Stochastic Language Generation Using Situated PCFGs

Caixia Yuan(✉), Xiaojie Wang, and Ziming Zhong

Beijing University of Posts and Telecommunications, Beijing 100876, China
{yuancx,xjwang,zimingzhong}@bupt.edu.cn

Abstract. This paper presents a purely data-driven approach for generating natural language (NL) expressions from its corresponding semantic representations. Our aim is to exploit a parsing paradigm for natural language generation (NLG) task, which first encodes semantic representations with a situated probabilistic context-free grammar (PCFG), then decodes and yields natural sentences at the leaves of the optimal parsing tree. We deployed our system in two different domains, one is response generation for a Chinese spoken dialogue system, and the other is instruction generation for a virtual environment in English language, obtaining results comparable to state-of-the-art systems both in terms of BLEU scores and human evaluation.

Keywords: Natural language generation · Meaning representation · Situated PCFG

1 Introduction

Natural language generation (NLG) is the task of constructing natural-language sentence from formal, abstract meaning representation (MR) (Reiter and Dale, 2000). Depending on the application at hand, the meaning representation can have various forms such as database records, domain knowledge bases, geo-information. It is generally assumed that the core tasks of language generation process can be split up into two stages: (1) content selection, which decides what meanings to express, and (2) surface realization, which expresses those meanings using natural language (Belz and Kow, 2009). Over the past decade, statistical methods for NLG have received considerable attention (e.g., Wong and Mooney, 2007; Belz, 2008; Konstas and Lapata, 2012; McKinley and Ray, 2014). However, this prior work is mostly based on hand-crafted generation rules, which are extensive, but also expensive. Furthermore, although it is a consensus that at a rather abstract level natural language generation can benefit a lot from its counterpart natural language understanding (NLU), the problem of leveraging NLU resources for NLG remains pretty much open.

In this paper, we propose a data-driven natural language generation model which exploits a PCFG parser to assist natural language generation. The basic idea underlying our method is that the generated sentence is licensed by a

J. Li et al. (Eds.): NLPCC 2015, LNAI 9362, pp. 64–75, 2015.
DOI: 10.1007/978-3-319-25207-0_6

context-free-grammar, and thus can be deduced from a parsing tree which encodes hidden structural linkage between meaning representation and its sentence expression. We operate in a setting in which we are given a set of records, where each record is a data pair consisting of a structured meaning representation and its natural language sentence. A situated PCFG - i.e., a PCFG with the context of application-specific concepts, is learned from data pairs and then used to guide generation processes for other previously unseen meaning representations. Table 1 exemplifies two records from the two applications at hand.

The strength of our approach is that it allows generation process to be represented as an optimization problem within a tree structure, without concerns about how the surfacial words are ordered and selected, and without the need to manually define PCFG derivations, which is one of the most important prerequisites in work of (Belz and Kow, 2009) and (Konstas and Lapata, 2012). We demonstrate the versatility and effectiveness of our method on (1) response generation for a situated Chinese spoken dialogue system (SDS)[1] for booking meeting rooms, and (2) GIVE (Generating Instructions in Virtual Environments)[2] challenge, within which a NLG module generates a sequence of English instructions that will help in a "treasure hunt" task in a virtual 3D environment.

Table 1. Examples of meaning representation input as a structured database and its corresponding natural language expression. Each meaning representation has several fields, each field has a value.

(a) SDS

Meaning representation	action1 confirm	object1 budget	value11 2,000	value12 2,500	action2 request	object2 date	value21 null	value22 null
Text	您的预算在2,000元到2,500元之间，请问您在哪天开会？ (Your budget is between 2,000 yuan and 2,500 yuan. When is the meeting scheduled?)							

(b) GIVE

Meaning representation	action move	direction slightly right	visible null	adj blue	object button	reference true	adj2 green	ref-object button
Text	Go to the blue button near to a green button. It should be in front of you slightly to the right.							

2 Related Work

Over the past decade, there has been a surge of interest in statistical techniques for natural language generation, a methodology that was largely inspired by the blossom of statistical natural language processing. Statistical NLG mainly follows two streams of research. The one is to introduce statistics at the sentence

[1] A demo can be found at http://www.aidc.org.cn:8008/WebContent/

[2] More about GIVE challenge can be seen at http://www.give-challenge.org/research/

generation level by training a model which refines or reranks candidate outputs of a handcrafted generator. A pioneering work is Langkilde and Knight's Nitrogen systems (Langkilde and Knight, 1998), which first generates a candidate set of sentences and then reranks them using an n-gram language model trained on news articles. Langkilde and Knight proved that the statistical post-processor yielded more fluent outputs and reduced the need for deep, hand-crafted grammars. In order to produce more customerized outputs, Walker et al. investigate a trainable sentence planner on the basis of feedback from users (Walker et al., 2002). The major drawbacks of such "overgenerate and rank" approach are their inherent computational cost and not grammatically informed.

The second stream of research has focused on introducing statistics at the generation decision level by training models that find the set of generation parameters maximizing an objective function, e.g., generating the most likely context-free derivations (Belz, 2008; Konstas and Lapata, 2012), or maximizing the expected reward using reinforcement learning (Rieser and Lemon, 2009; Dethlefs and Cuayahuitl, 2014). While such methods do not suffer from overgeneration problem, they still require a set of handcrafted generation rules or reward functions to derive a generation decision space within which an optimal sentence can be deduced statistically. Our model is closest to (Konstas and Lapata, 2013) who reformulates the Markov structure between a world state and a string of text depicted in (Liang, et al., 2009) into a set of CFG rewrite rules, and then deduces the best derivation tree for a set of database records. Although this Markov structure can capture a few elements of rudimentary syntax, it is essentially not linguistic grammars. Thus the sentences produced by this model are usually ungrammatically informed (for instance, its 1-best model outputs grammatically illegal sentences like "Milwaukee Phoenix on Saturday on Saturday on Saturday on Saturday"). (Konstas and Lapata, 2013) claims that dependency structure is an efficient complementary to CFG grammar, and incorporates dependency information between words into the reranking procedure to boost the performance.

Although conceptually related to (Konstas and Lapata, 2013), our model directly learns more grammatical rewrite rules from hybrid syntactic trees whose non-terminal nodes are comprised of phrasal nodes inheriting from a syntactic parser and conceptual nodes designed for encoding target meaning representation. Therefore, the learning aspect of two models is fundamentally different. We have a single CFG grammar that applies throughout, whereas they train different CFG grammar and dependency grammar respectively.

3 Problem Formulation

3.1 The Grammar

Following most previous works in this area (Liang, et al., 2009; Konstas and Lapata, 2013), we use the term record r to refer to a (m, w) pair. Each meaning representation m is described as several fields f, each field has value $f.v$. As exemplified in Table 1, each m in GIVE system has eight fields: action, direction,

visible, adj (adjunct), object, reference, adj2 (adjunct 2) and ref-object (reference object). Each field has a specific value. The value can be a string (e.g., blue, button), a numeric quantity (e.g., 2000, 2500), or null. The text is simply a sequence of words $w = (w_1, ..., w_{|w|})$.

Our goal is to learn a PCFG for paraphrasing a MR with NL expression. As mentioned in Section 2, prior research on CFG based natural language generation has mainly focused on relatively simple grammar, either hand-crafted grammars for deterministic parsing (Belz and Kow, 2009), or probabilistic regular grammar describing Markov dependency (Konstas and Lapata, 2013) among fields and word strings. In order to generate more grammatical sentence, the established grammar should capture recursive structure of phrases. Meanwhile, in order to generate sentence expressing target meanings, the grammar should also capture concept embeddings corresponding to desired meaning fields. Under this framework, the situated PCFG grammar we used for generation can be described as a 6-tuple:

$$G = \langle N_p, N_c, T, S, L, \lambda \rangle \tag{1}$$

where N_p is a finite set of non-terminal symbols inheriting from a phrase structure parser, N_c is a finite set of concept symbols corresponding with record fields, T is a finite set of NL terminal symbols (words), $S \in Np$ is a distinguished start symbol, L is a lexicon which consists of a finite set of production rules, and λ is a set of parameters that define a probability distribution over derivations under G.

3.2 Grammar Induction

In this section, we present a learning procedure for the proposed grammar described above. The input to the learning algorithm is a set of training sentences paired with their correct meaning representations (as illustrated in Table 1). The output from the learning algorithm is a PCFG describing both phrase and concept embeddings. The learning algorithm assumes that a phrase structure parser is available, but it does not require any prior knowledge of the MR syntax.

To describe the grammar learning procedure, we start with an example. Consider the NL sentence in Table 1(a). We first analyze its phrase structure using a syntactic parser whose non-terminals are syntactic categories (e.g., NP, VP and QP) and part-of-speech tags (e.g., PN, DEG and NN). The parser we used for GIVE and SDS are both the Stanford Parser[3]. Figure 1(a) outlines the partial parser tree of sentence in Table 1(a).

The meaning of the sentence is then integrated by adding conceptual symbols of its subparts into the parser tree. Figure 1 (b) shows a hybrid parse tree of Figure 1 (a). Here the non-terminal symbols in bold, BUDGET, VAL1 and VAL2, represent domain-specific concepts corresponding to fields *budget*, *value*1 and *value*2. To get the hybrid parse tree, we first align phrases in the NL with the actual MR fields mentioned using the model of (Liang, et al., 2009) which is learned in an unsupervised manner using EM to produce which words in the

[3] http://nlp.stanford.edu/software/lex-parser.shtml

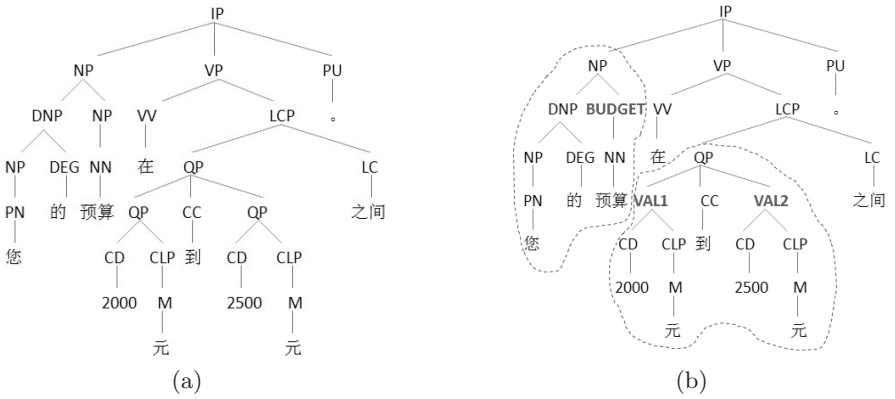

Fig. 1. Example of (a) a syntactic tree and (b) its corresponding hybrid tree from which the situated PCFG defined in Formula (1) is constructed. The subtree circled by dotted line contains conceptual node and its terminal derivations.

text were spanned by the fields. The aligned pairs are recorded in a temporary dictionary. Then for each phrase in the dictionary, we find the minimal subtree spanning it, and modify its ancestor node attached directly below the subtree's root node to the conceptual symbol of its aligned field. All ancestor nodes keep unchanged for phrases not in the alignment dictionary. The central characteristic of a tree structured representation is that component concept appears as a node in a tree, with its word realizations as terminal nodes derived by it. For example, the concept BUDGET has a terminal node "预算 (budget)", and VALUE1 "2,000元(2,000 yuan)", these could then form part of the representation for the sentence "您的预算在2,000元到2,500元之间。(Your budget is between 2,000 yuan and 2,500 yuan.)" The use of a recursive hybrid syntactic and conceptual structure is one characteristic that distinguishes the proposed grammar from earlier work in which meaning is represented by logical forms or regular grammars (Lu and Ng, 2011; Konstas and Lapata, 2013).

Given hybrid trees, N_p, N_c, T, S and the set of derivations that are possible are fixed, we only need to learn a probabilistic model parameterized by λ. Since the correct correspondence between NL words and MR fields is fully accessible, i.e., there is a single deterministic derivation associated with each training instance, model parameter λ can be directly estimated from the training corpus by counting. Because the derived trees output by parser can be noisy, we need to process them to obtain cleaner PCFG rules. We compare the 3-best trees produced by the Stanford Parser, and prune off inconsistent components voted by majorities when extracting and counting rules.

3.3 Decoding

Our goal in decoding is to find the most probable sentence $\hat{s}$ for a given meaning expression m:

$$\hat{s} = g(\underset{D \ s.t. \ m(D)=m}{\operatorname{argmax}} P(D|G) \cdot \ln(|D|+1)) \tag{2}$$

where g is a function that takes as input a derivation tree D and returns $\hat{s}$, $m(D)$ refers to the meaning representation of a derivation D, and $P(D|G)$ is product of weights of the PCFG rules used in a derivation D, the factor $\ln(|D|+1)$, offers a way to compensate the output sentence length $|D|$.

A conventional CKY-style decoder (Kasami, 1965; Younger, 1967) is not applicable to this work since the fields of MR do not exhibit a linear structure. We use a basic decoding introduced in (Konstas and Lapata, 2013) which is essentially a bottom-up chart-parsing algorithm. It first fills the diagonal cell of the chart with the top scoring words emitted by the unary productions of the type $A \rightarrow \alpha$, where A is a non-terminal symbol, and α is a terminal word. The extracted grammar is binarized such that decoding takes cubic time with respect to the sentence length.

In order to search among exponentially many possible generations for a given input, it would be preferable if we added to the chart a list of the top k words and production rules, and thus produced a k-best list of derivations at the root node, yielding k-best sentences at the leaf nodes. We do this k-best decoding using the lazy algorithm introduced in (Huang and Chiang, 2005) which delays the whole k-best calculation until after parsing. Then an external language model can be applied to rerank the k-best derivations. We examine two ways of intersecting language model, one is to rerank directly the k-best sentences after those sentences are generated (Langkilde and Knight, 1998), the other is to rerank derived partial trees in a timely manner with cube pruning (Huang and Chiang, 2005; Konstas and Lapata, 2013).

4 Empirical Evaluation

This section presents our experimental setup for assessing the performance of our model. We give details on our dataset, model parameters, the metric used for comparison and experimental results.

4.1 Data Set

We conducted experiments on a Chinese spoken dialogue system (SDS) for meeting room booking. Our NLG module receives structured input from dialogue management (DM) module and generates natural language response to user. The structured input includes dialogue actions (e.g., greet, request, confirm), objects (e.g., date, budget, location) and object values which can be a null. DM delivers to NLG at most two actions at a time. The SDS corpus consists of 1,406

formal meaning representations, along with their Chinese NL expressions written by 3 Chinese native speakers. The average sentence length for the 1,406-example data set is 15.7 Chinese words. We randomly select 1,000 record pairs as training data, and the remaining 406 as testing data.

In order to assess the generation performance across different domains and different languages, we conducted experiments on the platform provided by the Challenge on Generating Instructions in Virtual Environments (GIVE), a theory-neutral, end-to-end evaluation effort for NLG systems. The NLG model generates a sequence of English NL instructions guiding users performing a "treasure hunt" task in a virtual 3D environment. We obtain 63 American English written discourses in which one subject guided another in a treasure hunting task in the spirit of the GIVE-2 virtual worlds. In order to ensure quality of the gold data, we preprocessed the corpus to delete non-sense instructions (i.e., sentences not related with any environmental parameters or operation instructions, e.g., "lol.", "what?"), correct spelling mistakes and tokenize the abbreviations (e.g., "srry", "u"). Finally, 1,159 NL and MR pairs from 50 discourses are used for training, and 294 pairs from the remaining 13 discourses for testing. The average sentence length for the 1,453 sentences is 7.8 English words.

4.2 Evaluation Metric

To evaluate the quality of the generated sentences, the BLEU score (Papineni et al., 2002) is computed by comparing system-generated sentences with human-written sentences. Specifically, the BLEU score is the geometric mean of the precision of n-grams of various lengths, multiplied by a brevity penalty factor that penalizes candidate sentences shorter than the reference sentences. BLEU has a fairly good agreement with human judgment and has been used to evaluate a variety of language generation systems (Angeli et al., 2010; Konstas and Lapata, 2012).

In addition, we evaluated the generated text via a human judgment as designed in (Angeli et al., 2010). Human evaluators were presented with a meaning representation and were asked to rate its corresponding NL expression along two dimensions: fluency (is the text grammatical and overall understandable?) and semantic correctness (does the meaning conveyed by the NL sentences correspond to meaning representations?). Human evaluators used a five point rating scale where a high number indicates better performance. The averaged score of three difference human evaluators was computed.

4.3 Results

In order to compare our work with previous related work, we implement the method of (Konstas and Lapata, 2013) on our datasets. The BLEU scores of different systems are summarized in Table 2.

Table 2 compares BLEU scores achieved using the situated grammar described in Section 3.1 and 3.2 with that using the grammar described in (Konstas and Lapata, 2013). 1-BEST signifies results obtained from the basic decoder.

Table 2. BLEU scores on SDS and GIVE.

system	SDS	GIVE
1-BEST-Konstas	9.32	10.49
k-BEST-Konstas	19.14	21.70
k-BEST-LM-Konstas	21.85	24.26
1-BEST-Our	30.88	31.07
k-BEST-Our	31.82	30.26
k-BEST-LM-Our	31.96	31.21

k-BEST and k-BEST-LM are results obtained respectively from reranking after generation and reranking during generation described in Section 3.3. Here we set k=20 without more fine-tuning work. Since the training data and the sentence length of our applications are relatively small, we used a bigram language model with add-one smoothing. Regarding these results, one point should be noted that the sentence length N is not restricted as a fixed number, while varying from 1 to a length of the longest sentence in the training data. The sentences with different length are overall sorted to obtain the 1-BEST and the k-BEST.

From Table 2, we find that differences in BLEU scores between 1-BEST-Konstas and 1-BEST-Our are statistically significant (9.32 vs. 30.88 in SDS domain, and 10.49 vs. 31.07 in GIVE domain). Since the only difference between these two results is the grammars used, we have reason to justify that the situated grammar learnt from the phrase-concept-hybrid trees is superior for modeling NL and MR correspondence to that used in (Konstas and Lapata, 2013).

It is interesting to notice that k-BEST-Konstas observes substantial increase in performance compared to 1-BEST-Konstas in both two domains, while k-BEST-Our only achieves a slight increase compared to 1-BEST-Our. The same observation also happens for the timely reranking k-BEST-LM-Konstas and k-BEST-LM-Our. As reported in (Konstas and Lapata, 2013), statistical language model offers potentially significant advantages for the sequential Markov grammar. Meanwhile, these results verify the robustness of the proposed method. Another major advantage of our method over method of (Konstas and Lapata, 2013) is that it does not require any prior knowledge of the MR syntax for training. Therefore, transplanting our method to other NLG application is relatively easy.

Overall, k-BEST-LM performs better than k-BEST, but the improvement is moderate. In practice, k-BEST-LM is more commonly used due to its computational efficiency and integrity.

Table 3 shows the human ratings for each system and the gold-standard human-authored sentences. On both Chinese and English domains our system is significantly better than the 1-BEST-Konstas baseline in terms of grammatical coherence and semantic soundness.

In order to evaluate the quality of our generation system in a practical view, we implement our English NLG system on the GIVE platform.

10 undergraduate volunteers are enrolled to play the "treasure hunt" game following the NL instructions our system generated. We collected 103 turns of game in total, 58 turns are guided by instructions produced by 1-BEST-Our, 45 turns by k-BEST-LM-Our.

Table 3. Human ratings for syntactic fluency (SF) and semantic correctness (SC).

system	SDS		GIVE	
	SF	SC	SF	SC
1-BEST-Konstas	2.29	1.94	2.04	2.47
k-BEST-LM-Konstas	3.91	3.12	3.82	3.49
1-BEST-Our	4.36	3.95	4.02	3.88
k-BEST-LM-Our	4.34	4.33	4.22	4.05
HUMAN	4.76	4.89	4.58	4.13

GIVE challenge adopts several objective metrics so as to measure the success of instructions in a situated interaction scenario. Results of the objective metrics can be induced automatically from log files. Table 4 outlines objective metrics used in GIVE-2.5 challenge (Striegnitz, et al., 2011).

Table 4. Objective measures used in GIVE-2.5 challenge.

binary task success: Percent of the player get the trophy.
duration: Time in seconds from the end of the tutorial until retrieval of the trophy.
instructions: Number of instructions produced by the NLG system.
words: Number of words used by the NLG system.

Table 5 shows the comparison of our systems with the work of (Dethlefs and Cuayahuitl, 2014) which is one of state-of-the-art systems evaluated on GIVE challenge. In terms of task success, k-BEST-LM-Our outperforms 1-BEST-Our by 7%, while both of them are less than that of Dethlefs. But it is a consensus that, besides the quality of generated instructions, there are many other subjective factors influencing the task success rate, for example, whether the player is a native English speaker, to what content the player is familiar with the game.

It is worthy to notice that both 1-BEST-Our and k-BEST-LM-Our generate significantly less interactions which guarantees much shorter interaction time to finish a task. Averagely, 1-BEST-Our generates 10.4 words per instruction and k-BEST-LM-Our 14.0 words, while Dethlefs's system generates 9.8 words for each instruction. Our systems produce many instructions such as "click the green button in front of you to the left of the lamp", "turn right into the room ahead". In contrast, as reported (Striegnitz et al., 2011; Dethlefs and Cuayahuitl, 2014), most other GIVE systems output much shorter instructions such as

"click green", "turn right". Although brief enough, such instructions will lead more false actions due to lack of necessary reference information.

Table 5. Objective metrics for our systems compared with systems of (Dethlefs and Cuayahuitl, 2014).

metric	Dethlefs	1-BEST-Our	k-BEST-LM-Our
binary task success	0.80	0.67	0.74
duration	700	261	491
instructions	312.3	105.3	160.7
words	3075.6	1093	2249

5 Conclusions

We have presented a PCFG-based natural language generation method. In particular, the method learns situated PCFG rules from hybrid phrase-concept trees automatically augmented from the output of an existing syntactic parser. A compelling advantage of the proposed method is that it does not rely on prior knowledge of the MR syntax for training. We have shown the competitive results across different application domains in both Chinese and English language. Future extensions include deploying more efficient decoding algorithms, and richer structural features to rerank the derivations.

Acknowledgments. This work was partially supported by Natural Science Foundation of China (No. 61202248, No. 61273365), Discipline Building Planing 111 Base Fund (No. B08004) and Engineering Research Center of Information Networks, Ministry of Education.

References

1. Langkilde, I., Knight, K.: Generation that exploits corpus based statistical knowledge. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 704–710 (1998)
2. Reiter, E., Dale, R.: Building natural language generation systems. Cambridge University Press, New York (2000)
3. Walker, M.A., Rambow, O., Rogati, M.: Training a sentence planner for spoken dialogue using boosting. Computer Speech and Language **16**(3–4), 409–433 (2002)
4. Angeli, G., Liang, P., Klein, D.: A simple domain-independent probabilistic approach to generation. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, pp. 502–512 (2010)
5. Kim, J., Mooney, R.: Generative alignment and semantic parsing for learning from ambiguous supervision. In: Proceedings of the 23rd Conference on Computational Linguistics, Beijing, China, pp. 543–551 (2010)

6. Konstas, I., Lapata, M.: Concept-to-text generation via discriminative reranking. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Jeju, South Korea, pp. 369–378 (2012)
7. Konstas, I., Lapata, M.: A Global Model for Concept-to-Text Generation. Journal of Artificial Intelligence Research **48**(2013), 305–346 (2013)
8. Ratnaparkhi, A.: Trainable Approaches to Surface Natural Language Generation and Their Application to Conversational Dialog Systems. Computer Speech and Language **16**(3–4), 435–455 (2002)
9. Rieser, E., Lemon, O.: Natural language generation as planning under uncertainty for spoken dialogue systems. In: Proceedings of the 12th Conference of the European Chapter of the ACL, Athens, Greece, pp. 683–691 (2009)
10. Huang, L., Chiang, D.: Better k-best parsing. In: Proceedings of the 9th International Workshop on Parsing Technology, Vancouver, British Columbia, pp. 53–64 (2005)
11. Liang, P., Jordan, M., Klein, D.: Learning semantic correspondences with less supervision. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore, pp. 91–99 (2009)
12. Lu, W., Ng, H.T.: A probabilistic forest-to-string model for language generation from typed lambda calculus expressions. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK, pp. 1611–1622 (2011)
13. Wong, Y.W., Mooney, R.: Generation by inverting a semantic parser that uses statistical machine translation. In: Proceedings of the Human Language Technology and the Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, NY, pp. 172–179 (2007)
14. McKinley, N., Ray, S.: A decision-theoretic approach to natural language generation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, USA, pp. 552–561 (2014)
15. Dethlefs, N., Cuayahuitl, H.: Hierarchical reinforcement learning for situated natural language generation. Natural Language Engineering **21**(03), 391–435 (2014)
16. Belz, A.: Automatic Generation of Weather Forecast Texts Using Comprehensive Probabilistic Generation-Space Models. Natural Language Engineering **14**(4), 431–455 (2008)
17. Belz, A., Kow, E.: System building cost vs. output quality in data-to-text generation. In: Proceedings of the 12th European Workshop on Natural Language Generation, Athens, Greece, pp. 16–24 (2009)
18. Gargett, A., Garoufi, K., Koller, A., Striegnitz K.: The GIVE-2 corpus of giving instructions in virtual environments. In: Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC), Valletta, Malta (2010)
19. Striegnitz, K., Denis, A., Gargett, A., Garoufi, K., Koller, A., Theune, M.: Report on the second challenge on generating instructions in virtual environments (GIVE-2.5). In: Proceedings of the 13th European Workshop on Natural Language Generation (ENLG), Nancy, France, pp. 270–279 (2011)
20. Chen, Q., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Lan-guage Processing (EMNLP), Doha, Qatar, pp. 740–750 (2014)
21. Levy, R., Manning, C.D.: Is it harder to parse Chinese, or the Chinese Tree-bank? In: Proceedings of the ACL 2003, Sapporo, Japan, pp. 439–44 (2003)

22. Kasami, T.: An efficient recognition and syntax analysis algorithm for context-free languages. Tech. rep. AFCRL-65-758, Air Force Cambridge Research Lab, Bedford, Mas-sachusetts (1965)
23. Younger, D.H.: Recognition and parsing for context-free languages in time n3. Information and Control **10**(2), 189–208 (1967)
24. Papineni K., Roukos S., Ward, T., Zhu, W.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, pp. 311–318 (2002)
25. Benotti, L., Denis, A.: Giving instructions in virtual environments by corpus-based selection. In: Proceedings of the 12th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Portland, Oregon, pp. 68–77 (2011)