WDS'13 Proceedings of Contributed Papers, Part I, 12–17, 2013. ISBN 978-80-7378-250-4 © MATFYZPRESS

Towards a Truly Statistical Natural Language Generator for Spoken Dialogues

O. Dušek
Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics, Prague, Czech Republic.

Abstract. We present an introduction to the problem of Natural Language Generation (NLG) and give a brief survey of recent advances in this field, focusing on the usage of statistical methods. Most stress is put on NLG within Spoken Dialogue Systems. The paper concludes with an overview of problems posed by our planned development of a multilingual NLG system for a Spoken Dialogue System.

Introduction

NLG systems are used in many different areas: Spoken Dialogue Systems, Machine Translation, generation of short texts, such as weather reports or customer recommendations, Summarization or Question Answering. It is generally desirable that the systems be fast and simple, but at the same time provide some variation in the output (so as not to repeat the same sentences too often) and adapt easily to new areas of usage.

This paper is concerned with the adoption of statistical methods in NLG. We give an overview of the NLG problem in general and list the reasons for adopting statistical methods in this field. We then present a survey of recent statistical approaches to NLG and conclude with a proposal of a statistical NLG system for spoken dialogues.

The Problem of Natural Language Generation

The task of an NLG system is to create a natural language string that is "well-formed" and "human-like," i.e. grammatically correct and fluent, given some input data and a communication goal (e.g. to describe the data or have the user react in a specific way).

We first describe how a prototypical NLG system works and then briefly summarize the approaches implemented in current systems.

The Standard NLG Pipeline

The standard NLG pipeline as presented in a textbook [Reiter and Dale, 2000], which is at least partially reflected in most today's NLG systems, consists of two main steps: *content planning* (or *text planning*) and *surface realization*.

The task of the *content planning* step is, given input data and communication goal, to select content appropriate for output and perform basic structuring and ordering. The content planning step thus decides "what to say." Its intermediate output is the text plan, usually a tree-like data structure.

The second phase, *surface realization*, is then concerned with the question "how to say it." It consists of two sub-steps. First of them, *microplanning* or *sentence planning*, performs a detailed structuring of the content: aggregation of simple propositions into more complex sentences, choice of lexical and syntactic means, and decisions on referring expressions (e.g. pronouns). Its output, the so-called sentence plan(s), is then fed into the second sub-step, the *surface realization (proper)*. This last part of the whole NLG pipeline then linearizes the tree-shaped sentence plan into a natural language text and ensures that the output is grammatically correct.

Real Natural Language Generation Systems

While most existing NLG systems adhere to the standard NLG pipeline, very few of them concentrate on implementing it as a whole. Many generators, especially in Spoken Dialogue Systems, focus mainly on the content planning phase while using a simplistic surface realizer. Other NLG systems implement only the surface realization phase and expect a detailed sentence plan on the input. Some NLG systems are concerned only with ordering a given bag-of-words to produce a meaningful sentence.

The input data representation as well as all intermediate data structures in the pipeline vary greatly from system to system, which makes parts of different NLG systems generally incompatible. This is a direct consequence of their different intended usages.

Most systems are based on procedural rules, filling-in slots in templates, or grammars in various formalisms, such as Functional Unification Grammar [Elhadad and Robin, 1996] or Combinatory Categorial Grammar [White and Baldridge, 2003]. Newer systems, mostly from after 2000, introduce statistical methods into some parts of the generation process (see below).

Introducing Statistical Methods to Generation

NLG is one of the last areas in Natural Language Processing (NLP) to adopt statistical methods. There are several reasons for this, most of which are related to the position of NLG at the end of any NLP pipeline.

The advantage of rule-based methods is that their implementation is usually relatively simple, i.e. the generation is very fast and can be adjusted in a straightforward way. One has the ability to directly control the output: customize it for the given domain and ensure grammatical correctness. Since most NLG systems are deployed in a restricted setting, limited capabilities and lack of variation do not necessarily pose a significant problem. On the other hand, if a general-domain rule-based surface realizer is implemented for the given target language, it can be reused for new systems requiring this language.

Nevertheless, there are still important reasons for introducing statistical methods to NLG. Rule-based methods can become very difficult to maintain for more complicated domains or in a general-domain setting. Statistical methods are also much more easily adaptable to new domains: one only needs to retrain them using new training data. Moreover, they tend to be more robust to unexpected inputs than rule-based methods. The usage of statistical methods also promises to add more variation and, possibly, more naturalness to the output.

Trainable Content Planners

We now list the three most important approaches to trainable content planning: user models, overgeneration and ranking, and reinforcement learning.

User Models

One of the first attempts to introduce adaptivity into content planning was the introduction of user models [Moore et al., 2004; Walker et al., 2004; Carenini and Moore, 2006]. The content planners remained rule-based, but allowed the user to specify their preferences regarding the output by answering a set of simple questions. This approach was applied e.g. in a restaurant recommendation system or a flight information system. The user's answers controlled the importance of attributes such as flight duration or food quality.

Overgeneration and Ranking

Perhaps the first truly trainable content planning method was the overgeneration and ranking approach of Walker et al. [2001]. It used a rule-based sentence plan generator working with clause combining operations that would randomly sample several possible outputs, which were then fed into a statistical reranker trained on hand-annotated sentence plans. The reranker would then select the best variant.

This setting is adaptive and provides variation in outputs, but still requires a handcrafted module and the generation of many variants is rather computationally inefficient.

Trainable Content Planning: Reinforcement Learning

Recent content planners [Lemon, 2008; Rieser and Lemon, 2010] employ Reinforcement Learning to find the optimal presentation strategy. The generation is re-cast as a Markov Decision Process in a space of states connected with low-level NLG actions, such as summarizing database results or recommending the best result, that compose into a single utterance. The generator must plan a sequence of these actions to achieve a communicative goal, such as letting the user choose one of the presented results, in as few steps as possible. A Markov Decision Process is used to guide the decisions of the generator.

Trainable Surface Realizers

Statistical approaches to surface realization found in literature can be divided into three main groups: overgeneration and ranking, parameter optimization, and fully statistical realizers.

Overgeneration and Ranking

Same as in content planning, the first introduction of statistical methods into surface realization was the overgeneration and ranking approach. It requires a handcrafted realizer whose input is underspecified to make more output variants possible. The generated variants are then reranked and the best one selected.

Various reranking criteria have been tested, such as n-gram language models [Langkilde and Knight, 1998; Langkilde-Geary, 2002], tree models [Bangalore and Rambow, 2000], expected text-to-speech conversion quality [Nakatsu and White, 2006] or desired personality traits, such as extroversion, and expression alignment with the dialogue counterpart [Isard et al., 2006].

This approach achieved similar results as in content planning: greater variance, but at a higher computational cost.

Parameter Optimization

The parameter optimization approach to trainable surface realizers does not require overgeneration. However, the systems still need a handcrafted realizer which must furthermore have a set of adjustable parameters.

Paiva and Evans [2005] annotate desired linguistic features in a text corpus generated under many realizer parameter settings and connected them to desired parameter values via correlation analysis. Mairesse and Walker [2008] use machine learning to find generator parameters for various personality traits.

Statistical Surface Realizers

Some recent surface realizers do not require any handcrafted components. Most of them use techniques from Statistical Machine Translation (SMT): they translate from a formal language (semantic description) to a natural language. Wong and Mooney [2007] experiment with phrase-based SMT methods as well as with synchronous Context-Free Grammars using an inverted semantic parser. Lu et al. [2009] apply hybrid trees that combine semantic representations with natural language phrases in a tree structure.

Bohnet et al. [2010] employ a pipeline model based on the Meaning-Text Theory syntax. Using Support Vector Machines, it consecutively generates surface syntax trees from semantic trees, then proceeds to linearizing the output and generating morphology.

Fully Statistical Natural Language Generators

Recent works also describe fully statistical approaches to the whole NLG process. They are based on supervised learning and a hierarchical phrase-based approach. Mairesse et al. [2010] use Bayesian Networks for decoding and a stack-based hierarchy for data representation. Angeli et al. [2010] employ a three-level hierarchy of records, fields, and templates and generate sequences using a log-linear discriminative model.

However, both systems are only applicable to very limited domains and to languages with little morphology as they require a very detailed training data annotation and operate directly on word forms.

Towards a Multilingual Statistical Generator

Most works in NLG focus on a single target language. In doing that, many generators take the advantage of the lack of word inflection in languages such as English. We propose a novel approach to NLG that uses mostly statistical methods, yet is able to handle languages with complex morphology, such as Czech, and can be easily adapted for different domains and/or languages. We first summarize prior work on Czech NLG, then present our current experiments and a proposal of a future statistical NLG system.

Prior Works in Generation for Czech

A Czech NLG system based on the Functional Generative Description [FGD, Sgall et al., 1986] has been developed by Ptáček and Žabokrtský [2006]. It is a general domain rule-based surface realizer, which uses semantic, or tectogrammatical, tree structures as input. These structures (see Figure 1) contain content words as nodes along with their semantic functions, *functors*, such as `Predicate`, `Actor`, or `Time adverbial` (shown in green in Figure 1).

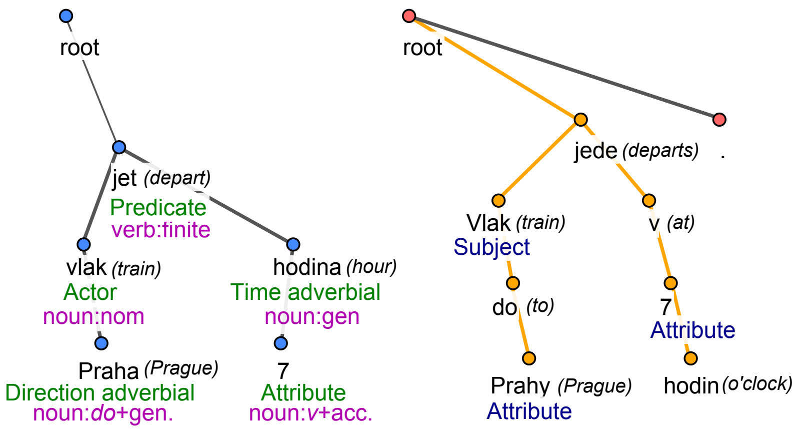

Figure 1. Tectogrammatic semantic tree based on Functional Generative Description (left) and a corresponding surface dependency tree (right)

Ptáček and Žabokrtský [2006] implemented a rule-based module to select the target syntactic shape (called *formemes*, purple in Figure 1), which they then used to construct a surface-syntactic tree. Correctly inflected word forms were predicted using a dictionary [Hajič, 2004] and the tree was linearized to produce the output text. Žabokrtský et al. [2008] employed the NLG module in a structural Machine Translation scenario, where formemes were determined directly by the translation model.

Our Experiments in a Spoken Dialogue Scenario

Building upon the works mentioned in the preceding section, we implemented a mostly rule-based surface realization module for the ALEX Spoken Dialogue System[1] and combined it with a template-based content planner. In addition to rule-based syntactic modules adapted from Žabokrtský et al. [2008], our realizer uses a statistical morphology generation module [Dušek and Jurčíček, 2013] to avoid dictionaries and to increase adaptivity. It is also able to intermix preset templates with parts of tectogrammatical representation to be generated. A template for our NLG system is shown in Figure 2.

```
Vlak [Praha|noun:do+gen|gender:fem,number:sg] jede
v [[7|adj:attr] hodina|noun:acc|gender:fem,number:pl].
```

Figure 2. A template with parts of tectogrammatical trees used in our setup (the sentence corresponds to the trees in Figure 1).

Planned Approach

We plan to gradually improve our NLG module up to an almost full adoption of statistical methods. We require a simple adaptation to new domains, variance (no fixed templates), and multilingualism (Czech and English at least).

As the FGD-based representation proved to be useful in surface realization, we intend to employ it in our system, possibly in a simplified form. However, we will also consider the benefits of other possible formalisms. We intend to develop a fully statistical content planner, probably inspired by MT-based methods previously used for surface realization (see "Statistical Surface Realizers"). The surface realizer will mix rule-based and statistical approaches as many grammar rules can be easily learned from corpora.

Acknowledgments. This research was partly funded by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221 and core research funding of Charles University in Prague.

References

Angeli, G., Liang, P., and Klein, D., A simple domain-independent probabilistic approach to generation, in *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, p. 502–512, 2010.

Bangalore, S. and Rambow, O., Exploiting a probabilistic hierarchical model for generation, in *Proc. of the 18th conference on Computational linguistics-Volume 1*, p. 42–48, 2000.

Bohnet, B., Wanner, L., Mille, S., and Burga, A., Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer, in *Proc. of the 23rd International Conference on Computational Linguistics*, p. 98–106, 2010.

Carenini, G. and Moore, J. D., Generating and evaluating evaluative arguments, *Artificial Intelligence*, *170*, 925–952, 2006.

Dušek, O. and Jurčíček, F., Robust multilingual statistical morphological generation models, in *Proc. of the ACL Student Research Workshop*, p. 158–164, 2013.

Elhadad, M. and Robin, J., An overview of SURGE: a reusable comprehensive syntactic realization component, Tech. rep., Dept. of Computer Science, Ben Gurion University, Beersheba, 1996.

Hajič, J., *Disambiguation of rich inflection: computational morphology of Czech*, Karolinum, 2004.

Isard, A., Brockmann, C., and Oberlander, J., Individuality and alignment in generated dialogues, in *Proc. of the Fourth International Natural Language Generation Conference*, p. 25–32, ACL, Stroudsburg, PA, USA, 2006.

Langkilde, I. and Knight, K., Generation that exploits corpus-based statistical knowledge, in *Proc. of the 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics-Volume 1*, p. 704–710, 1998.

[1] `https://sites.google.com/site/filipjurcicek/projects/vystadial`

Langkilde-Geary, I., An empirical verification of coverage and correctness for a general-purpose sentence generator, in *Proc. of the 12th International Natural Language Generation Workshop*, p. 17–24, 2002.

Lemon, O., Adaptive natural language generation in dialogue using reinforcement learning, *Proc. of SEMdial*, 2008.

Lu, W., Ng, H. T., and Lee, W. S., Natural language generation with tree conditional random fields, in *Proc. of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, p. 400–409, 2009.

Mairesse, F. and Walker, M., Trainable generation of big-five personality styles through data-driven parameter estimation, in *Proc. of the 46th Annual Meeting of the ACL (ACL)*, p. 165–173, 2008.

Mairesse, F., Gašić, M., Jurčíček, F., Keizer, S., Thomson, B., Yu, K., and Young, S., Phrase-based statistical language generation using graphical models and active learning, in *Proc. of the 48th Annual Meeting of the ACL*, p. 1552–1561, 2010.

Moore, J., Foster, M. E., Lemon, O., and White, M., Generating tailored, comparative descriptions in spoken dialogue, in *Proc. of FLAIRS*, 2004.

Nakatsu, C. and White, M., Learning to say it well: reranking realizations by predicted synthesis quality, in *Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, p. 1113–1120, ACL, Stroudsburg, PA, USA, 2006.

Paiva, D. S. and Evans, R., Empirically-based control of natural language generation, in *Proc. of the 43rd Annual Meeting of ACL*, p. 58–65, ACL, Stroudsburg, PA, USA, 2005.

Ptáček, J. and Žabokrtský, Z., Synthesis of Czech sentences from tectogrammatical trees, 2006.

Reiter, E. and Dale, R., *Building Natural Language Generation Systems*, Cambridge Univ. Press, 2000.

Rieser, V. and Lemon, O., Natural language generation as planning under uncertainty for spoken dialogue systems, in *Empirical methods in natural language generation*, p. 105–120, 2010.

Sgall, P., Hajičová, E., and Panevová, J., *The meaning of the sentence in its semantic and pragmatic aspects*, D. Reidel, Dordrecht, 1986.

Walker, M., Whittaker, S., Stent, A., Maloor, P., Moore, J., Johnston, M., and Vasireddy, G., Generation and evaluation of user tailored responses in multimodal dialogue, *Cognitive Science*, *28*, 811–840, 2004.

Walker, M. A., Rambow, O., and Rogati, M., SPoT: a trainable sentence planner, in *Proc. of 2nd meeting of NAACL*, p. 1–8, ACL, Stroudsburg, PA, USA, 2001.

White, M. and Baldridge, J., Adapting chart realization to CCG, in *Proc. of the 9th European Workshop on Natural Language Generation*, p. 119–126, 2003.

Wong, Y. W. and Mooney, R. J., Generation by inverting a semantic parser that uses statistical machine translation, in *Proc. of Human Language Technologies: The Conference of the North American Chapter of the ACL (NAACL-HLT-07)*, p. 172–179, 2007.

Žabokrtský, Z., Ptáček, J., and Pajas, P., TectoMT: highly modular MT system with tectogrammatics used as transfer layer, in *Proc. of the Third Workshop on Statistical Machine Translation*, p. 167–170, ACL, 2008.