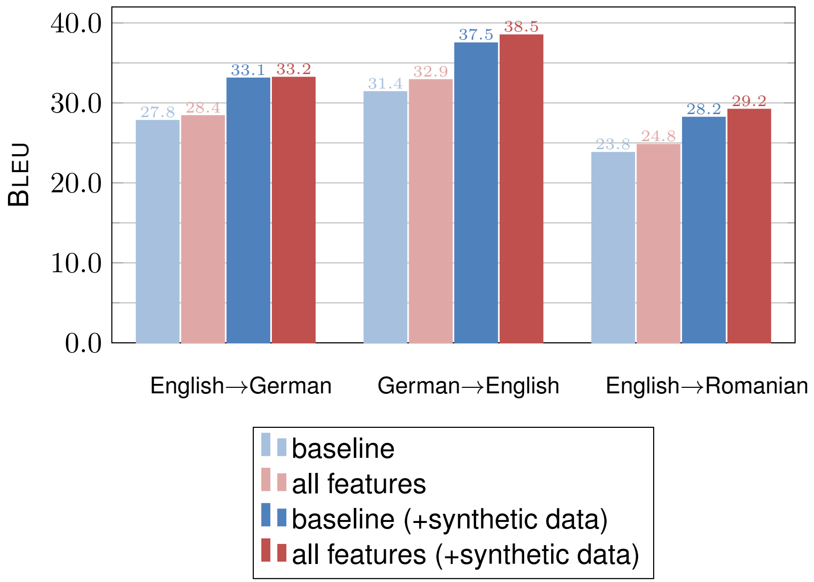
Features

- lemmas
- morphological features
- POS tags
- dependency labels
- BPE tags

Data

- WMT16 training/test data
- English↔German and English→Romanian

Results: BLEU ↑



Tree-Structured Encoders [Eriguchi et al., 2016]

- represent source sentence as binary tree
- leaf nodes: states of sequential RNN
- tree-based encoder computes state of k-th parent node (h_k^p) as function of left and right child nodes (h_k^l and h_k^r):

$$h_k^p = f(h_k^l, h_k^r)$$

- allow attention on original encoder states (leaves) and tree nodes

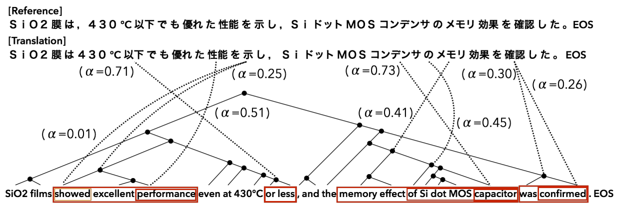


Figure 5: Translation example of a long sentence and the attentional relations by our proposed model.

Recurrent Neural Network Grammars

hypothesis:
- recurrent neural networks have recency bias
- instead, we want to induce syntactic bias

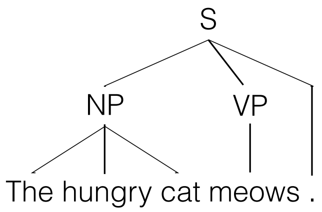
Recurrent Neural Network Grammars

One theory of hierarchy

- Generate symbols sequentially using an RNN
- Add some “control symbols” to rewrite the history periodically
  - Periodically “compress” a sequence into a single “constituent”
  - Augment RNN with an operation to compress recent history into a single vector (-> “reduce”)
  - RNN predicts next symbol based on the history of compressed elements and non-compressed terminals (“shift” or “generate”)
  - RNN must also predict “control symbols” that decide how big constituents are
- We call such models **recurrent neural network grammars**.

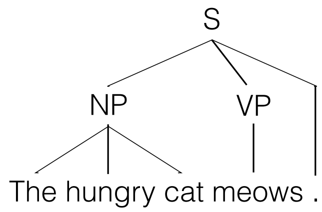
Recurrent Neural Network Grammars

Trees as sequences



Recurrent Neural Network Grammars

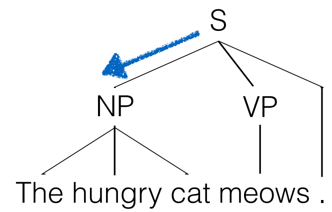
Trees as sequences



S(NP(The hungry cat) VP(meows) .)

Recurrent Neural Network Grammars

Trees as sequences

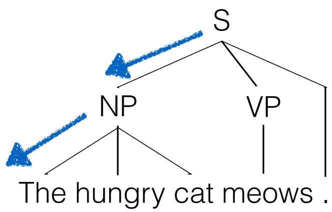


S(NP(The hungry cat) VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

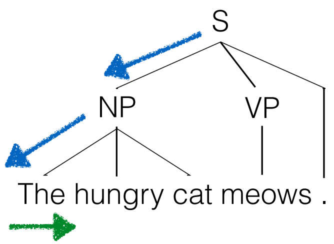


S(**NP(** The hungry cat) VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

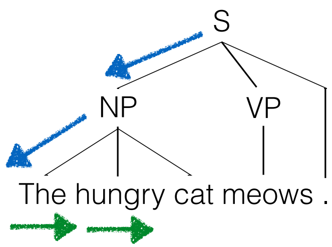


S(NP(**The** hungry cat) VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

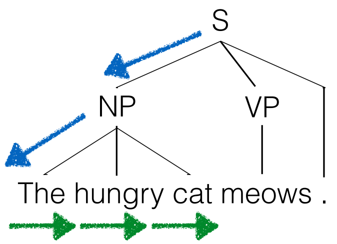


S(NP(The **hungry** cat) VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

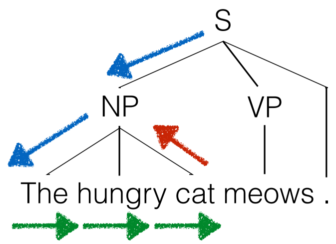


S(NP(The hungry **cat**) VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

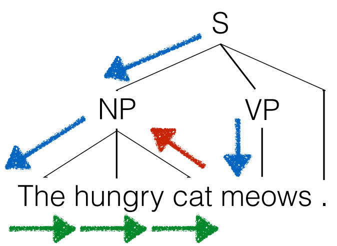


S(NP(The hungry cat **)** VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

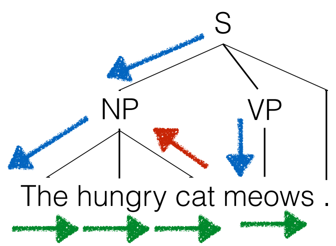


S(NP(The hungry cat) **VP(** meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

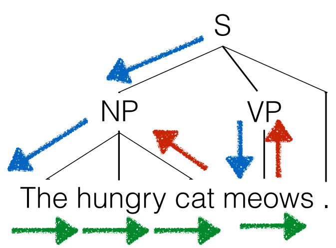


S(NP(The hungry cat) VP(**meows**) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

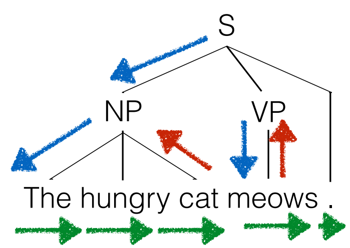


S(NP(The hungry cat) VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences

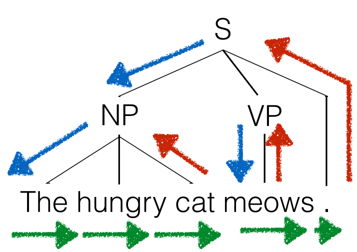


S(NP(The hungry cat) VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Trees as sequences



S(NP(The hungry cat) VP(meows) .)

Tree traversals correspond to stack operations

Recurrent Neural Network Grammars

Terminals	Stack	Action

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)
The hungry cat	(S (NP *The hungry cat*	

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)
The hungry cat	(S (NP *The hungry cat*	**REDUCE**

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)
The hungry cat	(S (NP *The hungry cat*	**REDUCE**
The hungry cat	(S (NP *The hungry cat*)	

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)
The hungry cat	(S (NP *The hungry cat*	**REDUCE**
The hungry cat	(S (NP *The hungry cat*)	
	(S (NP *The hungry cat*)	

Compress “The hungry cat” into a single composite symbol

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)
The hungry cat	(S (NP *The hungry cat*	**REDUCE**
The hungry cat	(S (NP *The hungry cat*)	

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)
The hungry cat	(S (NP *The hungry cat*	**REDUCE**
The hungry cat	(S (NP *The hungry cat*)	**NT**(VP)

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)
The hungry cat	(S (NP *The hungry cat*	**REDUCE**
The hungry cat	(S (NP *The hungry cat*)	**NT**(VP)
The hungry cat	(S (NP *The hungry cat*) (VP	

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	**NT**(NP)
	(S (NP	**GEN**(*The*)
The	(S (NP *The*	**GEN**(*hungry*)
The hungry	(S (NP *The hungry*	**GEN**(*cat*)
The hungry cat	(S (NP *The hungry cat*	**REDUCE**
The hungry cat	(S (NP *The hungry cat*)	**NT**(VP)
The hungry cat	(S (NP *The hungry cat*) (VP	**GEN**(*meows*)

Recurrent Neural Network Grammars

Terminals	Stack	Action
		NT(S)
	(S	NT(NP)
	(S (NP	GEN(*The*)
The	(S (NP *The*	GEN(*hungry*)
The hungry	(S (NP *The hungry*	GEN(*cat*)
The hungry cat	(S (NP *The hungry cat*	REDUCE
The hungry cat	(S (NP *The hungry cat*)	NT(VP)
The hungry cat	(S (NP *The hungry cat*) (VP	GEN(*meows*)
The hungry cat meows	(S (NP *The hungry cat*) (VP *meows*	REDUCE
The hungry cat meows	(S (NP *The hungry cat*) (VP *meows*)	GEN(.)
The hungry cat meows .	(S (NP *The hungry cat*) (VP *meows*) .	REDUCE
The hungry cat meows .	(S (NP *The hungry cat*) (VP *meows*) .)	

Recurrent Neural Network Grammars

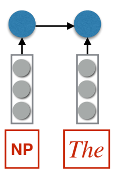
Syntactic Composition

Need representation for: (NP *The hungry cat*)

Recurrent Neural Network Grammars

Syntactic Composition

Need representation for: (NP *The hungry cat*)

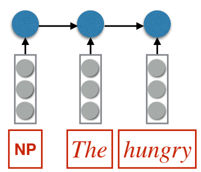


What head type?

Recurrent Neural Network Grammars

Syntactic Composition

Need representation for: (NP *The hungry cat*)

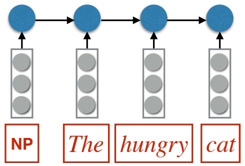


What head type?

Recurrent Neural Network Grammars

Syntactic Composition

Need representation for: (NP *The hungry cat*)

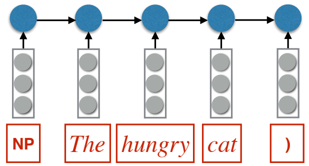


What head type?

Recurrent Neural Network Grammars

Syntactic Composition

Need representation for: (NP *The hungry cat*)

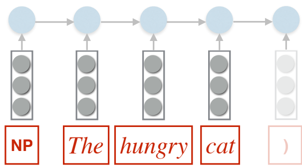


What head type?

Recurrent Neural Network Grammars

Syntactic Composition

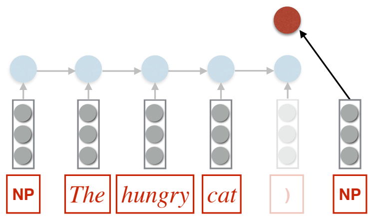
Need representation for: (NP *The hungry cat*)



Recurrent Neural Network Grammars

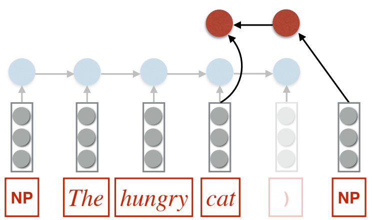
Syntactic Composition

Need representation for: (NP *The hungry cat*)



Syntactic Composition

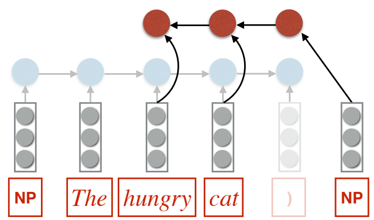
Need representation for: (NP *The hungry cat*)



Recurrent Neural Network Grammars

Syntactic Composition

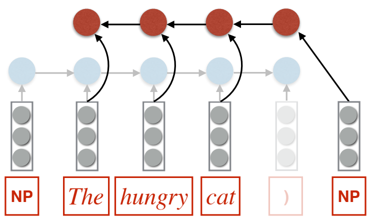
Need representation for: (NP *The hungry cat*)



Recurrent Neural Network Grammars

Syntactic Composition

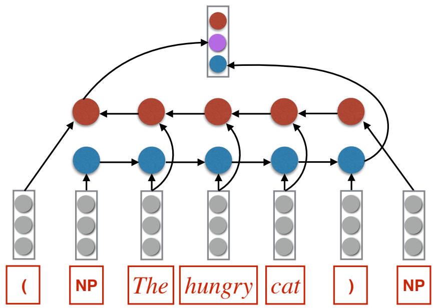
Need representation for: (NP *The hungry cat*)



Recurrent Neural Network Grammars

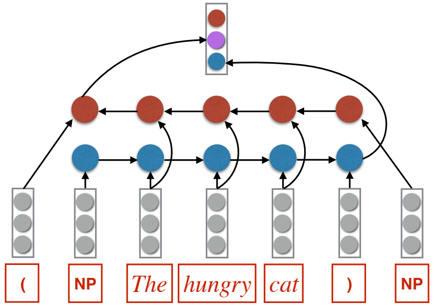
Syntactic Composition

Need representation for: (NP *The hungry cat*)



Recursion

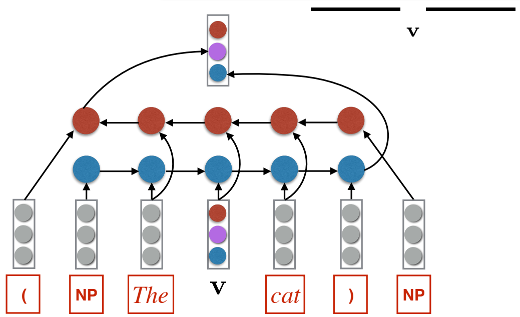
Need representation for: (NP *The hungry cat*)
(NP *The* (ADJP *very hungry*) *cat*)



Recurrent Neural Network Grammars

Recursion

Need representation for: (NP *The hungry cat*)
(NP *The* (ADJP *very hungry*) *cat*)



Recurrent Neural Network Grammars

RNNGs in MT

Eriguchi et al., Feb 2017

- Basic idea: learn decoder-encoder MT model and RNNG on parallel data with parsed target side, sharing target word embedding parameters. (multi-task learning). To translate, just use MT model.

	De-En	Ru-En	Cs-En	Jp-En
BLEU				
NMT	16.61	12.03	11.22	17.88
NMT+RG	16.41	**12.46**†	**12.06**†	**18.84**†
RIBES				
NMT	73.75	69.56	69.59	71.27
NMT+RG	**75.03**†	**71.04**†	**70.39**†	**72.25**†

Table 2: BLEU and RIBES scores by the baseline and proposed models on the test set. We use the bootstrap resampling method from Koehn (2004) to compute the statistical significance. We use † to mark those significant cases with $p < 0.005$.

Jp-En (Dev)	BLEU
NMT+RG	18.60
w/o Buffer	18.02
w/o Action	17.94
w/o Stack	17.58
NMT	17.75

Table 3: Effect of each component in RNNG.

Slide Credit

- slide credit for slides 2,4-6,24-27: Adam Lopez

Bibliography I

Collins, M., Koehn, P., and Kučerová, I. (2005).
Clause Restructuring for Statistical Machine Translation.
In ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 531–540, Ann Arbor, Michigan. Association for Computational Linguistics.

Du, J. and Way, A. (2017).
Pre-Reordering for Neural Machine Translation: Helpful or Harmful?
The Prague Bulletin of Mathematical Linguistic, (108):107–182.

Eriguchi, A., Hashimoto, K., and Tsuruoka, Y. (2016).
Tree-to-Sequence Attentional Neural Machine Translation.
In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 823–833, Berlin, Germany.

Genzel, D. (2010).
Automatically Learning Source-side Reordering Rules for Large Scale Machine Translation.
In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 376–384, Beijing, China. Coling 2010 Organizing Committee.

Liu, D. and Gildea, D. (2005).
Syntactic Features for Evaluation of Machine Translation.
In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 25–32, Ann Arbor, Michigan.

Miceli Barone, A. V. and Attardi, G. (2015).
Non-projective Dependency-based Pre-Reordering with Recurrent Neural Network for Machine Translation.
In Proceedings of the Ninth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 10–20, Denver, Colorado, USA. Association for Computational Linguistics.

Bibliography II

Nießen, S. and Ney, H. (2000).
Improving SMT quality with morpho-syntactic analysis.
In 18th Int. Conf. on Computational Linguistics, pages 1081–1085.

Xia, F. and McCord, M. (2004).
Improving a Statistical MT System with Automatically Learned Rewrite Patterns.
In Proceedings of Coling 2004, pages 508–514, Geneva, Switzerland. COLING.