Supplementary Materials for the ACL Submission: Data-Driven Broad-Coverage Grammars for Opinionated Natural Language Generation (ONLG)

February 7, 2017

A Architecture

In Figure 1 we present a bird's eye view of the data-driven, grammar-based, architecture we propose.

B Grammars

In Figures 2, 3 and 4 we demonstrate the parse tree and generation sequence of the phrase ”The good wife” in our Base representation, Lexicalized representation, and Relational-Realizational representation respectively.

C Obtaining Relational-Realizational Trees

Figure 5 demonstrates the annotating of phrase-structure trees with sentiment, relation labels and lexical heads along with the corresponding dependency graph of the same sentence and the full relational-realizational tree including the projection, configuration and realization nodes.

D Search Algorithm

Algorithm 1 lists the pseudo code for our grammar-based generation algorithm that uses beam-search for deriving a response tree from grammar rules.

E Evaluation - Result tables

Table 1 presents generated sentences for each type of the grammars we induced and the five sentiment levels for the first experiment – comparing grammars and the human-likeness survey. The automated evaluation scores for each one of the grammars are listed in Table 2.

Table 3 presents generated sentences for each grammar and sentiment level for the experiment evaluating the relevance of the generated responses to the original input document. The automated evaluation scores are presented in Table 4.

Table 5 shows the average human-likeness rating given for each grammar in our online survey. The corresponding ordinal mixed-effects regression analysis results are presented in Table 6.

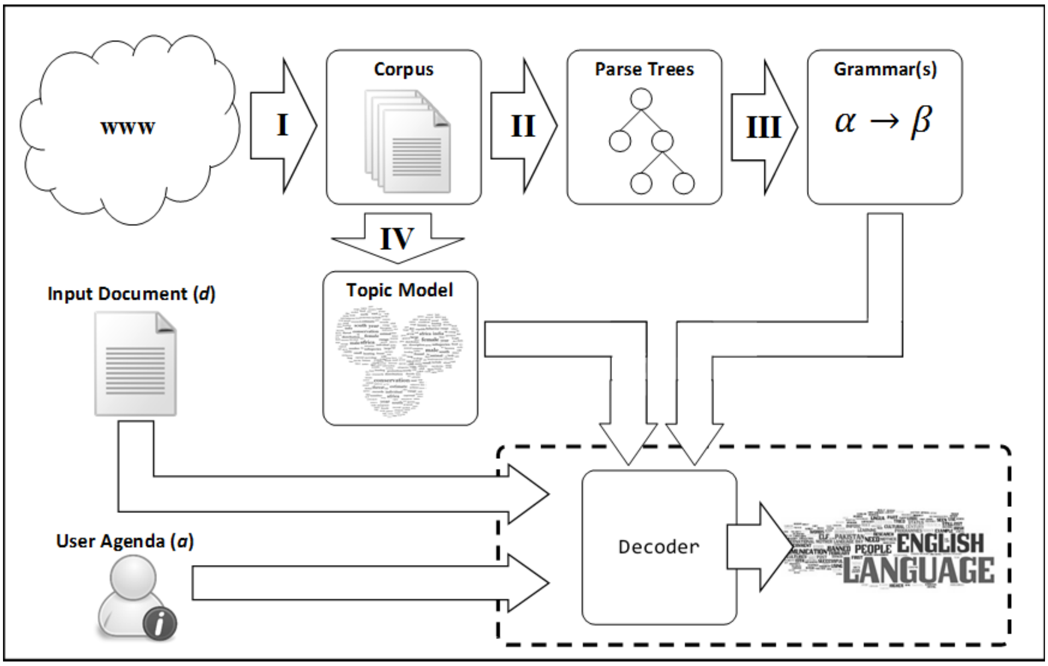
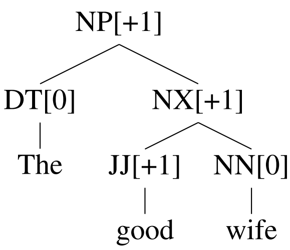


Figure 1: The end-to-end, data-driven, grammar-based generation architecture.

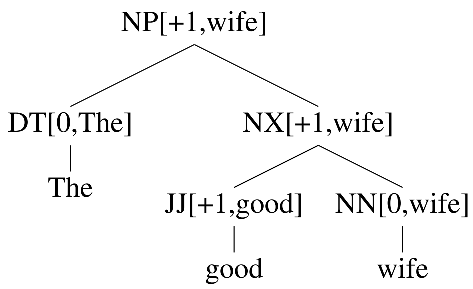


(a)

Type	LHS		RHS	
SYN	NP[+1]	→	DT[0]	NX[+1]
SYN	NX[+1]	→	JJ[+1]	NN[0]
LEX	DT[0]	→	The	
LEX	JJ[+1]	→	good	
LEX	NN[0]	→	wife	

(b)

Figure 2: Our Base grammatical representation, (a) is a sample tree and (b) is its generation sequence. Each non-terminal node is decorated with a phrase-structure label and a sentiment label. In this representation, lexical realization is conditioned only on pre-terminals and is independent of syntactic rules.

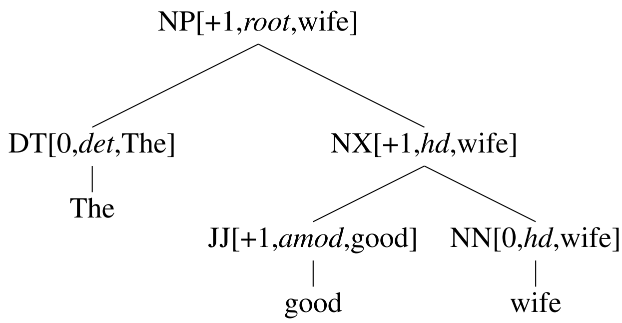


(a)

Type	LHS		RHS
HEAD	NP[+1,wife]	$\rightarrow_r$	NX[+1]
MOD	NP[+1,wife], NX[+1]	$\rightarrow_l$	DT[0]
LEX-H	NP[+1,wife],NX[+1]	$\rightarrow$	wife
LEX	NP[+1,wife], NX[+1,wife], DT[0]	$\rightarrow$	the
HEAD	NX[+1,wife]	$\rightarrow_r$	NN[0]
MOD	NX[+1,wife], NN[0]	$\rightarrow_l$	JJ[+1]
LEX-H	NX[+1,wife], NN[0]	$\rightarrow$	wife
LEX	NX[+1,wife], NN[0,wife],JJ[+1]	$\rightarrow$	good

(b)

Figure 3: Our Lexical (Lex) grammatical representation, (a) is a sample tree and (b) is its generation sequence. Each non-terminal node is decorated with a phrase-structure label, sentiment label and a lexical head. The Type column indicates Head (HEAD), Modifier (MOD), Lexical-Head (LEX-H) and Bi-Lexical (LEX) rules. Lexical dependents are generated during the derivation, conditioned on (part of the) structure.



(a)

Type	LHS		RHS
PROJ	NP[+1]	→	{*amod,det,hd*}@NP[+1]
CONF	{*amod,det,hd*}@NP[+1]	→	<*det*>@NP[+1],<{*amod,hd*}>@NP[+1]
REAL-C	<*det*>@NP[+1]	→	DT[0]
REAL-C	<{*amod,hd*} >@NP[+1]	→	NX[+1]
REAL-L	DT[0,*det*]@NP[+1,*hd*,wife]	→	The
REAL-L	NX[+1,*hd*]@NP[+1,*hd*,wife]	→	wife
PROJ	NX[+1]	→	{*amod,hd*} @NX[+1]
CONF	{amod, hd}@NX[+1]	→	<*amod*>@NX[+1] ,<*hd*>@NX[+1]
REAL-C	<*amod*>@NX[+1]	→	JJ[+1]
REAL-C	<*hd*>@NX[+1]	→	NN[0]
REAL-L	JJ[+1,*amod*]@NX[+1,*hd*,wife]	→	good
REAL-L	NN[+1,*hd*]@NX[+1,*hd*,wife]	→	wife

(b)

Figure 4: Our relational-realizational (RR) grammatical representation, (a) is a sample tree and (b) is its generation sequence. Each non-terminal node is decorated with a phrase-structure label, sentiment label, function label and a lexical head. The *Type* column indicate Projection (PROJ), Configuration (CONF), Realization–Constituency (REAL-C) and Realization–Lexicalization (REAL-L) rules. Lexical dependents are generated during the derivation, conditioned on (part of the) structure.

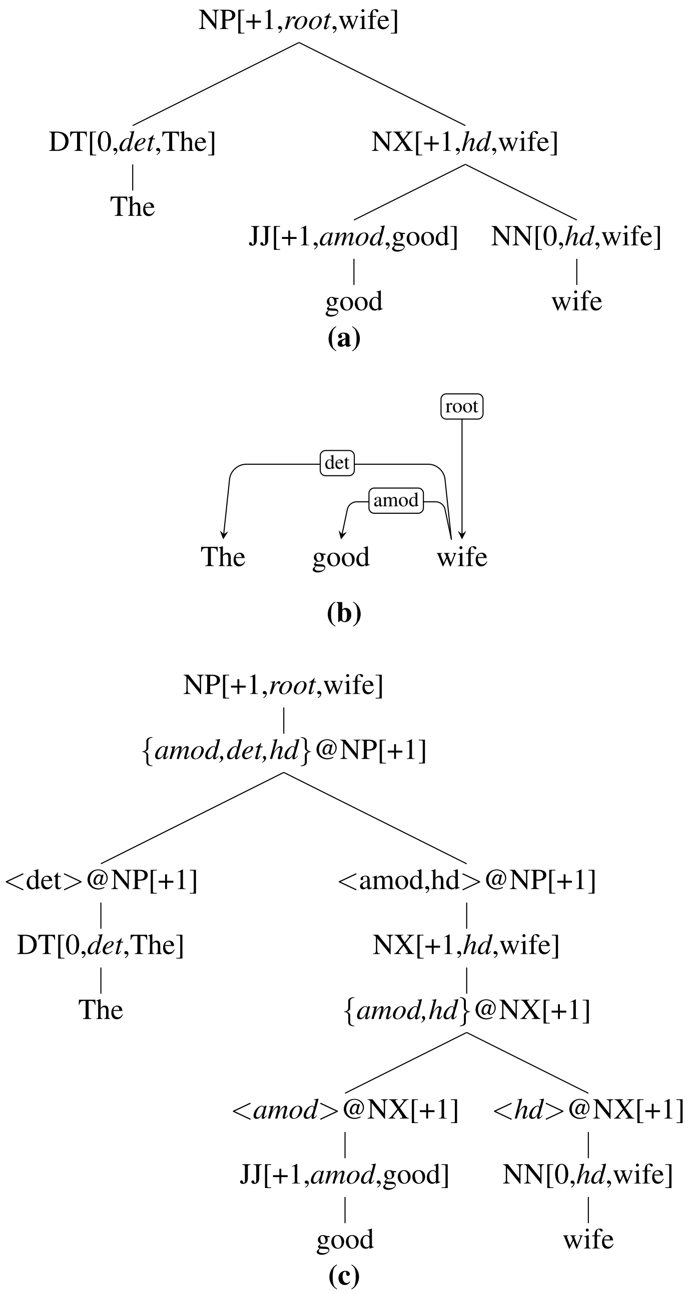


Figure 5: Annotated phrase-structure tree (a) and dependency graph (b) for the same sentence, and the corresponding RR derivation tree (c). Note that nodes with $[s_i, rel_i, l_i]$ correspond to the nodes in the decorated parse tree (a) and the other nodes represent the projection and configuration phases of the RR derivation.

Algorithm 1 Grammar-Based Generation with Beam-Search

```
Struct Payload                                          ▷ Used for book-keeping
    score                                               ▷ The score of the derivation
    path                                                ▷ The path in the ”forest” for derived tree
End Struct
Struct Rule
    score                                               ▷ Score of the rule
    lhs                                                 ▷ lhs of the rule
    rhs                                                 ▷ rhs of the rule (a list of 2 nodes)
End Struct
Struct Node
    annotation                                          ▷ The node’s annotation (type, sentiment, head)
    addRule(Rule)                                       ▷ A list of possible rules for expanding this node
End Struct

procedure BEAMSEARCHGENERATOR(topic, sentiment)
    root = new Node()
    startRules = getStartRules(topic, sentiment)
    candidates = new List(), i = 0
    for all startRule in startRules do
        candidate.add(new Payload(i, rule.getScore())   ▷ Payload with rule score
        root.addRule(rule)
        i++
    end for
    while candidate[0].size ¡ maxSize do
        intermediateList = new List()
        for all payload in candidates do
            node = getNode(root,payload.path)           ▷ Get the relevant node
                                                        ▷ according to path
            expansionRules = getNewRules(node)          ▷ Get possible rules for expansion
            i = 0
            for all rule in expansionRules do           ▷ process options in node
                score = payload.score + rule.getScore() ▷ Get the new score by adding
                                                        ▷ previous score to the score of this rule
                path = payload.path + i                 ▷ the new path is the path so far plus
                                                        ▷ the index of the rule in current node
                node.addRule(rule)                      ▷ add the new rule - expanding the derivation
                intermediateList.add(new Payload(path, score))   ▷ add new
                                                        ▷ payload/candidate
                i++
            end for
        end for
        candidates = getTop(intermediateList, k)        ▷ select the top trees so far
                                                        ▷ based on the normalized score
    end while
end procedure
```

Grammar	Sentiment	Sentence
PCFG	-2	(and badly should doesn't..
	-1	doesn't of the yankees..
	0	who is the the game,.
	1	is the the united states..
	2	is the best players..
LEX	-2	is a rhyme ... mahi mahi, and, I not quote Bunny.
	-1	Dumpster unpire are the villans.
	0	Derogatory big names symbols wider
	1	New england has been playful, and infrequent human.
	2	That's a huge award – having get fined!
RR	-2	he is very awkward, and to any ridiculous reason.
	-1	the malfeasance underscores the the widespread belief.
	0	the programs serve the purposes.
	1	McIlroy is a courageous competitor.
	2	The urgent service's a grand idea.

Table 1: Responses generated by the system with the different grammars and sentiment levels.

Grammar	Avg. LM Score		Avg. LM Score per word		Complete Sentences (%)	Sentiment Agreement / Polarity (%)	Avg. Length (words)
	Mean	CI	Mean	CI			
PCFG	-79.7	±0.054	-8.9	±0.007	20.1	13.3 / 41.8	9.5
LEX	-73.7	±0.016	-6.5	±0.002	67.3	**44.6 / 63.9**	12.3
RR	**-51.8**	±0.011	**-5.6**	±0.001	**95.7**	43.8 / 61.0	9.6
HUMAN	-50.1	±0.000	-5.4	±0.000	N/A	N/A	10.3

Table 2: Mean and 95% Confidence Interval (CI) of language model (LM) scores (evaluating *fluency*), complete sentences (evaluating *compactness*), and sentiment agreement. The last row, *HUMAN* refers to the collected human responses.

Grammar	Sentiment	Sentence
RR	-2	they deserve it, but I is fear.
	-1	the saga is correct.
	0	the indirect penalty?
	1	the job is correct.
	2	a salaries excels.
RRTM	-2	Unfortunately, they remind that to participate in baseball.
	-1	the franchise would he made?
	0	Probably the LONG time .
	1	In a good addition, he is a good baseball player.
	2	the baseball game sublime.

Table 3: Responses generated by the system using emission probabilities and topic models for the start rule selection.

Generator	Mean	CI
RR	0.473	$\pm$ 0.003
RRTM	0.424	$\pm$ 0.003
HUMAN	0.429	$\pm$ 0.000

Table 4: Mean and 95% Confidence Interval (CI) for generators with / without topic models scores (RRTM / RR respectively). The last row, *HUMAN* refers to the collected human responses.

Grammar	Mean	CI
PCFG	2.4561	$\pm$ 0.004
LEX	4.1681	$\pm$ 0.004
RR	3.7278	$\pm$ 0.004

Table 5: Mean and 95% Confidence Interval (CI) for human-likeness ratings on a scale of 1 (Certainly Computer) to 7 (Certainly Human).Higher rating is perceived as more human-like and is better.

Factor	b	Std. Error	z-value	$P(>\|z\|)$
G-LEX	2.90	0.189	15.32	<.00001
G-RR	2.33	0.164	14.20	<.00001
SENT	0.17	0.074	2.32	.020
NWORD	-1.60	0.107	-14.95	<.00001
POS	0.21	0.036	5.97	<.00001
G-LEX $\times$ SENT	-0.18	0.095	-1.91	.056
G-RR $\times$ SENT	0.44	0.096	4.53	<.00001
G-LEX $\times$ NWORD	1.31	0.117	11.16	<.00001
G-RR $\times$ NWORD	1.35	0.138	9.80	<.00001
NWORD $\times$ POS	0.10	0.037	2.81	.005

Table 6: Regression analysis results of the human-likeness survey.