Natural Language Generation for Text-to-Text Applications Using an Information-Slim Representation

Radu Soricut
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
radu@isi.edu

Abstract

I propose a representation formalism and algorithms to be used in a new language generation mechanism for text-to-text applications. The generation process is driven by both text-specific information encoded via probability distributions over words and phrases derived from the input text, and general language knowledge captured by n-gram and syntactic language models.

A Text-to-Text Perspective on Natural Language Generation

Many of today's most popular natural language applications – Machine Translation, Summarization, Question Answering – are text-to-text applications. That is, they produce textual outputs from inputs that are also textual. Because these applications need to produce well-formed text, it would appear natural that they are the favorite testbed for generic generation components developed within the Natural Language Generation (NLG) community. Instead of relying on generic NLG systems, however, most of the current text-to-text applications use other means to address the generation need. In Machine Translation (MT), for example, sentences are produced using application-specific "decoders", inspired by work on speech recognition, whereas in Summarization, summaries are produced as either extracts or using task-specific strategies. The main reason for which text-to-text applications do not usually involve generic NLG systems is that such applications do not have access to the kind of information (e.g., semantic representations or lexical dependencies) that the input representation formalisms of current NLG systems require.

Formalisms and Algorithms for Text-to-Text Natural Language Generation

In my thesis, I will develop a representation formalism and algorithms – together with formal proofs – to be used in defining a new language generation mechanism. I will also implement this generation mechanism in a generic NLG system adaptable to a variety of text-to-text applications.

A representation formalism suitable for this purpose needs to be compact (e.g., linear in the number of words available), information-slim (e.g., not based on hard-to-acquire information, such as semantic representations), capable of handling probability distributions (critical in many current text-to-text applications), and application independent. In my thesis, I extend the proposal of Nederhof and Satta (2004) to arrive at a formalism that satisfies the above requirements, called weighted IDL-expressions (WIDL-expressions). The WIDL formalism encodes meaning via words and phrases that are combined using a set of formally defined operators handling word choice (the $\vee$ operator), precedence (the $\cdot$ operator), phrasal combination (the $\times$ operator), and underspecified/free word order (the $\|$ operator) (see Figure 1(a)). A probability distribution is associated with each instance of these operators, which assigns a probability value to each of the strings encoded under these operators (see the δs in Figure 1(a)). This representation formalism uses information available in most text-based applications (i.e., appropriate words and phrases, perhaps with associated probability distributions), and has the potential to handle all the available information that is relevant to the generation process (such as bias for word choice, word order, phrasal combination, etc.).

Using the WIDL representation, my generation mechanism is driven by both the information encoded in the WIDL probability distributions and various sources of language knowledge, such as n-gram language models (Goodman 2001) and syntactic language models (Charniak 2001). The generation algorithms I develop intersect the language described by WIDL-expressions with various language model combinations, while preserving the compactness property of WIDL-expressions. The output of the generation process is the string encoded by the input WIDL-expression that receives the highest score under the combination of WIDL and language model scores used. In Figure 1, for example, the string *the prisoners were finally released* is generated from the given WIDL-expression after being scored by the WIDL-expression, by an n-gram language model, and also by a syntactic language model (which hypothesizes syntactic trees on top of strings to arrive at its estimate).

Training the language models used by my generation mechanism requires a language knowledge acquisition step. While an n-gram language model requires no amount of

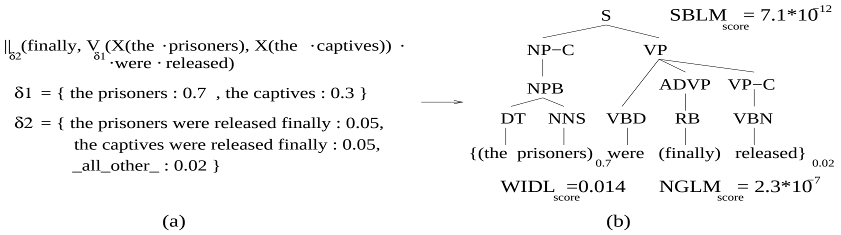
||δ2(finally, Vδ1(X(the ·prisoners), X(the ·captives)) · ·were · released)
δ1 = { the prisoners : 0.7 , the captives : 0.3 }
δ2 = { the prisoners were released finally : 0.05,
the captives were released finally : 0.05,
_all_other_ : 0.02 }
(a)
S
SBLMscore = 7.1*10-12
NP−C
VP
NPB
ADVP
VP−C
DT
NNS
VBD
RB
VBN
{(the prisoners)0.7 were (finally) released}0.02
WIDLscore =0.014
NGLMscore = 2.3*10-7
(b)

Figure 1: Mapping WIDL-expressions into syntactic parse trees. The WIDL$_{score}$ results from the probability distributions δ_1 and δ_2; the NGLM$_{score}$ results from evaluating the realized string *the prisoners were finally released* under the n-gram language model, and the SBLM$_{score}$ results from evaluating the hypothesized syntactic tree under the syntactic language model.

annotation/supervision to train, a syntactic language model requires unobserved variables (the syntactic trees) to train its parameters. The amount of annotated data available for this purpose (1 million words of English from the Wall Street Journal, known as the PennTree Bank) is insufficient to provide high coverage for learning lexical dependencies, which are critical for a high-performance syntactic language model. Bootstrapping techniques, such as self-training and co-training (Bloom & Mitchell 1998), have been shown to be ineffective for learning lexical dependencies for improving state-of-the-art parsing accuracy (Steedman *et al.* 2003). In the context of learning lexical dependencies for syntactic language modeling, however, the accuracy of a syntactic language model increases if self-training is used. Using a lexicalized statistical syntactic parser, I automatically acquire large amounts of language knowledge in the form of lexical and syntactic dependencies from unannotated text. These dependencies, albeit noisy, help increasing the ability of the syntactic language model to distinguish bad sentence realizations from good ones. This is considerably less expensive than relying on human expertise to create such knowledge, either by hand or through a corpus annotation process. It also has the advantage that new language knowledge events can easily be learned for different genres, domains, and even languages.

Current Status of Formalism and Algorithm Development

I have already defined the formalism of WIDL-expressions, and devised algorithms for intersecting WIDL-expressions with n-gram language models. I have proved the correctness of these algorithms, and showed that their complexity is linear in the complexity of the input WIDL expression. I have also implemented a syntactic language model starting from a lexicalized parsing model proposed by Collins (2003), and bootstrapped the model using large amounts of unannotated text. I performed preliminary experiments that show that the self-training procedure helps the syntactic language model perform significantly better on a word-ordering task. I also have empirical results that show that a log-linear combination between an n-gram language model and a bootstrapped syntactic language model outperforms any of these models taken separately on a word-ordering task.

Future Work on Text-to-Text Natural Language Generation

I still need to develop and implement algorithms for intersecting WIDL-expressions with a combination between an n-gram language model and a syntactic language model, and to prove the formal properties of these algorithms. I also plan to show the effectiveness of this generic, WIDL-based NLG system by implementing two end-to-end text-based applications. The first application is a machine translation system. I plan to capture translation model information using the probabilistic framework of WIDL-expressions, and target language model information using an n-gram and syntactic language model combination. The second application is a headline generation system. For such a summarization task, I plan to use information extraction techniques to associate scores with key words/phrases in the document(s) to be summarized, and map them into WIDL-expressions. An n-gram and syntactic language model combination will be in charge with realizing the most probable headline from a given WIDL-expression.

References

Bloom, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 92–100.

Charniak, E. 2001. Immediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

Collins, M. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*.

Goodman, J. 2001. A bit of progress in language modeling. Technical Report MSR-TR-2001-72, Microsoft Research.

Nederhof, M.-J., and Satta, G. 2004. IDL-expressions: a formalism for representing and parsing finite languages in natural language processing. *Journal of Artificial Intelligence Research* 287–317.

Steedman, M.; Osborne, M.; Sarkar, A.; Clark, S.; Hwa, R.; Hockenmaier, J.; Ruhlen, P.; Baker, S.; and Crim, J. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of the 11th Conference of the European Association for Computational Linguistics*.