From IGT to precision grammar: French verbal morphology

Emily M. Bender, David Wax and Michael Wayne Goodman

1 Introduction

Interlinear glossed text (IGT, the familiar three-line format of linguistic examples) can be an extremely rich source of linguistic information, when linguists follow best practices in creating it (e.g., the Leipzig glossing rules, Comrie et al. 2003). The ODIN project (http://www.csufresno.edu/odin; Lewis 2006) recognized the value of IGT data as a reusable data type and has created a searchable IGT database. This paper represents early efforts in a project to combine aggregations of IGT with a second source of linguistic knowledge to automatically produce implemented formal grammars. The second source of linguistic knowledge is the LinGO Grammar Matrix customization system (Bender et al. 2010). The Grammar Matrix is a multilingual grammar engineering project which includes a cross-linguistic core HPSG (Pollard and Sag 1994) grammar and a set of analyses for cross-linguistically variable phenomena which can be selected via a web-based questionnaire. As an initial pilot study, we focus on verb morphology (including morphotactics and the morphosyntactic effects of affixes) and we begin with a best-case scenario: For our IGT, we use the complete paradigm for the French verb *faire* ('to do/make') provided by Olivier Bonami (pc), including 15,658 phonologically transcribed, morphologically segmented and glossed verb forms.

2 Background

The Grammar Matrix is a cross-linguistic grammar resource whose goals include: facilitating the rapid development of precision implemented grammars for any natural language, supporting grammar engineering for linguistic hypothesis testing, and combining depth of syntactic analysis with breadth of typological perspective (Bender et al. 2010). All of these goals can be better met to the extent that we can automatically translate linguistic analyses encoded in IGT into working, Grammar Matrix-derived grammars.

The Grammar Matrix is accessible via a web-based questionnaire (http://www.delph-in.net/matrix/customize). Users fill out this questionnaire with typological information about a language as well as lexical information including lexical types, lexical entries and lexical rules. The customization system then selects information from 'libraries' of stored analyses based on these specifications and outputs a working grammar fragment that is compatible with the DELPH-IN (http://www.delph-in.net) suite of grammar development and deployment tools.

Following HPSG practice, the Grammar Matrix handles morphology with lexical rules. The morphotactic framework (Goodman and Bender 2010) groups lexical rules into 'position classes' (PCs), which are supertypes to lexical rule types. PC definitions specify the input to the class (where in the derivation the PC appears), optionality, and any co-occurrence restrictions with other PCs. Lexical rule definitions specify the form of the affix (if any), the morphosyntactic or morphosemantic constraints (when available from the Matrix libraries), and co-occurrence restirctions. These co-occurrence restrictions can be seen as an extension of Beesley and Karttunen's (2003) flag diacritics, with the added ability to test for unification of flags where the flag-values exist in a type-hierarchy. In general, the system handles concatenative morphology only; following Bender and Good (2005) we relegate the handling of morphophonological changes to a separate processor.

To make the discussion below more concrete, (1) gives a sample IGT instance, drawn from ODIN. We refer to the first line as the 'language line', the second as the 'gloss line' and the third as the 'translation line'. The gloss line can be divided into tokens (e.g., 'ASP-cook-ACC') or grams (e.g., 'ASP' or 'cook'). Grams identifying roots or stems (e.g., 'cook' but not 'ASP') are also called lemmas.

(1) *Lu-lutu-in* *ng* *lalaki* *ang* *adobo.*
ASP-cook-ACC CS man ANG adobo
'The man will cook the adobo.' [tgl] (Rackowski and Richards 2005)

3 MOM (Matrix-ODIN Mash-up) System Overview

The MOM system takes as input a collection of IGT and produces as output a 'choices' file, which records the set of specifications that the Grammar Matrix needs to create a working, customized HPSG grammar fragment. In this case, we are using as input the French IGT described in §4. This process happens over 5 different steps: (i) the identification of verbs in the translation line, (ii) the alignment of translation words to gloss grams, (iii) the alignment of gloss grams to language words, (iv) the alignment of gloss grams to language morphemes for the verbs, and finally (v) the combination of lexical rules into position classes. All five of these steps are explained below. The first four of these steps must occur to create the necessary output, while the final step creates a more usable choices file for further (manual) customization.

The first step is the identification of verbs in the translation line. The system provides the list of sentences to the Charniak parser (Charniak 2000). The system then identifies as verbs any terminal node associated with a preterminal labeled with any verb tag. Using (1) as an example, the English string *The man will cook the adobo.* would be passed to the parser, which returns a parse tree. From the parse tree, the system would determine that the token 'cook' is the only verb in the translation line.

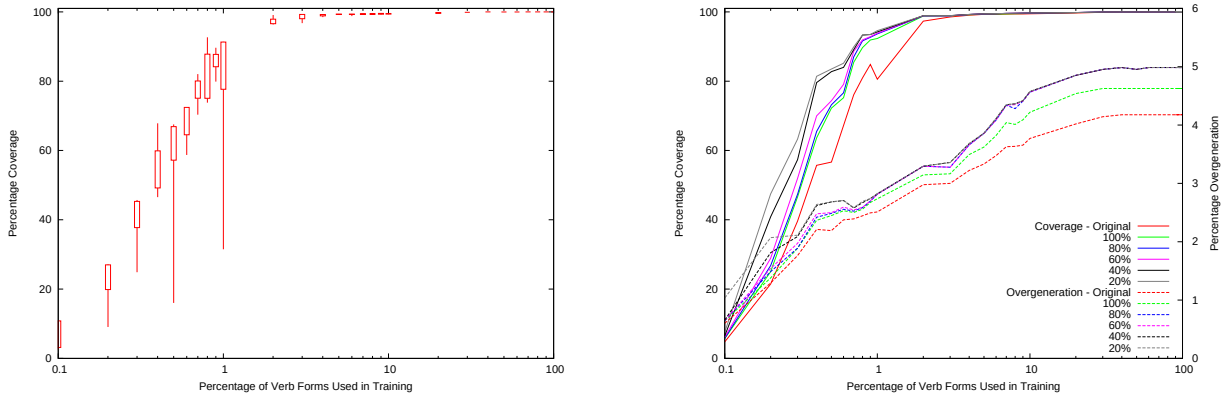
The next three steps involve taking strings from the IGT to create source and target pairs for alignment with GIZA++ (Och and Ney 2000), a statistical alignment tool used for machine translation. These alignments tell the system which elements from the gloss and language lines are associated with the verbs from the translation line. The alignments created by GIZA++ are one-to-many alignments, but the MOM system aligns both from the source to the target and the target to the source, and then takes the intersection of those to create a one-to-one alignment.

In the second step, we are interested in finding out which gloss grams represent the glossing of verbs. The system gains this information by aligning the words of the translation line with the individual grams of the gloss line. In (1), this entails aligning the translation line and the separated gloss grams ('ASP cook ACC CS man ANG adobo.') and results in the pairs *cook*/*cook*, *man*/*man* and *adobo*/*adobo*.

Step three identifies the verbs in the language line. The system does this by aligning the gloss words (tokens) with the language words. In (1), the system would align the strings 'ASP-cook-ACC CS man ANG adobo' and 'Lu-lutu-in ng lalaki ang adobo.' The alignment should match the tokens in the order they appear. This step will have marked *Lu-lutu-in* as the verb in the language line.

After identifying the verbs for both the gloss and translation line, step four aligns the grams in the gloss line with the morphemes in the language line, for verbs only. The grams are used to determine what syntactic and semantic effects to assign to each lexical rule. For example, when the system encounters 'NOM.3PL' as the gloss of an affix in the French data, it instructs the customization system to create a lexical rule for that affix that constrains the verb's subject to be 3rd person plural. The step four alignments are also used to identify verb roots in the language line, for creating lexical entries. In (1), this step will align the tokens into morpheme/feature pairs, like ASP/*lu*, cook/*lutu*, and ACC/*in*.

The output of step four is used in the creation of the first version of the choices file, in which each affix with a unique set of features and spelling is given a unique position class. This tends to create a more restricted grammar, with both lower coverage and lower overgeneration. To create a more robust grammar, the system combines the many position classes (step five). In an iterative process, it determines which current position classes are the best candidates for combination by comparing the amount of inputs each position class shares. It then combines the two position classes with the highest



(a) Coverage by training data

(b) Coverage/overgeneration by input overlap threshold

Figure 1: System performance by training data size, log scale

percent of shared inputs. A lower percent limit is provided to halt this process. Once there aren't two position classes which share more inputs than the threshold, the choices file is printed.

In the above steps that use GIZA++, we improve system performance with boosting techniques. GIZA++ was originally designed to produce alignments across bitexts where both sides are natural language. IGT, however, is more structured than that, and we can often tell heuristically that certain alignments are accurate. To exploit this information, the system adds 'sentence' pairs to the training data which consist of single tokens on each side (translation/gloss or gloss/language). These additions to the training corpus help GIZA++ recognize the alignments we know to be correct and, as a side-effect, improve the other alignments as well. To construct the 'one-to-one' sentences for steps between the translation line and gloss line, the system looks for tokens which are exactly the same or where one is a substring of the other. In the later steps when aligning between the gloss lines and language lines, the boosting data is created when the source and target both contain the same number of elements. Since the gloss and language lines should be token aligned, these tend to be well-formed IGT, which can be aligned heuristically. We use that data to help improve the alignments of noisier IGT examples.

4 Evaluation

The French data consists entirely of different forms of the French verb *faire* written phonetically in IPA. The analysis represented in the data set treats the so-called clitics as affixes (cf. Miller and Sag 1997), bringing the total number of forms to 15,658. In addition to the phonetic forms, a gloss is generated for each verb. Both the verb and the glossing are generated from a Perl script by Olivier Bonami. To be consistent with the IGT format, a mock translation ('do') is added to each IGT instance. Since the fifteen thousand forms we are working with represent the entire set of grammatical forms, we also use it as the test case. In addition to the correct forms, we generated 12,208 incorrect forms. These ungrammatical forms were generated from a combination of rearranging existing morphemes, adding non-morphemes, and duplication of morphemes. training over different amounts of data, but the testing was always over the entire set of grammatical and ungrammatical forms. Our tests using over 10% of the total data for training represent one sample each. However, for smaller training set sizes, we ran 10 train-test runs at each sample size. The numbers presented here are the averages of each set.

We evaluated the performance over two metrics: coverage, the amount of grammatical forms the grammar would parse; and overgeneration, the amount of incorrect forms the grammar would parse.

To see how the amount of training data affected system performance, we started by looking at 10% increments. We found that even with only 10% of the data available for training, the system was able to parse over 99% of the grammatical forms. We next looked at smaller training sets in increments of 1%. The grammars generated with 2% or more were still performing over 96%, and the grammars created with only 1% averaged around 80%. Finally, we trained the system from 0.1% to 0.9% of the data in steps of 0.1%. This finally lead to the expected decline in performance. The results for the coverage of the generated grammars before reducing position classes can be seen in Figure 1a.

The results above reflect first-pass grammars, with each lexical rule assigned to its own position class. We can increase coverage by combining lexical rules into position classes, effectively allowing inputs for particular rules that are not seen in the training data. As usual, such increases in coverage are accompanied by increases in over-generation (cases where the unseen inputs are in fact ungrammatical). We created grammars with combined position classes using different thresholds for the combination of lexical rules ('minimum limits') using the same training sets as in Figure 1a. The average improvement to coverage by minimum limit can be seen in Figure 1b. It tended to have a more dramatic effect on the grammars which neither performed poorly nor exceptionally. As expected the overgeneration also went up as the minimum limit to shared inputs decreased. This is especially harmful to the grammars which had the highest amounts of coverage. The changes to overgeneration can be seen in Figure 1b.

5 Conclusion and Future Work

This initial test of the MOM system with the French verb form paradigm has demonstrated that morphotactics and morphosyntactic constraints can be extracted from IGT. The French data represent an idealized test case, in two ways: First, IGT collected from linguistics papers or field projects will likely not contain complete paradigms for any particular verb. Second, naturally occurring IGT tends to have more noise (incomplete glossing, non-standard grams, etc.) than this data set.

Regarding the first dimension of idealization, the experiments reported here show that our system can achieve good coverage with only a tiny proportion of the paradigm given in the training data. Regarding the second, future work will include experiments on the noisier IGT available from ODIN. Additionally, we would like to expand the same principles used in verbal morphology toward other parts of speech as well as expanding the syntactic and semantic information extracted from IGT sources.

References

Beesley, K.R., and L. Karttunen. 2003. *Finite State Morphology*. CSLI Publications.

Bender, E.M., S. Drellishak, A. Fokkens, L. Poulson, and S. Saleem. 2010. Grammar customization. *Research on Language & Computation* 1–50.

Bender, E.M., and J. Good. 2005. Implementation for discovery: A bipartite lexicon to support morphological and syntactic analysis. In *CLS 41: The Panels*.

Charniak, E. 2000. A maximum-entropy-inspired parser. In *NAACL 2000*, 132–139.

Comrie, B., M. Haspelmath, and B. Bickel. 2003. The Leipzig glossing rules. http://www.eva.mpg.de/lingua/resources/glossing-rules.php.

Goodman, M.W., and E.M. Bender. 2010. What's in a Word? Refining the Morphotactic Infrastructure in the Lingo Grammar Matrix Customization System. Presented at the Workshop on Morphology and Formal Grammar, Paris.

Lewis, W.D. 2006. ODIN: A model for adapting and enriching legacy infrastructure. In *Proceedings of the e-Humanities Workshop*, Amsterdam.

Miller, P.H., and I.A. Sag. 1997. French clitic movment without clitics or movement. *Natural Language and Linguistic Theory* 15:573–639.

Och, F.J., and H. Ney. 2000. Improved statistical alignment models. In *ACL 2000*, 440–447.

Pollard, C., and I.A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago, IL and Stanford, CA: The University of Chicago Press and CSLI Publications.

Rackowski, A., and N. Richards. 2005. Phase edge and extraction: A Tagalog case study. *Linguistic Inquiry* 36:565–599.

Acknowledgments This material is based upon work supported by the National Science Foundation under Grant No. 0644097. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.