Towards an Optimal Lexicalization in a Natural-Sounding Portable Natural Language Generator for Dialog Systems

Inge M. R. De Bleecker
Department of Linguistics
The University of Texas at Austin
Austin, TX 78712, USA
imrdb@mail.utexas.edu

Abstract

In contrast to the latest progress in speech recognition, the state-of-the-art in natural language generation for spoken language dialog systems is lagging behind. The core dialog managers are now more sophisticated; and natural-sounding and flexible output is expected, but not achieved with current simple techniques such as template-based systems. Portability of systems across subject domains and languages is another increasingly important requirement in dialog systems. This paper presents an outline of LEGEND, a system that is both portable and generates natural-sounding output. This goal is achieved through the novel use of existing lexical resources such as FrameNet and WordNet.

1 Introduction

Most of the natural language generation (NLG) components in current dialog systems are implemented through the use of simple techniques such as a library of hand-crafted and pre-recorded utterances, or a template-based system where the templates contain slots in which different values can be inserted. These techniques are unmanageable if the dialog system aims to provide variable, natural-sounding output, because the number of pre-recorded strings or different templates becomes very large (Theune, 2003). These techniques also make it difficult to port the system into another subject domain or language.

In order to be widely successful, natural language generation components of future dialog systems need to provide natural-sounding output while being relatively easy to port. This can be achieved by developing more sophisticated techniques based on concepts from deep linguistically-based NLG and text generation, and through the use of existing resources that facilitate both the natural-sounding and the portability requirement.

We might wonder what exactly it means for a computer to generate 'natural-sounding' output. Computer-generated natural-sounding output should not mimic the output a human would construct, because spontaneous human dialog tends to be teeming with disfluencies, interruptions, syntactically incorrect and incomplete sentences among others (Zue, 1997). Furthermore, Oberlander (1998) points out that humans do not always take the most efficient route in their reasoning and communication. These observations lead us to define natural-sounding computer-generated output to consist of utterances that are free of disfluencies and interruptions, and where complete and syntactically correct sentences convey the meaning in a concise yet clear manner.

Secondly we can define the portability requirement to include both domain and language independence. Domain-independence suggests that the system must be easily portable between different domains, while language-independence requires that the system must be able to accommodate a new natural language without any changes to the core components.

Section 2 of this paper explains some prerequisites, such as the NLG pipeline architecture our system is based on, and the FrameNet and Word-Net resources. Next an overview of the system ar-

chitecture and implementation, as well as an in-depth analysis of the lexicalization component are presented. Section 3 presents related work. Section 4 outlines a preliminary conclusion and lists some outstanding issues.

2 System Architecture

2.1 Three-Stage Pipeline Architecture

Our natural language generator architecture follows the three-stage pipeline architecture, as described in Reiter & Dale (2000). In this architecture, the generation component of a text generation system consists of the following subcomponents:

- The **document planner** determines what the actual content of the output will be on an abstract level and decides how pieces of content should be grouped together.

- The **microplanner** includes lexicalization, aggregation, and referring expression generation tasks.

- The **surface realizer** takes the information constructed by the microplanner and generates a syntactically correct sentence in a natural language.

2.2 Lexical Resources

The use of FrameNet and WordNet in our system is critical to its success. The FrameNet database (Baker et al., 1998) is a machine-readable lexico-graphic database which can be found at http://framenet.icsi.berkeley.edu/. It is based on the principles of Frame Semantics (Fillmore, 1985). The following quote explains the idea behind Frame Semantics: "The central idea of Frame Semantics is that word meanings must be described in relation to *semantic frames* – schematic repre-sentations of the conceptual structures and patterns of beliefs, practices, institutions, images, etc. that provide a foundation for meaningful interaction in a given speech community." (Fillmore et al., 2003, p. 235). In FrameNet, lexical units are grouped in frames; frame hierarchy information is provided for each frame, in combination with a list of se-mantically annotated corpus sentences and syntac-tic valence patterns.

WordNet is a lexical database that uses conceptual-semantic and lexical relations in order to group lexical items and link them to other groups (Fellbaum, 1998).

2.3 System Overview

Our system, called LEGEND (LExicalization in natural language GENeration for Dialog systems) adapts the pipeline architecture presented in section 2.1 by replacing the document planner with the dialog manager. This makes it more suitable for use in dialog systems, since the dialog manager decides on the actual content of the output in dialog systems. Figure 1 below shows an overview of our system architecture.
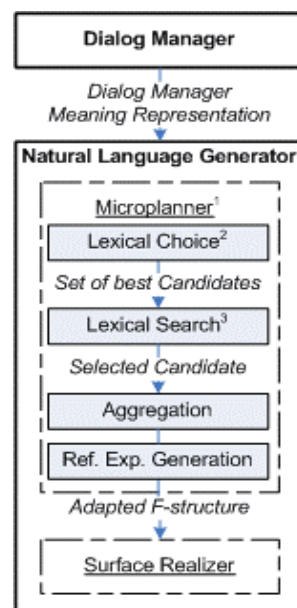
Figure 1. System Architecture

As figure 1 shows, the dialog manager provides the generator with a dialog manager meaning representation (DM MR), which contains the content information for the answer.

Our research focuses on the lexicalization sub-component of the microplanner (number 1 in fig-ure 1). Lexicalization is further divided into two processes: lexical choice and lexical search. Based on the DM MR, the lexical choice process (number 2 in figure 1) constructs a set of all potential output candidates. Section 2.5 describes the lexical choice process in detail. Lexical search (number 3 in fig-ure 1) consists of the decision algorithm that de-

cides which one of the set of possible candidates is most appropriate in any situation. Lexical search is also responsible for packaging up the most appropriate candidate information in an adapted F-structure, which is subsequently processed through aggregation and referring expression generation, and finally sent to the surface realizer. Section 2.6 describes the details of the lexical search process.

2.4 Implementation Details

Given time and resource constraints, our implementation will consist of a prototype (written in Python) of the lexical choice and lexical search processes only of the microplanner. We take a DM MR as our input. Aggregation and referring expression generation requirements are hard-coded for each example; algorithm development, identification and implementation for these modules is beyond the scope of this research.

Our system uses the LFG-based XLE system's generator component as a surface realizer. For more information, refer to Shemtov (1997) and Kaplan & Wedekind (2000).

2.5 Lexical Choice

The task of the lexical choice process is to take the meaning representation presented by the dialog manager (refer to figure 1), and to construct a set of output candidates. We will illustrate this by taking a simple example through the entire dialog system. The example question and answer are deliberately kept simple in order to focus on the workings of the system, rather than the specifics of the example.

Assume this is a dialog system that helps the consumer in buying camping equipment. The user says to the dialog system: "Where can I buy a tent?" The speech recognizer recognizes the utterance, and feeds this information to the parser. The semantic parser parses the input and builds the meaning representation shown in figure 2. The main event (main verb) is identified as the lexical item *buy*. The parser looks up this lexical item in FrameNet, and identifies it as belonging to the *commerce_buy* frame. This frame is defined in FrameNet as: "… describing a basic commercial transaction involving a buyer and a seller exchanging money and goods, taking the perspective of the buyer." (http://framenet.icsi.berkeley.edu/). All

other elements in the meaning representation are extracted from the input utterance.

> **Event**: buy
> **Frame**: commerce_buy
> **Query**: location
> **Agent**: 1st pers sing
> **Patient**: tent

Figure 2. Parser Meaning Representation

This meaning representation is then sent to the dialog manager. The dialog manager consults the domain model for help in the query resolution, and subsequently composes a meaning representation consisting of the answer to the user's question (figure 3). For our example, the domain model presents the query resolution as "Camping World", the name of a (fictitious) store selling tents. The DM MR also shows that the *Agent* and the *Patient* have been identified by their frame element names. This DM MR serves as the input to the microplanner, where the first task is that of lexical choice.

> **Event**: buy
> **Frame**: commerce_buy
> **Query Resolution**: place "Camping World"
> **Agent**: buyer (1st p.s. => 2nd p.s.)
> **Object**: goods ("tent")

Figure 3. Dialog Mgr Meaning Representation

In order to construct the set of output candidates, the lexical choice process mines the FrameNet and WordNet databases in order to find acceptable generation possibilities. This is done in several steps:

- In step 1, lexicalization variations of the main *Event* within the same frame are identified.

- Step 2 consists of the investigation of lexical variation in the frames that are one link away in the hierarchy, namely the frame the current frame inherits from, and the sub-frames, if any exist.

- Step 3 is concerned with special relations within FrameNet, such as the 'use'-relation The lexical variation within these frames is investigated.

We return to our example in figure 3 to clarify these 3 steps.

In step 1, appropriate lexical variation within the same frame is identified. This is done by listing all

lexical units of same syntactic category as the original word. The following verbs are lexical units in *commerce_buy*: *buy, lease, purchase, rent.* These verbs are not necessarily synonyms or near-synonyms of each other, but do belong to the same frame. In order to determine which of these lexical items are synonyms or near-synonyms, we turn to WordNet, and look at the entry for *buy*. The only lexical item that is also listed in one of the senses of *buy* is *purchase*. We thus conclude that *buy* and *purchase* are both good verb candidates.

Step 2 investigates the lexical items in the frames that are one link away from the *commerce_buy* frame. *Commerce_buy* inherits from *getting*, and has no subframes. The lexical items of the *getting* frame are listed. The lexical items of the *getting* frame are: *acquire, gain, get, obtain, secure.* For each entry, WordNet is consulted as a first pruning mechanism. This results in the following:

- Acquire: get

- Gain: acquire, win

- Get: acquire

- Obtain: get, find, receive, incur

- Secure: no items on the list

How exactly lexical choice determines that *get* and *acquire* are possible candidates, while the others are not (because they aren't suitable in the context in which we use them) is as of yet an open issue. It is also an open issue whether WordNet is the most appropriate resource to use for this goal; we must consider other options, such as Thesaurus, etc…

In step 3 we investigate the other relations that FrameNet presents. To date, we have only investigated the 'use relation'. Other relations available are the inchoative and causative relations. At this point, it is not entirely clear how those relations will prove to be of any value to our task. The *commerce_buy* frame uses *commerce_goods_transfer*, which is also used by *commerce_sell.* We find our frame elements *goods* and *buyer* in the *commerce_sell* frame as well. Lexical choice concludes that the use of the lexical items in this frame might be valuable and repeats step 1 on these lexical items.

After all 3 steps are completed, we assume our set of output candidates to be complete. The set of output candidates is presented to the lexical search process, whose task it is to choose the most appropriate candidate. For the example we have been using throughout this section, the set of output candidates is as follows:

- You can buy a tent at Camping World.

- You can purchase a tent at Camping World.

- You can get a tent at Camping World.

- You can acquire a tent at Camping World.

- Camping World sells tents.

As mentioned at the beginning of this section, this example is very simple. For this reason, one can definitely argue that the first 4 output possibilities could be constructed in much simpler ways than the method used here, e.g. by simply taking the question and making it an affirmative sentence through a simple rule. However, it should be pointed out that the last possibility on the list would not be covered by this simple method. While user studies would need to provide backup for this assumption, we feel that possibility 5 is a very good example of natural-sounding output, and thus proves our method to be valuable, even for simple examples.

2.6 Lexical Search

The set of output candidates for the example above contains 5 possibilities. The main task of the lexical search process is to choose the most optimal candidate, thus the most natural-sounding candidate (or at least one of the most natural-sounding candidates, if more than one candidate fits that criterion). There are a number of directions we can take for this implementation.

One option is to implement a rule-based system. Every output candidate is matched against the rules, and the most appropriate one comes out at the top. Problems with rule-based systems are well-known: they must be handcrafted, which is very time-consuming, constructing the rule base such that the desired rules fire in the desired circumstances is somewhat of a "black" art, and of course a rule base is highly domain-dependent. Extending and maintaining it is also a laborious effort.

Next we can look at a corpus-based technique. One suggestion is to construct a language model of the corpus data, and use this model to statistically

determine the most suitable candidate. Langkilde (2000) uses this approach. However, the main problem here is that one needs a large corpus in the domain of the application. Rambow (2001) agrees that most often, no suitable corpora are available for dialog system development.

Another possibility is to use machine learning to train the microplanner. Walker et al. (2002) use this approach in the SPOT sentence planner. Their ranker's main purpose is to choose between different aggregation possibilities. The authors suggest that many generation problems can successfully be treated as ranking problems. The advantage of this approach is that no domain-dependent hand-crafted rules need to be constructed, and no existence of a corpus is needed.

Our current research idea is somewhat related to option two. A relatively small domain-independent corpus of spoken dialogue is semi-automatically labeled with frames and semantic roles. For each frame, all the occurrences in the corpus are ordered according to their frequency for each separate valence pattern. This model is then used as a comparator for all output candidates, and the most optimal one (most frequent one) will be selected. This approach is currently not implemented; further work needs to determine the viability of the approach.

Independent of the method used to find the most suitable candidate, the output must be packaged up to be sent to the surface realizer. The XLE system expects a fairly detailed syntactic description of the utterance's argument structure. We construct this through the use of FrameNet and its valence pattern information. In returning to our example, let's assume the selected candidate is "Camping World sells tents." Its meaning representation is as follows:

> **Event**: sell
> **Frame**: commerce_sell
> **Seller**: Camping World
> **Goods**: tents

Figure 4. "Camping World sells tents."

FrameNet provides an overview of the frame elements a given frame requires ("core elements") and those that are optional ("peripheral elements"). For the *commerce_sell* frame, the two core elements are *Goods* and *Seller*. It also provides an overview of the valence patterns that were found in the annotated sentences for this frame. FrameNet

does not include frequency information for each annotation. We thus need to pick a valence pattern at random. One way of doing this is to find a pattern that includes all (both) frame elements in our utterance, and then use the (non-statistical) frequency information. Figure 5 shows that, for our example above, this results in:

> FE_Seller *sell* FE_goods
> With the following syntactic pattern:
> NP.Ext *sell* NP.Obj

No. Annotated	Patterns	
	Goods	Seller
3	--	NP.Ext
2	NP.Comp	NP.Ext
27	NP	--
4	NP.Ext	PP[by].Comp
27	NP.Obj	NP.Ext

Figure 5. Valence Patterns "commerce_sell"

Thus our output to the surface realizer indicates that the seller frame element fills the subject role and consists of an NP, while the goods frame element fills the object role and consists of an NP. Given this syntactic pattern information that we gather from FrameNet, we are able to construct an F-structure that is suitable as the input to the surface realizer.

3 Related Work

To date, only a limited amount of research has dealt with deep linguistically-based natural language generation for dialog systems. Theune (2003) presents an extensive overview of different NLG methods and systems. A number of stochastic-based generation efforts have been undertaken in recent years. These generators generally consist of an architecture similar to ours, in which first a set of possible candidates is constructed, followed by a decision process to choose the most appropriate output. Some examples are the Nitrogen system (Langkilde and Knight, 1998) and the SPoT trainable sentence planner (Walker et al., 2002).

4 Outlook and Future Work

We propose a novel approach to lexicalization in NLG in order to generate natural-sounding speech in a portable environment. The use of existing

lexical resources allows a system to be more portable across subject domains and languages, as long as those resources are available for the targeted domains and languages. FrameNet in particular allows us to generate multiple possibilities of natural-sounding output while WordNet helps in a first step to prune this set. FrameNet is further applied on an existing corpus to help with the final decision on choosing the most optimal candidate among the presented possibilities. The valence pattern information in FrameNet helps constructing the detailed syntactic pattern required by the surface realizer.

A number of issues need further consideration, including the following:

- lexical choice: investigation of semantic distances (step 2 of algorithm), use of WordNet and/or other resources for first-step pruning.

- lexical search: develop initial research ideas further and implement

- a user study to assess whether the goals of natural-sounding output and portability have successfully been fulfilled.

Furthermore, for this generator to be used in a real-life environment, the entire dialog system must be developed; for our research purposes, we have left out the construction of a semantic parser, the dialog manager, and an appropriate domain model. We have also not focused on the development of the aggregation and referring expression generation subtasks in the microplanner.

References

Baker, Collin F. and Charles J. Fillmore and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the COLING-ACL,* Montreal, Canada.

Dale, Robert and Ehud Reiter. 1995. Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science* 18:233-263.

Fellbaum, Christiane. 1998. A Semantic Network of English: The Mother of All WordNets. In *Computers and the Humanities,* Kluwer, The Netherlands, 32: 209-220.

Fillmore, Charles J. and Christopher R. Johnson and Miriam R.L. Petruck. 2003. Background to FrameNet. In *International Journal of Lexicography.* Vol. 16 No. 3. 2003. Oxford University Press. Oxford, UK.

Fillmore, Charles J. 1985. Frames and the semantics of understanding. In *Quaderni di Semantica*, Vol. 6.2: 222-254.

Oberlander, Jon. 1998. Do the Right Thing… but Expect the Unexpected. *Computational Linguistics.* Volume 24, Number 3. September 1998, pp. 501-507. The MIT Press, Cambridge, MA.

Shemtov, Hadar. 1997. Ambiguity Management in Natural Language Generation, PhD Thesis, Stanford.

Kaplan, R. M. and J. Wedekind. 2000. LFG generation produces context-free languages. In *Proceedings of COLING-2000*, Saarbruecken, pp. 297-302.

Langkilde, Irene. 2000. Forest-based Statistical Sentence Generation. In Proceedings of the *North American Meeting of the Association for Computational Linguistics (NAACL),* 2000.

Langkilde, Irene and Kevin Knight. 1998. Generation that Exploits Corpus-Based Statistical Knowledge. In *Proceedings of Coling-ACL 1998.* Montréal, Canada.

Rambow, Owen, 2001. Corpus-based Methods in Natural Language Generation: Friend or Foe? Invited talk at the *European Workshop for Natural Language Generation*, Toulouse, France.

Reiter, Ehud and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press. Cambridge, UK.

Theune, Mariët. 2000. From data to speech: language generation in context. Ph.D. thesis, Eindhoven University of Technology.

Theune, Mariët. 2003. Natural Language Generation for Dialogue: System Survey. University of Twente. Twente, the Netherlands.

Walker, Marilyn and Owen Rambow and Monica Rogati. 2002. Training a Sentence Planner for Spoken Dialogue Using Boosting. *Computer Speech and Language, Special Issue on Spoken Language Generation*, July 2002.

Zue, Victor. 1997. Conversational Interfaces: Advances and Challenges. Keynote in *Proceedings of Eurospeech 1997.* Rhodes, Greece.