Data-Driven Broad-Coverage Grammars for Opinionated Natural Language Generation (ONLG)

Tomer Cagan
School of Computer Science
The Interdisciplinary Center
Herzeliya, Israel
cagan.tomer@idc.ac.il

Stefan L. Frank
Centre for Language Studies
Radboud University
Nijmegen, The Netherlands
s.frank@let.ru.nl

Reut Tsarfaty
Mathematics and Computer Science
The Open University of Israel
Ra'anana, Israel
reutts@openu.ac.il

Abstract

Opinionated natural language generation (ONLG) is a new, challenging, NLG task in which we aim to automatically generate human-like, subjective, responses to opinionated articles online. We present a data-driven architecture for ONLG that generates subjective responses triggered by users' agendas, based on automatically acquired wide-coverage generative grammars. We compare three types of grammatical representations that we design for ONLG. The grammars interleave different layers of linguistic information, and are induced from a new, enriched dataset we developed. Our evaluation shows that generation with Relational-Realizational (Tsarfaty and Sima'an, 2008) inspired grammar gets better language model scores than lexicalized grammars à la Collins (2003), and that the latter gets better human-evaluation scores. We also show that conditioning the generation on topic models makes generated responses more relevant to the document content.

1 Introduction

Interaction in social media has become increasingly prevalent nowadays. It fundamentally changes the way businesses and consumers behave (Qualman, 2012), it is instrumental to the success of individuals and businesses (Haenlein and Kaplan, 2009) and it also affects political regimes (Howard et al., 2011; Lamer, 2012). In particular, *automatic* interaction in natural language in social media is now a common theme, as seen in the rapid popularization of chat applications, chat-bots, and "smart agents" aiming to conduct human-like interactions in natural language.

So far, generation of human-like interaction in general has been addressed mostly commercially, where there is a movement towards online response automation (Owyang, 2012; Mah, 2012), and movement away from script-based interaction towards interactive chat bots (Mori et al., 2003; Feng et al., 2006). These efforts provide an automated one-size-fits-all type of interaction, with no particular expression of particular sentiments, topics, or opinions. In academia, work on generating human-like interaction focused so far on generating responses to tweets (Ritter et al., 2011; Hasegawa et al., 2013) or taking turns in short dialogs (Li et al., 2017). However, the architectures assumed in these studies implement *sequence to sequence* (seq2seq) mappings, which do not take into account topics, sentiments or agendas of the intended responders.

Many real-world tasks and applications would benefit from automatic interaction that is generated intendedly based on a certain user profile or agenda. For instance, this can help promoting a political candidate or a social idea in social media, aiding people forming and expressing opinions on specific topics, or, in *human-computer interfaces* (HCI), making the computer-side generated utterances more meaningful, and ultimately more human-like (assuming that human-like interaction is very often affected by opinion, agenda, style, etc.).

In this work we address the *opinionated natural language generation* (ONLG) task, in which we aim to automatically generate human-like responses to opinionated articles. These responses address particular topics and reflect diverse sentiments towards them, in accordance to predefined user agendas. This is an open-ended and unstructured generation challenge, which is closely tied to the communicative goals of actual human responders.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1331–1341
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1122

In previous work we addressed the ONLG challenge using a *template-based* approach (Cagan et al., 2014). The proposed system generated subjective responses to articles, driven by *user agendas*. While the evaluation showed promising results in human-likeness and relevance ratings, the template-based system suffers from low output variety, which leads to a learning effect that reduced the perceived human-likeness of generated responses over time.

In this work we tackle ONLG from a data-driven perspective, aiming to circumvent such learning effects and repetitive patterns in template-based generation. Here, we approach generation via automatically inducing *broad-coverage generative grammars* from a large corpus, and using them for response generation. More specifically, we define a grammar-based generation architecture and design different grammatical representations suitable for the ONLG task. Our grammars interleave different layers of linguistic information — including phrase-structure and dependency labels, lexical items, and levels of sentiment — with the goal of making responses both human-like and relevant. In classical NLG terms, these grammars offer the opportunity for both *micro-planning* and *surface realization* (Reiter and Dale, 1997) to unfold together. We implement a generator and a search strategy to carry out the generation, and sort through possible candidates to get the best ones.

We evaluate the generated responses and the underlying grammars using automated metrics as well as human evaluation inspired by the Turing test (cf. Cagan et al. (2014) and Li et al. (2017)). Our evaluation shows that while *relational realizational* (RR) inspired grammars (Tsarfaty and Sima'an, 2008) get good language model scores, simple head-driven *lexicalized* grammars à la Collins (2003) get better human rating and are more sensitive to sentiment. Furthermore, we show that incorporating topic models into the grammar-based generation makes the generated responses more relevant to the document content. Finally, our human evaluations results show no learning effect. That is, human raters are unable to discover in the generated responses typical structures that would lead them to consider the responses machine-generated.

The remainder of this paper is organized as follows. In Section 2 we discuss the formal model, and in Section 3 we present the proposed end-to-end ONLG architecture. In Section 4 we introduce the grammars we define, and we describe how we use them for generation in Section 5. We follow that with our empirical evaluation in Section 6. In Section 7 we discuss related and future work, and in Section 8 we summarize conclude.

2 The Formal Model

Task Definition. Let d be a document containing a single article, and let a be a user agenda as in Cagan et al. (2014). Specifically, a user agenda a can consist of one or more pairs of a *topic* (represented by a weighted bag-of-words) and an associated *sentiment*. Let c be an analysis function on documents such that $c(d)$ yields a set of *content elements* which are also pairings of topics and sentiments. The operation $\otimes$ represents the intersection of the sets of content elements in the document and in the user agenda. We cast ONLG as a prediction function which maps the intersection $a \otimes c(d)$ to a sentence $y \in \Sigma^*$ in natural language (in our case, Σ is the vocabulary of English):

$$f_{response}(a \otimes c(d)) = y \tag{1}$$

For any non-empty intersection, a response is generated which is related to the topic of the intersection and the sentiments defined towards this topic. The relation between the sentiment in the user agenda and the sentiment reflected in the document is a simple *xor* function: when the user and the author share a sentiment toward a topic the response is positive, else it is negative.

Objective Function. Let G be a formal generative grammar and let T be the set of trees strongly generated by G. In our proposed data-driven, grammar-based, generation architecture, we define $f_{response}$ as a function selecting a most probable tree $t \in T$ derived by G, given the intersection of document content and user agenda.

$$f_{response}(a \otimes c(d)) = \underset{\{w|w=yield(t),t\in T\}}{\operatorname{argmax}} P(w,t|a \otimes c(d)) \tag{2}$$

Here, $w = yield(t)$ is the sequence of terminals that defines the leaves of the tree, which is then picked as the generated response.

Assuming that G is a context-free grammar, we can spell out the probabilistic expression in Equation (2) as a history-based probabilistic model where $root(t)$ selects a starting point for the

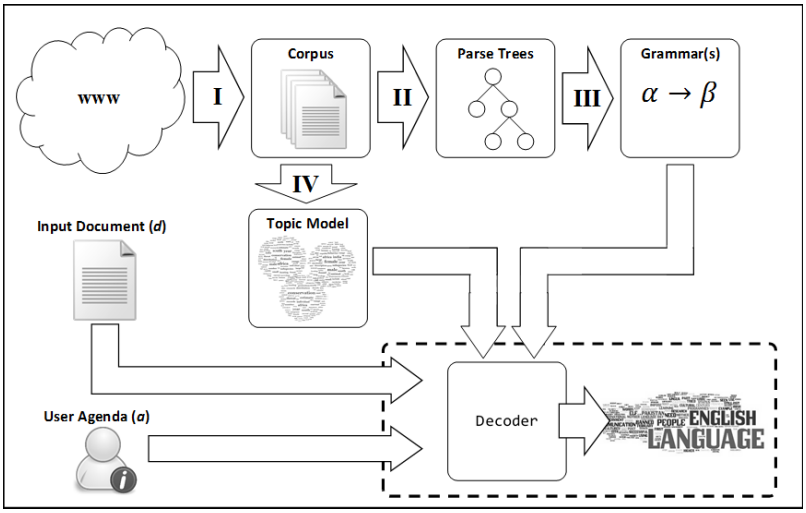


Figure 1: The end-to-end, data-driven, grammar-based generation architecture.

derivation, $der(t)$ selects the sequence of syntactic rules to be applied, and $yield(t)$ selects the sequence of terminals that forms the response all conditioned on the derivation history.

$$P(w,t|.) = P(root(t)|a \otimes c(d)) \quad (3a)$$
$$\times P(der(t)|root(t), a \otimes c(d)) \quad (3b)$$
$$\times P(yield(t)|root(t), der(t), a \otimes c(d)) \quad (3c)$$

Using standard independence assumptions, Eq. (3) may be re-written as a chain of local decisions, conditioned on selected aspects of the generation history, marked here by the function Φ.

$$P(w,t|.) \approx P(root|\Phi(a \otimes c(d))) \times \quad (4a)$$
$$\prod_{rule_j \in der(t)} P(rule_j|\Phi(root, a \otimes c(d))) \times \quad (4b)$$
$$\prod_{w_i \in yield(t)} P(w_i|\Phi(t, a \otimes c(d))) \quad (4c)$$

In words, the probability of the starting rule (4a) is multiplied with the probability of each of the rules in the derivation (4b) and the probability of each of the terminal nodes in the tree (4c). Each decision may be conditioned on previously generated part(s) of the structure, as well as the intersection of the input document content and user agenda.

3 The Architecture

A bird's-eye view of the architecture we propose is depicted in Figure 1. The process consists of an offline component containing (I) *corpus collection*, (II) *automatic annotation*, (III) *grammar induction*, and (IV) *topic-model training*. The induced grammar along with a predefined user agenda and the pre-trained topic model are provided as input to the online generation component, which is marked with the dashed box in Figure 1.

In (I) *corpus collection*, we collect a set of documents D with corresponding user comments. The documents in the corpus are used for training a topic model (IV), which is used for topic inference given a new input document d. The collected comments are used for inducing a wide-coverage grammar G for response generation.

To realize the goal of ONLG, we aim to jointly model opinion, structure and lexical decisions in our induced grammars. To this end, in (II) *automatic annotation* we enrich the user comments with annotations that reflect different levels of linguistic information, as detailed in Section 4.

In (III) *grammar induction* we induce a generative grammar G from the annotated corpus, following the common methodology of inducing PCFGs from syntactically annotated corpora (Charniak, 1995; Collins, 2003). We traverse the annotated trees from (III) and use maximum likelihood estimation for learning rule probabilities. No smoothing is done, and in order to filter noise from possibly erroneous parses, we use a frequency cap to define which rules can participate in derivations.

We finally define and implement an efficient grammar-based generator, termed here the *decoder*, which carries out the generation and calculates the objective function in Eq. (4). The algorithm is described in Section 5.

4 The Grammars

Base Grammar. A central theme in this research is generating sentences that express a certain sentiment. Our base grammatical representation is inspired by the Stanford sentiment classification parser (Socher et al., 2013) which annotates every non-ternminal node with one of five sentiment classes $s \in \{-2, -1, 0, 1, 2\}$.

Formally, each non-terminal in our base grammar includes a constituency category C and a sentiment class label s. The derivation of depth-1 trees with a parent node p and two daughters d_1, d_2 will thus appear as follows:

$$C_p[s_p] \rightarrow C_{d1}[s_{d1}] \;\; C_{d2}[s_{d2}]$$

The generative story imposed by this grammar is quite simple: each non-terminal node annotated with a sentiment can generate either a sequence of non-terminal daughters, or a single terminal node.

An example of a subtree and its generation sequence is given in Figure 2(Base). Here we see a positive NP which generates two daughters: a neutral DT and a positive NX. The positive NX generates a neutral noun NN and a positive modifying adjective JJ on its left. Such a derivation can yield NP terms such as "the good wife" or "an awesome movie", but will not generate "some terrible words". In this grammar, lexical realization is generated conditioned on local pre-terminals only, and independently of the syntactic structure.

While the generative story is simple, this grammar can capture complex interactions of sentiment. Such interactions take place in tree structures that include elements that may affect polarity, such as negation, modal verbs and so on (see Socher et al. (2013) and examples therein). In this work we assume a completely data-driven approach wherein such structures are derived based on previously observed sentiment-interactions in sentiment-augmented parses.

Lexicalized Grammar. Our base grammar suffers from a clear pitfall: the structure lacks sensitivity to lexical information, and vice versa. This base grammar essentially generates lexical items as an afterthought, conditioned only on the local part-of-speech label and sentiment value. Our first modification of the base grammar is *lexicalization* in the spirit of Collins (2003).

In this representation each non-terminal node is decorated with a phrase-structure category C and a sentiment label s, and it is augmented with a lexical *head* l_h. The lexical head is common to the parent and the left (or right) daughter. A new lexical item, termed *modifier* l_m, is introduced in the right (left) daughter. The resulting depth-1 subtree for a parent p with daughters d_1, d_2 and a lexical head on the left (without loss of generality) is:

$$C_p[s_p, l_h] \rightarrow C_{d1}[s_{d1}, l_h] \;\; C_{d2}[s_{d2}, l_m]$$

Lexicalization makes the grammar more useful for generation as lexical choices can be made at any stage of the derivation conditioned on part of the structure. But it has one drawback – it assumes very strong dependence between lexical items that happen to appear as sisters.

To overcome this, we define a *head-driven* generative story that follows the model of Collins (2003), where the mother non-terminal generates first the head node, and then, conditioned on the head it generates a modifying constituent to the left (right) of the head and its corresponding modifying lexical dependent. An example subtree and its associated head-driven generative story is illustrated in Figure 2(Lex).

Relational-Realizational Grammar. Generating phrase-structures along with lexical realization can manage *form* — control how sentences are built. For coherent generation we would like to also control for the *function* of nodes in the derivation. To this end, we define a grammar and a generative story in the spirit of the *Relational-Realizational* (RR) grammar of Tsarfaty (2010).

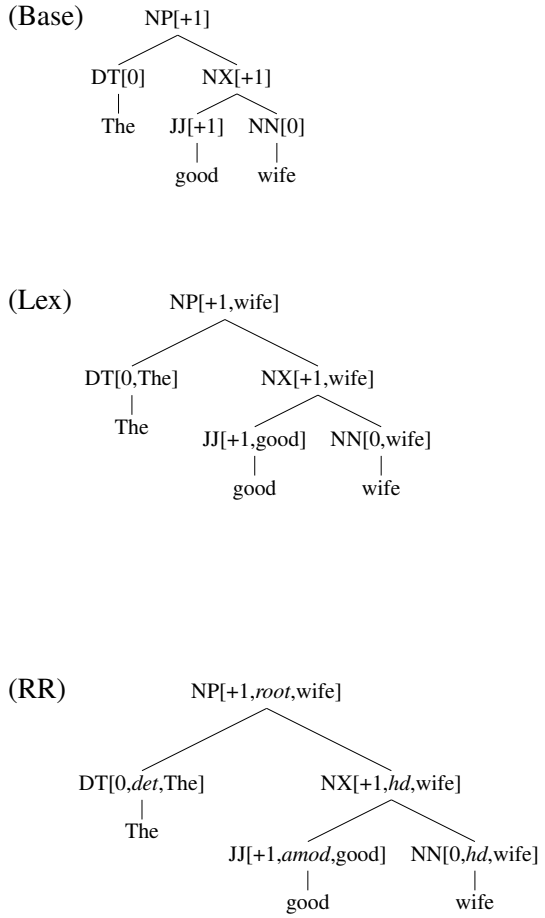
In our RR-augmented trees, each non-terminal node includes, on top of the phrase-structure category C, the lexical head l and the sentiment s, a relation label dep_i which determines its functional role in relation to its parent. The functional component will affect the selection of daughters so that the derived subtree fulfils its function. A depth-1 subtree will thus appear as follows:

$$C_i[s_i, dep_i, l_i] \rightarrow C_j[s_j, dep_j, l_i] \;\; C_k[s_k, dep_k, l_k]$$

The generative story of our RR representation follows the three-phase process defined by Tsarfaty and Sima'an (2008) and Tsarfaty (2010):

(i) *projection*: given a constituent and a sentiment value, generate a set of grammatical relations which define the functions of the daughters to be generated.

(a)



(b)

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	

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]	→	wife
LEX	NP[+1,wife], NX[+1,wife], DT[0]	→	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]	→	wife
LEX	NX[+1,wife], NN[0,wife],JJ[+1]	→	good

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

Figure 2: Our grammatical representations, with (a) a sample tree and (b) its generation sequence. A rule of type SYN marks syntactic rules, LEX indicates lexical realization, HEAD, MOD indicate head selection and modifier selection, PROJ,CONF,REAL indicate projection, configuration and realization, respectively. The @ sign indicates aspects in the generation history that the production is conditioned on (Φ in eq. 4).

(ii) *configuration*: given a constituent, sentiment and an unordered set of relations, an ordering for the relations is generated. Unlike the original RR derivations which fully order the set, here we partition the set into two disjoint sets (one of which is a singleton) and order them. This modification ensures that we adhere to binary trees.

(iii) *realization*: For each function-labels' set we select the daughter's constituent realizing it. We first generate the constituent and sentiment realizing this function, and then, conditioned on the constituent, sentiment, head and function, we select the lexical dependent.

An example tree along with its RR derivation is given in Figure 2(RR).

5 Grammar-Based Generation

Our grammar-based generator is a top-down algorithm which starts with a frontier that includes a selected root, and expands the tree continually by substituting non-terminals at the left-hand-side of rules with their daughters on the right hand side, until no more non-terminals exist. This generation procedure yields one sentence for any given root. Due to independence assumptions inherent in the generative processes we defined, there is no guarantee that generated sentences will be completely grammatical, relevant and human-like. To circumvent this, we develop an over-generation algorithm that modifies the basic algorithm to select multiple rules at each generation point, and apply them to uncover several derivation trees, or a *forest*.

We then use a variation on the beam search algorithm (Reddy, 1977) and devise a methodology to select the k-best scoring trees to be carried on to the next iteration. Specifically, we use a Breadth-First algorithm for expanding the tree and define a dynamic programming algorithm that takes the score of a derivation tree of $n-1$ *expanded* nodes, selects a new rule for the next non-expanded node, and from it, calculates the score of the expanded tree with now n nodes. For comparing the trees, we computed a score according to Eq. (4) for the tree generated so far, and used an average node score to neutralize size difference between trees.

To make sure our responses target a particular topic, we propose to condition the selection of lexical items at the root on the *topic* at the intersection of the document content and user agenda, essentially preferring derivations that yield words related to the input topic distribution. In practice we use topic model scores to estimate the root rule probability, selecting lexical item(s) for generation to start with:

$$\hat{P}(root(t)|a \otimes c(d)) = \hat{P}(\text{ROOT} \rightarrow l_1 l_2 | a \otimes c(d)) = \sum_{c=1}^{N} \sum_{i=1}^{2} tm\_weight(c) * word\_weight(c, l_i) \quad (5)$$

where $tm\_weight(c)$ is the weight of topic c in the topic distribution at the document-agenda intersection, and $word\_weight(c, l_i)$ is the weight of the lexical head word l_i within the word distribution of topic c in the given topic model.

The generation process ends when all derivations reach (at most) a pre-defined height (to avoid endless recursions). We then re-rank the generated candidates. The re-ranking is based on a 3-grams language model on the raw yield of the sentence, divided by the length of the sentence to obtain a per-word average and avoid length biases.[1]

6 Evaluation

Goal. We aim to evaluate the grammars' applicability to the ONLG task. Set in an open domain, it is not trivial to find a "gold-standard" for this task, or even a method to obtain one. Our evaluation thus follows two tracks: an automated assessment track, where we quantitatively assess the responses, and a Turing-like test similar to that of Cagan et al. (2014), where we aim to gauge human-likeness and response relevance.

[1]Here we use Microsoft's WebLM API which is part of the Microsoft Oxford Project (Microsoft, 2011).

Materials. We collected a new corpus of news articles and corresponding user comments from the NY-Times®web site, using their open Community API. We focus on sports news, which gave us 3,583 news articles and 13,100 user comments, or 55,700 sentences. The articles are then used for training a topic model using the Mallet library (McCallum, 2002). Next, we use the comments in the corpus to induce the grammars. To obtain our Base representation we parse the sentences using the Stanford CoreNLP suite (Manning et al., 2014) which can provide both phrase-structure and sentiment annotation. To obtain our Lexicalized representation we follow the same procedure, this time also using a head-finder which locates the head word for each non-terminal. To obtain the Relational-Realizational representation we followed the algorithm described in Tsarfaty et al. (2011), which, given both a constituency parse and a dependency parse of a sentence, unifies them into a lexicalized and functional phrase-structure. The merging is based on matching spans over words within the sentence.[2]

Setup. We simulated several scenarios. In each, the system generates sentences with one grammar ($G \in \{$*Base, Lex, RR*$\}$) and one scoring scheme (with/without topic model scores). The results of each simulation are 5,000 responses for each variant of the system, consisting of 1,000 sentences for each sentiment class, $s \in \{-2, -1, 0, 1, 2\}$. The same 5000 generated sentences were used in all experiments. We set the generator for trees of maximum depth of 13 which can yield up to 4096 words. In reality, the realization was of much shorter sentences. Examples for generated responses are given in Table 1.

6.1 Comparing Grammars

Goal and Metrics. In this experiment we compare and contrast the generation capacity of the grammars, using the following metrics:

(i) *Fluency* measures how grammatical or natural the generated sentences are. We base this measure on a probabilistic language model which gives an indication of how common word-sequences within the sentence are. We express fluency as a Language Model (LM) score which is calculated using the Microsoft Web ML API to get aggregated minus-log probabilities of all 3-grams

[2]The collected corpus and supplementary annotations are available at www.tomercagan.com/onlg.

in the sentence. The aggregated score is then normalized to give a per-word average in order to cancel any effects of sentence length.

(ii) *Sentiment Agreement* measures whether the inferred sentiment of the response matches the input sentiment parameter used for generation. Specifically, we take the raw yield of the generated tree (a sentence) and run it through the sentiment classifier implemented in Socher et al. (2013), to assign the full sentence one of 5 sentiment classes between -2 and $+2$. During evaluation, we compare the classified sentiment of the generated sentence is with the sentiment entered as input for the derivation of the sentence, and report the rate of agreement on (a) level ($-2.. + 2$) and (b) polarity ($-/+$), which is a more relaxed measure.

(iii) The *Consiceness/tightness* metric aims to evaluate which grammar derives a simpler structure across generations of similar content. Our tightness evaluation is based on the percentage of sentences that were fully realization as terminals within the specific height limit;[3] we simply observe how many trees have all leaves as terminal symbols. Intuitively, tighter grammars lead to improved performance and better control over the generated content. It is possible to think of what it captures in terms Occams Razor, preferring the simpler structure to derive comparable outcome.

Empirical Results The results of our evaluation are presented in Table 2. With respect to the above metrics, the RR grammar was more compact and natural compared to the lexicalized (LEX) grammar: the per-word LM Score for the RR is -5.6 as compared to -6.5 for LEX. Also, RR has 95.7% complete sentences as compared to only 67.3% for LEX. The LEX grammar was more sensitive to the sentiment input but only slightly, having a 44.6% sentiment agreement and 63.9% sentiment polarity agreement compared to 43.8% and 61.0% for RR grammar. The BASE grammar gave the worst performance for all measures. This provides preliminary evidence in support of incorporating surface realization (lexicalization) into the syntactic generation, rather than filling slots in retrospect.

6.2 Testing Relevance

Goal and Metrics Next we aim to evaluate the relevance of the responses to the input document triggering the response. We do so by calculating

[3] A height of 13 makes a maximum sentence length of $2^{13-1} = 2^{12} = 4096$ words.

Grammar	Sentiment	Sentence
BASE	-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			
BASE	-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 scores, and measures of compactness and sentiment agreement. The last row, *HUMAN* refers to the collected human responses.

Topic Agreement, a measure that, given a trained topic model, determines how close the topic distribution of the input document and that of the generated response are. We use L2 to calculate the distance between the inferred topic distribution vectors. We focus here on relevance testing for the RR grammar, which gave superior LM scores. In this test we use two generators – RR generator as defined above, and RRTM generator that uses the scoring scheme of Equation (5) to select a start rule deriving the root lexical item. Example sentences of each generator are presented in Table 3.

Empirical Results The results of the two generators and their average distance from the topic distribution of the input document are presented in Table 4. Here we see that the generator using topic models for selecting start rules (RRTM) gets topic distribution that is closer to the input document's topic distribution. The last row, HUMAN, calculates the distance between the topic distributions in the documents and their human responses from the collected corpus. The fact that RRTM outperforms HUMAN is not necessarily surprising, as sentences in human responses are typically from longer paragraphs where some sentences are more generic, used as connectives, interjections, etc.

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.

6.3 Human Surveys

Goal and Procedure. We evaluate human-likeness of the generated responses by collecting data via an online survey on Amazon Mechanical Turk. In the survey, participants were asked to judge whether generated sentences were written by a human or a computer. The participants were screened to have a good level of English and reside in the US. Each survey comprised of 50 randomly ordered trials. In each trial the participant was shown a response. The task was to categorize each response on a 7-point scale with labels 'Certainly human/computer', 'Probably human/computer', 'Maybe human/computer' and 'Unsure'. In 50 trials the participant was exposed to 3-4 sentences for each grammar/sentiment combination.

Empirical Results. Average human-likeness ratings (scale 1–7) are presented in Table 5. Here, we see that sentences generated by the lexicalized grammar were perceived as most human-like. This result is in contrast with the automatic evaluation. Such a discrepancy need not be very surprising, as noted by others before (Belz and Reiter, 2006). Cagan et al. (2014) show that there are extra-grammatical factors affecting human-likeness, e.g. world knowledge. We hypothesise that the LEX grammar, which relies heavily on lexical co-occurrences frequencies, is better at replicating world knowledge and idiomatic phrases thus judged as more human.

Grammar	Mean	CI
BASE	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 (scaling 1:low–7:high).

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 of the human survey.

In a qualitative inspection on a sample of the results we could verify that the LEX grammar tends to replicate idiomatic sequences while the RR grammar generates novel phrases in a more compositional fashion. Grammaticality is not hindered by it, but apparently human-likeness is.

We also run an ordinal mixed-effects regression, which is an appropriate way to analyse discrete rating data. Regression model predictors were Grammar (G), sentiment level (SENT), response length (NWORD), position of response in rating session (POS), and all two-way interactions between these. Quantitative predictors were standardized and non-significant ($p > .05$) interactions were dropped from the fitted model. By-participant random intercepts and slopes of G and SENT were included as random effects.

Table 6 displays the fitted model fixed effects, with BASE grammar as the reference level. Consistent with Table 5, we see that LEX and RR score significantly higher on human likeness than BASE. These effects are modulated by sentiment: more positive sentiment makes BASE and RR more human-like (respectively: $b = 0.17$ and $b = 0.44$) whereas the LEX grammar becomes less human like (although this effect is only marginally significant: $b = -.18$). In addition, these effects are also modulated by sentence length in #words – longer sentences make BASE less human-like ($b = -1.60$) but RR and LEX more human-like (respectively: $b = 1.31$ and $b = 1.35$)

Importantly, there is a weak but significant *positive* effect of position ($b = 0.21$), indicating that human-likeness ratings increase over the course of a rating session. This effect does not depend on the grammar, but is somewhat stronger for longer

sentences ($b = 0.10$). The position effect contrasts markedly with the decrease of human-likeness ratings that (Cagan et al., 2014) ascribed to a learning effect: there, raters noticed the repetitive structure and took this to be a sign that the utterances were machine generated. The fact that we find no such effect means that our grammars successfully avoided such repetitiveness.

7 Related and Future Work

NLG is often cast as a *concept-to-text* (C2T) challenge, where a structured record is transformed into an utterance expressing its content. C2T is usually addressed using template-based (Becker, 2002) or data-driven (Konstas and Lapata, 2013; Yuan et al., 2015) approaches. In particular, researchers explored data-driven grammar-based approaches (Cahill and van Genabith, 2006), often assuming a custom grammar (Konstas and Lapata, 2013) or a closed-domain approach (DeVault et al., 2008). ONLG in contrast is set in an open domain, and expresses multiple dimensions (grammaticality, sentiment, topic).

In the context of social media, generating responses to tweets has been cast as a sequence-to-sequence (seq2seq) transduction problem, and has been addressed using statistical machine translation (SMT) methods (Ritter et al., 2011; Hasegawa et al., 2013). In this seq2seq setup, moods and sentiments expressed in the past are replicated or reused, but these responses do not target particular topics and are not driven by a concrete user agenda. An exception is a recent work by Li et al. (2016), exploring a persona-based conversational model, and Xu et al. (2016) who encode loose structured knowledge to condition the generation on. These studies present a stepping stone towards full-fledge neural ONLG architectures with some control over the user characteristics.

The surge of interest in neural network generation architectures has spawned the development of seq2seq models based on encoder-decoder setup (Sordoni et al. (2015); Li et al. (2016, 2017) and references therein). These architectures require a very large dataset to train on. In the future we aim to extend our dataset and explore neural network architectures for ONLG that can encode a user-agenda, a document, and possibly stylistic choices (Biber and Conrad, 2009; Reiter and Williams, 2010) — in the hope of yielding more diverse, relevant and coherent responses to online content.

8 Conclusion

We approached ONLG from a data-driven perspective, aiming to overcome the shortcomings of previous template-based approaches. Our contribution is threefold: (i) we designed three types of broad-coverage grammars appropriate for the task, (ii) we developed a new enriched data-set for inducing the grammars, and (iii) we empirically demonstrated the strengths of the LEX and RR grammars for generation, as well as the overall usefulness of sentiment and topic models incorporated into the syntactic derivation. Our results show that the proposed grammar-based architecture indeed avoids the repetitiveness and learning effects observed in the template-based ONLG.

To the best of our knowledge, this is the first data-driven agenda-driven baseline for ONLG, and we believe it can be further improved. Some future avenues for investigation include improving the relevance and human-likeness results by improving the automatic parses quality, acquiring more complex templates via abstract grammars, and experimenting with more sophisticated scoring functions for reranking. With the emergence of deep learning, we further embrace the opportunity to combine the *sequence-to-sequence* modeling view explored so far with conditioning generation on speakers agendas and user profiles, pushing the envelope of opinionated generation further. Finally, we believe that future work should be evaluated *in situ*, to examine if, and to what extent, the generated responses participate in and affect the discourse (feed) in social media.

References

Tilman Becker. 2002. Practical, template-based natural language generation with TAG. In *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+6)*.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *Proceeding of EACL'06*. pages 313–320.

D. Biber and S. Conrad. 2009. *Register, Genre, and Style*. Cambridge Textbooks in Linguistics. Cambridge University Press. https://books.google.de/books?id=0HUhombmOJUC.

Tomer Cagan, Stefan L. Frank, and Reut Tsarfaty. 2014. Generating subjective responses to opinionated articles in social media: An agenda-driven architecture and a Turing-like test. In *Proceedings of the Joint Workshop on Social Dynamics and*

Personal Attributes in Social Media. Association for Computational Linguistics, pages 58–67. http://www.aclweb.org/anthology/W/W14/W14-2708.

Aoife Cahill and Josef van Genabith. 2006. Robust PCFG-based generation using automatically acquired LFG approximations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL-44, pages 1033–1040. https://doi.org/10.3115/1220175.1220305.

Eugene Charniak. 1995. Parsing with context-free grammars and word statistics. Technical report, Providence, RI, USA.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics* 29(4):589–637. https://doi.org/10.1162/089120103322753356.

David DeVault, David Traum, and Ron Artstein. 2008. Practical grammar-based NLG from examples. In *Proceedings of the Fifth International Natural Language Generation Conference*. Association for Computational Linguistics, Stroudsburg, PA, USA, INLG '08, pages 77–85. http://dl.acm.org/citation.cfm?id=1708322.1708338.

Donghui Feng, Erin Shaw, Jihie Kim, and Eduard Hovy. 2006. An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of Intelligent User Interface (IUI-2006)*. pages 171–177.

Michael Haenlein and Andreas M. Kaplan. 2009. Flagship brand stores within virtual worlds: The impact of virtual store exposure on real-life attitude toward the brand and purchase intent. *Recherche et Applications en Marketing (English Edition)* 24(3):57–79. https://doi.org/10.1177/205157070902400303.

Takayuki Hasegawa, Nobuhiro Kaji, Naoki Yoshinaga, and Masashi Toyoda. 2013. Predicting and eliciting addressee's emotion in online dialogue. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 964–972. http://www.aclweb.org/anthology/P13-1095.

Philip N. Howard, Aiden Duffy, Deen Freelon, Muzammil Hussain, Will Mari, and Marwa Mazaid. 2011. Opening closed regimes: What was the role of social media during the Arab spring? *Project on Information Technology and Political Islam.* http://pitpi.org/index.php/2011/09/11/opening-closed-regimes-what-was-the-role-of-social-media-during-the-arab-spring/.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *Journal of Artificial Intelligence Research* 48:305–346.

Wiebke Lamer. 2012. Twitter and tyrants: New media and its effects on sovereignty in the Middle East. *Arab Media and Society* http://www.arabmediasociety.com/?article=798.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. *CoRR* abs/1603.06155. http://arxiv.org/abs/1603.06155.

Jiwei Li, Will Monroe, Tianlin Shi, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *CoRR* abs/1701.06547. http://arxiv.org/abs/1701.06547.

Paul Mah. 2012. Tools to automate your customer service response on social media. Visited August 2013. http://www.itbusinessedge.com/blogs/smb-tech/tools-to-automate-your-customer-service-response-on-social-media.html.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. `http://www.cs.umass.edu/~mccallum/mallet`.

Microsoft. 2011. Microsoft cognitive services. https://www.microsoft.com/cognitive-services/en-us/web-language-model-api.

Kyoshi Mori, Adam Jatowt, and Mitsuru Ishizuka. 2003. Enhancing conversational flexibility in multimodal interactions with embodied lifelike agent. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*. ACM, New York, NY, USA, IUI '03, pages 270–272. https://doi.org/10.1145/604045.604096.

Jeremiah Owyang. 2012. Brands Start Automating Social Media Responses on Facebook and Twitter. Visited August 2013. http://techcrunch.com/2012/06/07/brands-start-automating-social-media-responses-on-facebook-and-twitter/.

Erik Qualman. 2012. *Socialnomics: How social media transforms the way we live and do business*. John Wiley & Sons, Hoboken, NJ, USA, 2nd edition. https://books.google.co.il/books?id=yAqD19i2U0UC.

D. Raj Reddy. 1977. Speech understanding systems: summary of results of the five-year research effort at Carnegie-Mellon University. Technical report, Carnegie-Mellon University.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3(1):57–87. https://doi.org/10.1017/S1351324997001502.

Ehud Reiter and Sandra Williams. 2010. Generating texts in different styles. In Shlomo Argamon, Kevin Burns, and Shlomo Dubnov, editors, *The Structure of Style - Algorithmic Approaches to Understanding Manner and Meaning.*, Springer, pages 59–75.

Alan Ritter, Colin Cherry, and William B. Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '11, pages 583–593. http://dl.acm.org/citation.cfm?id=2145432.2145500.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, pages 1631–1642.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 196–205. http://www.aclweb.org/anthology/N15-1020.

Reut Tsarfaty. 2010. *Relational-Realizational Parsing*. Ph.D. thesis, Institute for Logic, Language and Computation, University of Amsterdam.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2011. Evaluating dependency parsing: Robust and Heuristics-Free Cross-Annotation evaluation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. pages 385–396. http://www.aclweb.org/anthology/D11-1036.

Reut Tsarfaty and Khalil Sima'an. 2008. Relational-realizational parsing. In *Proceedings of the 22Nd International Conference on Computational Linguistics*. Association for Computational Linguistics, pages 889–896. http://dl.acm.org/citation.cfm?id=1599081.1599193.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2016. Incorporating loose-structured knowledge into LSTM with recall gate for conversation modeling. *CoRR* abs/1605.05110. http://arxiv.org/abs/1605.05110.

Caixia Yuan, Xiaojie Wang, and Qianhui He. 2015. *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, Association for Computational Linguistics, chapter Response Generation in Dialogue Using a Tailored PCFG Parser, pages 81–85. http://aclweb.org/anthology/W15-4713.