A demo of FORGe: the Pompeu Fabra Open Rule-based Generator

Simon Mille [1], **Leo Wanner** [1,2]
[1] Universitat Pompeu Fabra, Roc Boronat 138, 08018 Barcelona, Spain
[2]Institució Catalana de Recerca i Estudis Avançats (ICREA),
Lluis Companys 23, 08010 Barcelona, Spain
firstname.lastname@upf.edu

Abstract

This demo paper presents the multilingual deep sentence generator developed by the TALN group at Universitat Pompeu Fabra, implemented as a series of rule-based graph-transducers.

1 Introduction

FORGe (Mille et al., 2017)[1] is a pipeline of graph transducers which, coupled with lexical resources, allows for generating texts, starting from a variety of abstract input structures. The current generator has been mainly developed for English on the dependency Penn Treebank (Johansson and Nugues, 2007) automatically converted to predicate-argument structures, and on Abstract Meaning Representations, using the SemEval'17 data (May and Priyadarshi, 2017). It is currently being adapted to languages such as Spanish, German French, and Polish, in the context of ontology-to-text generation as part of a dialogue system. Our generator follows the theoretical model of the Meaning-Text Theory (Mel'čuk, 1988), and performs the following actions: (i) syntacticization of predicate-argument graphs; (ii) introduction of function words; (iii) linearization and retrieval of surface forms.

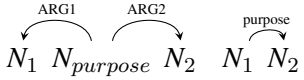
2 Overview of the system

In this section, we briefly describe the input to the system and the successive transductions .

[1]See this paper for an evaluation of the system in the context of the SemEval AMR-to-text generation challenge.

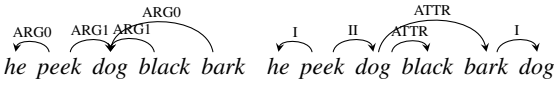
2.1 Inputs

The input structures can be trees or acyclic graphs that contain linguistic information only, which includes meaning bearing units and predicate-argument relations such as *ARG0* (if licensing external arguments, as in PropBank (Kingsbury and Palmer, 2002)), *ARG1*, *ARG2*, ..., *ARGn*). In order to allow for more compact representations, the generator can also handle "non-core" predicates as edges, be it with a generic label *nonCore*, or with a typed label such as *purpose*; see, for example two alternative representations of a *purpose* meaning between two nodes N_1 and N_2:



2.2 Generation of the deep syntactic structure

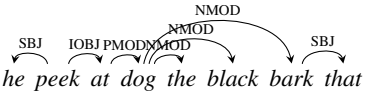
First of all, parts of speech are assigned to each node of the structure. Then, during this transduction, a top-down recursive syntacticization of the semantic graph is performed. It looks for the syntactic root of the sentence, and from there for its syntactic dependent(s), for the dependent(s) of the dependent(s), and so on. We first identify the root of a syntactic tree in case the original input structure does not contain one, and then, produce a well-formed tree that covers as much of the input graph as possible, while avoiding the possible dependency conflicts. In the following example, "peek" is chosen as the root (Left: predicate-argument; Right: Deep-Syntax):



Proceedings of The 10th International Natural Language Generation conference, pages 245–246,
Santiago de Compostela, Spain, September 4-7 2017.

2.3 Introduction of function words

The next step towards the realization of the sentence is the introduction of all idiosyncratic words (prepositions, auxiliaries, determiners, etc.) and of a fine-grained (surface-)syntactic structure that gives enough information for linearizing and resolving agreements between the different words. For this task, we use a valency (subcategorization) lexicon built automatically from PropBank and NomBank (Meyers et al., 2004). During this transduction, anaphora are resolved, and personal pronouns are introduced in the tree (this includes possessive, relative and personal pronouns). See, e.g., how the preposition "at" is introduced in the following surface-syntactic structure:



2.4 Resolution of morpho-syntactic agreements, linearization, and retrieval of surface forms

In order to resolve agreements, the rules for this transduction check the governor/dependent pairs, together with the syntactic relation that links them together. Some other rules order governor-dependent pairs and siblings with one another. We then match the triple <lemma><POS><morpho-syntactic features> with an entry of a morphological dictionary and simply replace the triple by the surface form. The final sentence corresponding to the running example would be *He peeks at the black dog that barks*.

3 A flexible multilingual generation pipeline

The presented pipeline is flexible from several perspectives. First, it is quite easily adaptable to different types of inputs; for instance, it took only one week to adapt it to the AMRs of SemEval'17. Second, many rules are language-independent, and others can be easily adapted to other languages, which means that, with good quality lexical resources, the effort for building a generator in a new language is minimal. Finally it is possible to substitute some parts of the pipeline with statistical modules, as, e.g., the transition between deep-and surface-syntax (Ballesteros et al., 2015) or the linearization step (Bohnet et al., 2011), in order to overcome a possible lack of coverage of the rules.

During the demo session, participants will be encouraged to play with the generator through a graphical interface, in order to see all the details of a generation process (in English, with some examples in German and Polish).

Acknowledgments

The work described in this paper has been partially funded by the European Commission under the contract numbers FP7-ICT-610411, H2020-645012-RIA, H2020-700024-RIA, and H2020-700475-RIA.

References

Miguel Ballesteros, Bernd Bohnet, Simon Mille, and Leo Wanner. 2015. Data-driven sentence generation with non-isomorphic trees. In *Proceedings of NAACL:HLT*, pages 387–397, Denver, CO, May–June. ACL.

Bernd Bohnet, Simon Mille, Benoît Favre, and Leo Wanner. 2011. StuMaBa: From deep representation to surface. In *Proceedings of ENLG*, pages 232–235, Nancy, France.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Poceedings of NODALIDA*, pages 105–112, Tartu, Estonia.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of LREC*, pages 1989–1993, Las Palmas, Canary Islands, Spain.

Jonathan May and Jay Priyadarshi. 2017. Semeval-2017 task 9: Abstract meaning representation parsing and generation. In *Proceedings of SemEval-2017*, pages 534–543, Vancouver, Canada, August. Association for Computational Linguistics.

Igor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. SUNY Press, Albany.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank Project: An interim report. In *Proceedings of the Workshop on Frontiers in Corpus Annotation, (HLT/NAACL)*, pages 24–31, Boston, MA, USA.

Simon Mille, Roberto Carlini, Alicia Burga, and Leo Wanner. 2017. Forge at semeval-2017 task 9: Deep sentence generation based on a sequence of graph transducers. In *Proceedings of SemEval-2017*, pages 917–920, Vancouver, Canada, August. Association for Computational Linguistics.