Spanish Resource Grammar version 2023

Olga Zamaraeva, Lorena S. Allegue, Carlos Gómez-Rodríguez
Universidade da Coruña, CITIC, Departamento de Ciencias de la Computación
y Tecnologías de la Información
Campus de Elviña s/n, 15071, A Coruña, Spain
{olga.zamaraeva, lorena.suarez, carlos.gomez}@udc.es

Abstract

We present the latest version of the Spanish Resource Grammar (SRG), a grammar of Spanish implemented in the HPSG formalism. Such grammars encode a complex set of hypotheses about syntax making them a resource for empirical testing of linguistic theory. They also encode a strict notion of grammaticality which makes them a resource for natural language processing applications in computer-assisted language learning. This version of the SRG uses the recent version of the Freeling morphological analyzer and is released along with an automatically created, manually verified treebank of 2,291 sentences. We explain the treebanking process, emphasizing how it is different from treebanking with manual annotation and how it contributes to empirically-driven development of syntactic theory. The treebanks' high level of consistency and detail makes them a resource for training high-quality semantic parsers and generally systems that benefit from precise and detailed semantics. Finally, we present the grammar's coverage and overgeneration on 100 sentences from a learner corpus, a new research line related to developing methodologies for robust empirical evaluation of hypotheses in second language acquisition.

Keywords: grammars, treebanks, Spanish, HPSG, syntactic theory

1. Introduction

Among the various approaches to computational linguistics, formal grammars are a link between linguistic theory and natural language processing (NLP). By formal grammars we mean fully explicit linguistic formalisms encoding the general principles and operations involved in generating syntactic structure. Such formalisms are tied to fully-fledged theories of syntax and are developed by linguists independently of specific NLP needs or tasks. For example, Minimalism (Chomsky, 1995), Lexical Functional Grammar (LFG; Kaplan et al., 1981), Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994) are linguistic theories of syntax. In contrast, the Penn-Treebank (PTB; Marcus et al., 1993, Bies et al., 2015) and the Universal Dependencies (UD; De Marneffe et al., 2021) do not explicitly encode why something should be labeled in a certain way, and for that reason we consider them labeling conventions but not fully-fledged theories of syntax. It is hard to ensure that a labeling convention for complex structure is followed consistently. Formal grammars, on the other hand, not only encode complex linguistic hypotheses explicitly but, if implemented on the computer, map sentences to structures automatically and are fully consistent. Any error in them can be fixed systematically. Any previously labeled data can then be re-labeled automatically.

Grammars take a long time to develop and the structures produced by them are harder to use than the annotation schemes developed specifically for NLP. For example, parsing can be much slower and the software stack generally needs to be more complex. However, grammars remain one of the few clear and long-term links between linguistics and NLP. In recent practice, formal implemented grammars have been used for computer assisted language learning (CALL) applications including grammar coaching (Flickinger and Yu, 2013; da Costa et al., 2016; Morgado da Costa et al., 2020).[1] They have also been used to create high quality treebanks to train semantic parsers (Buys and Blunsom, 2017; Chen et al., 2018; Lin et al., 2022). In particular, Lin et al. (2022) report a 35% error reduction and 14% absolute accuracy gain due to their use of the precise and consistent semantic representations generated by the English Resource Grammar (ERG; Flickinger, 2000, 2011). In this paper, we present the latest version of a grammar which can be used to create such high quality training data for Spanish.

The Spanish Resource Grammar (Marimon, 2010a; Marimon et al., 2014) is the second biggest implemented HPSG grammar (see §2.1). The latest version that we present uses a newer morphophonological analyzer through a completely reimplemented, easily editable Python interface. With this new interface, it is possible to use the grammar with the state-of-the-art HPSG parsers,

[1] By grammar coaching, we mean detecting a grammar mistake and analyzing it linguistically to provide informed feedback rather than correcting the sentence that is considered wrong (grammar correction). Both grammar coaching and grammar correction are NLP tasks in the context of workshops such as Building Educational Applications (BEA; Kochmar et al., 2023).

LREC-COLING 2024, pages 15093–15104
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

tailoring it as necessary. We report the SRG's accuracy on a portion of the AnCora/TIBIDABO corpus (Taulé et al., 2008; Marimon, 2010b) for the first time and present an example of using a learner corpus to find areas for improvement in the grammar's encoded analyses.

We present work that is unusual in the sense that we are breathing new life into a valuable resource which remained dormant for at least 10 years. Unlike other software, grammars do not become obsolete in the sense that they encode robust linguistic theories. For that reason, we are convinced that the SRG should be reintegrated into the computational linguistics landscape, providing the community with a resource similar to the English Resource Grammar (ERG). Like other software, grammars do become obsolete since they depend on tools which may become outdated, and fixing such dependencies can be expensive. We present a year of work that went into enabling the SRG to work with a better parser and establishing its accuracy on 2K sentences — a time consuming process which has to be done once, before automatic tools can be leveraged to quickly compare new iterations. Building upon this foundation, the grammar can be expanded such that its coverage improves.

The paper is organized as follows. In Section 2, we explain briefly the formalism behind the grammar implementation and what treebanking means in the context of grammar engineering. We also dedicate a section to crediting the original version of the grammar which took its author years to build. Section 3 describes what we did to bring the SRG up to date with the SOTA grammar engineering tools. Section 4 is dedicated to evaluation. It gives an overview of a set of phenomena that are covered, as revealed by a specially constructed test suite (§4.1); presents the results of the parsing with the grammar of 2,291 sentences from a Spanish news corpus (§4.2) along with the discussion of the issues that the evaluation reveals; and of 100 sentences from a Spanish learner corpus (§4.3). Section 4.3 includes an example of how the implemented grammar helps study syntactic hypotheses rigorously. The example shows a tension in the analyses that would likely remain overlooked if one did not implement them on the computer and did not run the grammar on a corpus containing ungrammatical examples.

2. Background and Methodology

This section describes the formalism used in the Spanish Resource grammar (§2.1-§2.2), related work, and the history of the Spanish Resource Grammar project (§2.3). It explains how grammar-based treebanking is different from using labeling conventions for manual annotation (§2.4).

2.1. Grammar engineering and DELPH-IN

Grammar engineering is a discipline for implementing syntactic theory on the computer. The ultimate research goal of grammar engineering is to refine syntactic theory in a general way and improve our systematic understanding of how human language works. A grammar implementation is a set of files. Parsers take such grammar implementations as input along with the sentences to parse the sentences according to the hypotheses encoded explicitly in the grammar files. As already mentioned, the theories underlying the formalisms used in grammar engineering encode the general principles hypothesized for syntactic structure by syntacticians. The associated level of complexity means they are less easy to use for NLP tasks but have bigger generalizability potential compared to labeling conventions.

There are several grammar engineering initiatives couched in various formalisms (Butt and King, 2002; Müller, 2015; Collins and Stabler, 2016). DELPH-IN (DEep Linguistic Processing with Hpsg INitiative; Copestake 2002a)[2] stands out as one with active international collaborations, an annual summit, and an emphasis on practical applications. The English Resource Grammar (ERG; Flickinger, 2000, 2011) is the largest engineered grammar we are aware of (including outside of DELPH-IN). It empowered the creation of a large high-quality treebank originally published as Oepen et al. 2004 with regular updates with each ERG release.[3] Another unique initiative within DELPH-IN is the Grammar Matrix (Bender et al., 2002, 2010; Zamaraeva et al., 2022), a system for automatic grammar creation based on typological description. The Grammar Matrix outputs grammar fragments which can then be developed further. DELPH-IN projects include grammars of Japanese, Chinese, Singaporean English, Hausa, German, Indonesian, Norwegian, Portuguese, Bulgarian, and more.[4] A grammar of Spanish of a non-trivial size was only implemented in the DELPH-IN formalism, to our knowledge.[5].

2.2. HPSG and MRS

DELPH-IN grammar engineering uses the HPSG and the MRS formalisms.[6] Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994)

[2] https://delph-in.github.io/docs/home/Home/

[3] http://svn.delph-in.net/erg/tags/2023

[4] https://delph-in.github.io/docs/grammars/GrammarCatalogue/

[5] The CoreGram project (Müller, 2015) included a starter Spanish grammar but, as far as we know, focused on other languages later on.

[6] For a more detailed overview of the relationship between the HPSG theory and computational linguistics, see Bender and Emerson 2021.

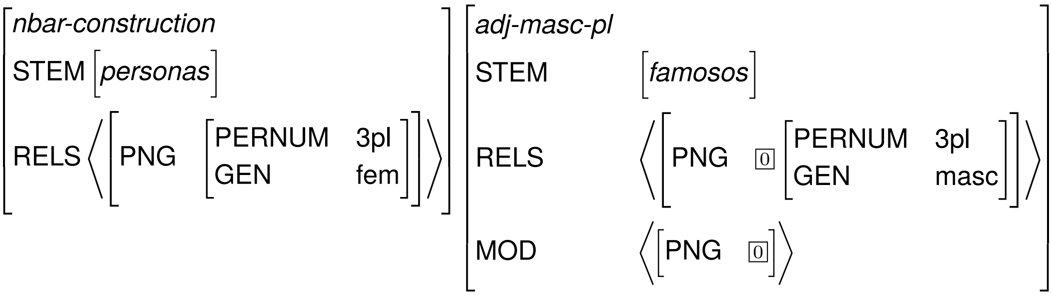

Figure 1: Two abbreviated feature structures produced by the SRG. Note the incompatible gender.

is a constraint-based unification theory of syntax (Carpenter, 1992). The formalism is fully explicit and serves as the foundation for multiple grammar engineering initiatives. HPSG sees syntactic structures as a hierarchy of phrasal and lexical types which can be instantiated as graphs containing feature-value pairs. The type hierarchy determines which values are compatible (can unify) and which are not. During unification-based parsing (e.g. Carroll, 1993; Callmeier, 2000; Crysmann and Packard, 2012; Slayden, 2012), first, lexical analysis is performed and then a parse chart is built bottom-up, attempting to account for the entire input string with one feature structure that is compatible with the *root conditions* (a set of constraints defining a full sentence). If something in a candidate feature structure cannot unify, the structure is discarded. Two simplified HPSG structures are presented in Figure 1. They correspond to two words from an example from a learner corpus (1).

(1) *Mis abuelos
my.3PL grandparent.MASC.PL
son personas
be.3PL.PRES.IND person.FEM.3PL
famosos.
famous.MASC.PL
Intended: 'My grandparents are famous people.' [spa; Yamada et al. 2020]

These structures (simplified for presentation) illustrate that two words have incompatible agreement values and so could not be used to form a noun phrase. The structure for the adjective (right) specifies an identity between the person, number, and gender (PNG) features of the adjective and the word that it modifies (the identity is represented as the numbered tag $\boxed{0}$). The values such as *3pl* and *fem* come from the type hierarchy (Figure 2) while the orthographies come from the lexicon, in this case paired with an external morphological analyzer. Since *fem* and *masc* do not unify,[7] the structure on the left could not possibly be on the MOD list of the structure on the right.

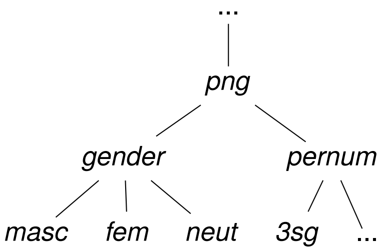

Figure 2: A portion of the type hierarchy

While an HPSG structure can be visualized as a tree (as shown later in Figure 5), in reality it is a more complex graph which includes full information on all the constraints (the tree only includes node labels which are not meaningful on their own and serve only for exposition). A graph for a full sentence can be visualized as an attribute-value matrix like the ones in Figure 1 but with the full set of constraints arising from the complete grammar. The *type* of such a structure is a phrasal type rather than a lexical type (phrasal and lexical types all belong to the same type hierarchy which is partially shown in Figure 2).

While Figure 1 shows a simple example of gender agreement, HPSG can model syntactic complexity in full detail. This is particularly useful when semantic nuances accessible through the syntax-semantics interface matter. HPSG has been used to solve issues related to negation scope (Packard et al., 2014; Zamaraeva et al., 2018) and compositionality generally (Lin et al., 2022).

Semantics in DELPH-IN is modeled via the Minimal Recursion Semantics formalism (MRS; Copestake et al., 2005). Any HPSG structure includes semantic constraints. Such constraints are partially shown in Figure 1 as RELS but a non-simplified structure produced by a DELPH-IN grammar includes a full MRS. An MRS is a bag of predications which include information about various semantic properties of the structure, including quantifier, negation, and modification scope; tense and aspect of events; person, number, and gender of entities, and infor-

[7] The fact that they do not unify is determined by the type hierarchy; here we omit the detailed explanation of the mechanics of unification which can be found in e.g. Copestake 2002b and Copestake 2002a.

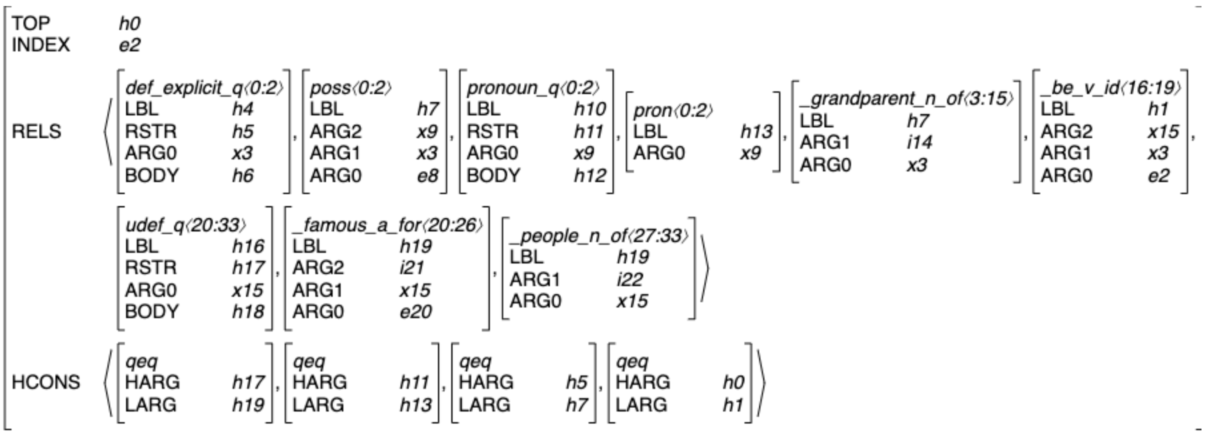
TOP h0
INDEX e2
RELS
def_explicit_q⟨0:2⟩ LBL h4 RSTR h5 ARG0 x3 BODY h6
poss⟨0:2⟩ LBL h7 ARG2 x9 ARG1 x3 ARG0 e8
pronoun_q⟨0:2⟩ LBL h10 RSTR h11 ARG0 x9 BODY h12
pron⟨0:2⟩ LBL h13 ARG0 x9
_grandparent_n_of⟨3:15⟩ LBL h7 ARG1 i14 ARG0 x3
_be_v_id⟨16:19⟩ LBL h1 ARG2 x15 ARG1 x3 ARG0 e2
udef_q⟨20:33⟩ LBL h16 RSTR h17 ARG0 x15 BODY h18
_famous_a_for⟨20:26⟩ LBL h19 ARG2 i21 ARG1 x15 ARG0 e20
_people_n_of⟨27:33⟩ LBL h19 ARG1 i22 ARG0 x15
HCONS
qeq HARG h17 LARG h19
qeq HARG h11 LARG h13
qeq HARG h5 LARG h7
qeq HARG h0 LARG h1

Figure 3: MRS for the sentence *My grandparents are famous people*. The main event is labeled *e2*; its dependencies are *x3, x15*. RSTR is used to track the scope of quantifiers and modification.

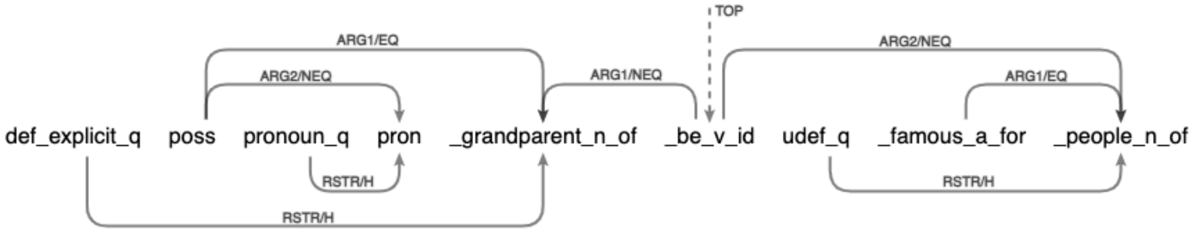

Figure 4: Dependency MRS for the sentence *My grandparents are famous people*.

mation structure. The MRS for the intended meaning of sentence (1) is given in Figure 3. An MRS can be automatically converted to a dependency structure (Figure 4).[8]

2.3. The Spanish Resource Grammar

The Spanish Resource Grammar (SRG) (Marimon, 2010a; Marimon et al., 2014) is the second biggest DELPH-IN grammar. It has 226 phrase structure types, 504 lexical rule types, 543 lexical types, and a lexicon of 54,510 lemmas.[9] The morphophonological analysis is done externally by Freeling (Padró and Stanilovsky, 2012; Carreras et al., 2004).[10] An input sentence is first run through Freeling which outputs one or more possible lemma-tag pairs for each word. The Freeling output is passed to the parser. The parser is a separate tool which takes a DELPH-IN grammar as input along with the input sentence. The parser maps the lemmas to the lexical entries in the lexicon and the tags to the lexical rules. The lexical rules are designed to ensure that the word is analyzed as an HPSG feature structure with the appropriate feature values such as specific values for gender, number, etc. When the SRG was first developed, the parser used was the PET parser (Callmeier, 2000). It has since stopped being supported but the grammar was left with a dependency in the form of the Freeling-parser interface.

SRG's accuracy[11] with respect to any corpora was never published (as far as we can tell). The coverage[12] for 17K sentences from AnCora (Taulé et al., 2008) with the PET parser was reported in Marimon 2010a and Marimon 2010b but these coverage figures include undesirable structures and were obtained with a slower parser that would time out often. We have obtained higher coverage (92% vs. 74%) thanks to the better parser speed and have manually verified the accuracy for sentences up to and including length 10.[13]

[8] Both figures were generated by the DELPH-IN online demo: http://delph-in.github.io/delphin-viz/demo.

[9] We would like to stress that the SRG was originally developed not by us and we have the big privilege to build on this substantial prior research that took years of effort by Marimon and her colleagues.

[10] https://nlp.lsi.upc.edu/freeling/

[11] **Accuracy:** how many grammatical sentences get assigned the desired semantic structure. This is one of the main metrics for grammar evaluation.

[12] **Coverage:** how many grammatical sentences are assigned some (any) constituency structure by the grammar, including not only correct but also spurious structures (leading to wrong or broken semantics). This metric is relevant for grammar development but less so for ultimate evaluation.

[13] In this process, we had access to treebanking decisions made by Marimon but we could not directly incorpo-

2.4. DELPH-IN Treebanking

In the context of DELPH-IN grammars, treebanking is in a sense the opposite to the treebanking in the settings such as Universal Dependencies (UD). Treebanks like UD are created manually with the goal to then train statistical tools on them. Conversely, DELPH-IN treebanks are created automatically by the manually built grammar. While it is clear that creating treebanks automatically is generally preferable due to higher consistency and scalability, there are two caveats: (1) language is highly ambiguous; (2) the grammar is not perfect. This means the grammar may generate many structures for each sentence, some of which may be semantically implausible or plain wrong. Therefore, the treebank generated automatically has to be gone through manually at least once, to pick the correct/desired tree and record it as the "gold" result for the particular sentence, or to record any "bugs" in the grammar that need to be fixed. The verification is done with respect to the semantic structure (MRS; Figure 3); in the context of DELPH-IN grammars, the specifics of the constituency structure are secondary as long as they lead to the correct semantics.[14]

Manual treebanking in DELPH-IN is the necessary step to train parse selection models required in most realistic applications. Human language is ambiguous, and the desired semantic structure is often determined only by pragmatics. That is outside of the scope of a syntactic theory, and an HPSG grammar will dutifully produce all the structures that it considers *syntactically* possible, and statistical tools are required to choose the *pragmatically* best one.[15]

The recorded gold results from a treebank can be automatically compared to a parse forest that represents new results in the next iterations of the grammar development. In other words, if a bug is fixed in the grammar or a new analysis is added, the impacts of the change can be assessed by the number and types of differences resulting from running the grammar on the same set of sentences and automatically comparing the output with the recorded gold.

The process of picking the gold tree from the full parse forest is time consuming but it is still faster and more consistent than creating a treebank manually from scratch.[16] Also, the bigger the treebanks, the better the parse ranking model, and once we are confident enough of the parse ranking and the grammar quality, we can parse new data and use it without verification.

In this section, we described the grammar engineering methodology including treebanking, a way of annotating corpora for syntactic structure consistently and semi-automatically. The next section summarizes the improvements we introduced to the SRG, in particular to be able to grow the treebanks more efficiently.

3. Summary of improvements

In this iteration of the SRG development, we achieved four main objectives: (1) have it work with the parsers ACE (Crysmann and Packard, 2012, with regular releases since 2012),[17] and the recent open-source version of the parser LKB (Carroll, 1993, with regular releases since the publication date);[18] (2) have it use a recent version of the Freeling morphophonological analyzer, v4.1;[19] (3) establish the current coverage and accuracy of the grammar on (a portion of) the AnCora corpus; (4) use the grammar on a learner corpus, as a step towards using it in CALL applications and to better understand the current grammar's overgeneration.[20] Adding new analyses to the grammar for it to support more phenomena is future work; first we needed to establish where it is now.

To reach these objectives, we have (1) revised the portion of the grammar responsible for the inflectional lexical rules to match Freeling v4.1 morphophonological analyzer tagset; (2) implemented a new, modular Python interface between Freeling, the grammar, and the ACE and LKB parsers;[21] (3) re-parsed the data up to length 20 with the ACE parser; (4) semi-manually verified the accuracy of the grammar on the sentences up to and including length 10; (5) explored the current grammar coverage and accuracy and documented them in

rate them due to the Freeling versioning incompatibility.

[14] This is characteristic of a syntax theory in development. We assume that we do not yet have a full understanding of what the complete set of correct syntactic analyses is, so we assess them via the correctness of the semantics that they yield.

[15] It is worth noting that pragmatically less plausible structures may still be meaningful, and there may be scenarios where producing them is a desideratum. A purely statistical system would be poor at such a task.

[16] Of course, the reverse is true about the grammar: it takes a lot of time to build, compared to a statistically trained resource.

[17] http://sweaglesw.org/linguistics/ace/

[18] https://delph-in.github.io/docs/tools/LkbFos/

[19] https://nlp.lsi.upc.edu/freeling/index.php/node/1

[20] **Overgeneration:** outputting wrong structures for grammatical sentences or any structure for an ungrammatical sentence.

[21] We are indebted to John Carroll for implementing several required modifications in the open-source version of the LKB parser.

the form of GitHub issues;[22] (6) prepared a new dataset based on an existing learner corpus to explore the grammar's overgeneration. In summary, we present a version of the grammar which is ready to use with SOTA DELPH-IN parsers and whose coverage and limitations are clearer. The version of the SRG corresponding to this paper can be found on GitHub under Releases v0.3.3.[23]. The release includes the treebanks.

4. Grammar evaluation

In this section, we present an assessment of the SRG that we performed for the first time thanks to the improvements summarized in §3. Generally speaking, grammars can be deficient in two ways: they can lack accuracy (not provide a correct structure for a sentence) or they can overgenerate (provide a wrong structure for a sentence). Accuracy and overgeneration are therefore the two principal metrics we use to evaluate grammars. We start from targeted evaluation of the accuracy on a set of constructed illustrative examples of linguistic phenomena (§4.1) and then present a larger-scale assessment of the accuracy on the AnCora/TIBIDABO corpus (§4.2). Measuring overgeneration can be harder, since corpora of ungrammatical examples are not common. Overgeneration can be estimated indirectly through noticing excessive ambiguity; if the grammar yields thousands of structures for a sentence, it is usually a sign of overgeneration, because even though natural languages are highly ambiguous, it is usually not the case that one sentence has thousands of possible meanings. We give the SRG ambiguity figure and comment on it at the end of §4.2. In §4.3, we suggest using a learner corpus for studying overgeneration, although what we present in this paper is merely a starting point.

4.1. The MRS test suite

The MRS test suite is a collection of sentences illustrating semantic phenomena that are accessible through syntax.[24] It is a way to assess a grammar's quality with respect to a range of linguistic phenomena by examining the adequacy of the MRS representations of the sentences illustrating the phenomena that the grammar yields. Across languages, the MRS structures for the listed sentences will in some cases be similar and in others they will not be, depending on how differently the languages in question express certain meanings. The test suite was first compiled for English in the context of the ERG development, and the English suite consists of 107 sentences.[25] The phenomena include different kinds of dependencies, scope of negation, scope of modification, implicit arguments (ellipsis), interrogatives, imperatives, and so on. The expectation is that we can compile a similar test suite for any language, as we expect to find such phenomena in most languages of the world. We also expect some differences because languages vary in the degree to which certain semantic phenomena are exposed through syntax. The semantically analogous sentences in the MRS test suites for different languages should correspond to each other by the number ID. The items which have no correspondence should have unique IDs.

We had the MRS test suite for Spanish compiled for the original release (Marimon, 2010a). We have edited the test suite to better reflect the facts of the Spanish language related to e.g. flexible word order interacting with focus. On the other hand, we corrected some mistakes where a Spanish sentence was identified as an equivalent to an English one where in reality the sentence had different semantics and thus should have been assigned a different ID. After adding some examples which seemed missing and removing some examples which seemed redundant,[26] the updated test suite consists of 106 sentences.[27]

The current SRG accuracy on the MRS test suite is 81%. Examining the items for which the grammar did not yield a correct analysis has allowed us to document some areas where the grammar should be improved. Based on the results of running the grammar on the MRS test suite, we opened 11 new issues in the SRG GitHub repository including the ones related to: Missing analysis of imperfective and perfective aspect distinction in some cases; missing possessive relations in some cases; missing interrogative semantics in many cases (underspecification between a question and a proposition, which is expected in Spanish yes-no questions but not in e.g. *wh*-questions); broken dependencies in some complex clauses including relative clauses and subordinate clauses, again, in some cases; insufficient implementation of the semantics associated with object clitics and the clitic *se* (a structure similar to the correct structure is yielded by the grammar but the dependency between the subject

[22] https://github.com/delph-in/srg/issues

[23] https://github.com/delph-in/srg/releases/tag/v0.3.3

[24] https://github.com/delph-in/docs/wiki/MatrixMrsTestSuite

[25] https://github.com/delph-in/docs/wiki/MatrixMrsTestSuiteEn

[26] Relevant discussion: https://delphinqa.ling.washington.edu/t/abrams-wiped-the-table-clean-in-spanish/881

[27] https://github.com/delph-in/srg/blob/main/tsdb/txt-id/mrs/mrs-updated.txt

sentence length	number of sentences	coverage	accuracy	times hit RAM limit
1	65	1.0	1.0	0
2	177	0.94	0.94	0
3	181	0.91	0.89	0
4	219	0.91	0.86	0
5	229	0.92	0.87	0
6	211	0.91	0.83	0
7	246	0.91	0.76	0
8	278	0.93	0.82	0
9	326	0.92	0.78	5
10	359	0.91	0.76	3
all	2291	0.92	0.82	8

Table 1: SRG accuracy on the first 10 portions of the TIBIDABO treebank

and the clitic is broken). All of these issues are major but it is expected that the grammar does not yet handle all of them perfectly because it is still a relatively young grammar in terms of the time that went into its development so far. The point of the MRS test suite is to provide a good estimate of where the grammar is now and where to go next.

4.2. TIBIDABO

The TIBIDABO treebank (Marimon, 2010b) is a version of the AnCora corpus (Taulé et al., 2008) sorted by sentence length (to see the effects of sentence length on the HPSG parsing speed; see §4.2.1) and annotated for HPSG structure. Marimon (2010b) reports coverage (but not accuracy) on the sentences up to length 40. We were able to recover 5,894 annotated sentences representing sentence length 1-19, representing 33% of the AnCora corpus. The rest of the TIBIDABO treebank appears to have been lost. We intend to rebuild it.

For the 5,894 sentences we have recovered, we had parse forests which were partially verified for the gold standard. But since Freeling 4.0 updates resulted in some incompatibility with the previous version, we had to look at each and every tree again.[28] For the latest release, we have managed to examine the parse forests and verify the presence of the correct tree for sentences up to and including length 10 (2,291 sentences in total). Together with the Freeling interface overhaul, the verified portion of the treebank constitutes the main contribution of this paper.

Table 1 shows the results we have so far on TIBIDABO. The coverage is stable at 92% although we expect it to go down as sentences become longer. The accuracy already goes down noticeably as the length goes up. Both coverage and accuracy suffer due to two main reasons: (1) parser limitations on long sentences, which could be overcome externally to the grammar (§4.2.1); and (2) genuine lack of the correct analyses encoded in the grammar (§4.2.2).

4.2.1. Parser limitations on longer sentences

HPSG parsing is relatively slow. The goal of the parsing is exhaustive search in a large space of possible complex structures (Carroll, 1993; Crysmann and Packard, 2012). With grammars which admit high ambiguity (see §4.2.3), the size of the parse chart can quickly become prohibitive as sentence length grows. This issue has been explored with respect to the ERG (Dridan et al., 2008; Dridan, 2009, 2013; Zamaraeva and Gómez-Rodríguez, 2023), and similar solutions can be applied to the SRG. Once the treebanks become big enough, a statistical model can be trained to filter out unlikely edges from the chart. Currently, to parse a sentence of length 10 takes 1 second/sentence on average.

4.2.2. Summary of genuine coverage issues

Apart from the issues documented in relation with the MRS test suite, we have documented two groups of problems: (1) issues related to (possibly wrong) Freeling tags; (2) issues related to multiword expressions — not an easily solved problem because there is no universal treatment of MWE that would not involve trade-offs (Contreras Kallens and Christiansen, 2022; Sag et al., 2002). In addition, there are linguistic issues which do not immediately form a group and which call for individual investigation and possible reanalysis of portions of the lexicon and the type hierarchy.[29]

Any change in the grammar may have wide effects on its behavior with respect to data. For that reason, extending coverage requires rigorous testing. We present one example of lack of coverage

[28] This does raise questions about the long-term desirability of the Freeling dependency; it may be possible to instead model the morphology directly in the grammar.

[29] The documented issues can be found here: `https://github.com/delph-in/srg/issues`.

and its preliminary investigation. The work for increasing the coverage is ongoing and new figures will be reported in future versions of the grammar.

(2) Mis amigos pueden venir si
my.PL friends can.3PL.PRES come.INF if
quieren.
want.3PL.PRES
'My friends can come if they want.' [spa]

Example (2) and similar examples are not parsed by the grammar. What we discovered is, according to the grammar, the verb *querer* ('to want') is assigned to a type which does not allow the kind of long-distance dependency that is required to form the sentence. In the sentence, the subject is shared between 'come' and 'want' and is not overtly present in the clause where 'want' is the predicate. The issue is related to the overall complex analysis of long-distance dependencies in the grammar which was not fully finished (as far as we can tell). Developing the analysis will automatically improve not only the coverage with respect to sentence (2) but with respect to many more examples containing this kind of long-distance dependencies.

4.2.3. Studying excessive ambiguity

The current version of the SRG has high ambiguity (482 structures per sentence, on average on the portions of TIBIDABO length 1-10). While natural languages including Spanish are highly ambiguous, having millions of structures per sentence (which is the case for some sentences) is clearly overgeneration. Such extreme figures are explained combinatorically; the longer the sentence, the more possibilities for different interpretations for each word and then each subconstituent containing each of those possibilities for each word. In some cases, this is inevitable and has to be sustained. In others, it may turn out that an additional constraint will preclude a number of chart edges from being hypothesized by the parser without any loss in the accuracy. The investigation for reducing ambiguity is ongoing work on which we do not report here.

4.3. COWSLH2

COWSLH2 is a corpus of written Spanish learner language developed at UC Davis (Yamada et al., 2020). The corpus contains over 100K sentences in the form of essays written by college students. Some sentences are annotated for grammatical errors. For the purposes of this paper, we semi-randomly selected 100 sentences of length up to 8. Of them, 36 are considered "ungrammatical" in the sense that they have some learner usage not characteristic of proficient Spanish speakers.[30] The remaining 64 are grammatical sentences. We ran the SRG on the sentences. Ideally, we would like the SRG to parse only the 64 grammatical ones. As for the ungrammatical ones, ideally, we would expect it to reject them (not assign any structure). Of course, we know the SRG is not perfect, so the purpose of this exercise is to see where the room for improvement is.

coverage	accuracy	overgeneration
100%	87%	61%

Table 2: SRG accuracy and overgeneration on 100 learner sentences

4.3.1. Assessing overgeneration with a learner corpus

Table 2 shows the results of running the SRG on the 100 short sentences from the learner corpus. The coverage is 100% meaning all of the 64 grammatical sentences were assigned some HPSG structure. However, that does not mean the corresponding semantics is the desired one; the accuracy is 87%. The large overgeneration figure (61%) means that the grammar currently generates some structure for many ungrammatical sentences.

The SRG's large overgeneration on the portion of COWSLH2 (61%) is not unexpected; controlling for overgeneration requires regularly testing the grammar with ungrammatical sentences, which is done routinely in e.g. the Grammar Matrix project (Bender et al., 2010) but, since larger grammars typically prioritize coverage over large corpora, overgeneration can grow. The bigger point here is that our ideas about how the grammar works are typically far from perfect and must be tested empirically and computationally. Following Bierwisch (1963), Butt et al. (1999), Bender (2008), Fokkens (2014), Müller (2015), Zamaraeva et al. (2022), *inter alia*, we emphasize that overgeneration and other problems with the grammar are easy to overlook if one (1) does not implement the grammar and only considers sets of syntactic analyses in isolation and on paper; (2) only tests the grammar on cherry-picked examples. Running the grammar on a learner corpus is one method of assessing overgeneration.

Consider one example of how the learner corpus helps us quickly find problems in the grammar.

(3) *Mis abuelos
my.3PL grandparent.MASC.PL
son personas
be.3PL.PRES.IND person.FEM.3PL
famosos.
famous.MASC.PL
Intended: 'My grandparents are famous people.' [spa; Yamada et al. 2020]

[30] The third author, whose first language is Spanish, verified the grammaticality of the sentences.

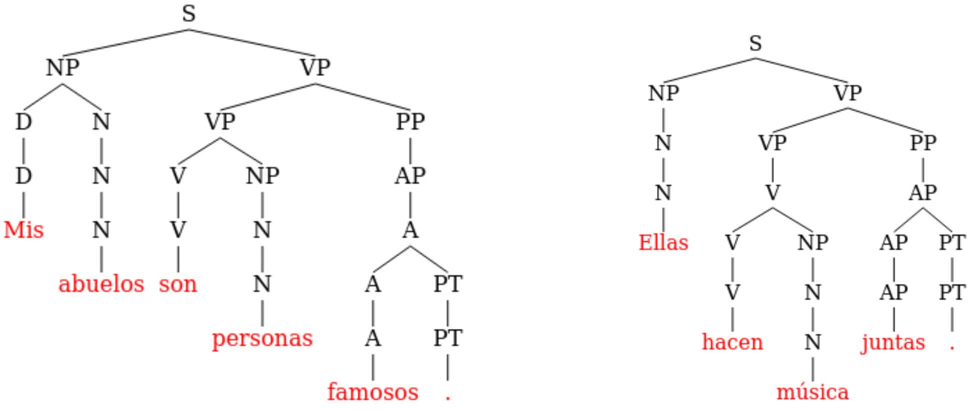

Figure 5: Left: A spurious tree for the Spanish sentence (3). Right: A correct analysis for sentence (4). (Unary chains in HPSG may represent lexical rules as well as rules for e.g. argument drop.)

The ungrammatical sentence (1) repeated here as (3) is actually parsed by the SRG. Examining the assigned structure, we see that the adjective *famosos* is attached high in the VP subtree, modifying the verb phrase rather than the noun (Figure 5). The semantics of such a structure appears nonsensical.[31] However, disallowing adjectives from attaching high generally of course is not the solution; looking at where else this structure occurs in the corpus, we find "healthy" examples like (4), where the structure is sensible and necessary.

(4) Ellas hacen
they.3PL.FEM do.3PL.PRES.IND
música juntas.
music.FEM.3SG united.FEM.PL
'They play music together.' [spa; Yamada et al. 2020]

What we find, then, is that in the SRG, the analysis of adjectives serving as verb modifiers applies too freely. But this is not the question of a simple reassignment of *famoso* to a different lexical type. The adjective *famoso* can be predicative (5), even if it cannot modify a head-complement construction such as *son personas*.

(5) Mis abuelos
my.3PL grandparent.MASC.PL
son famosos.
be.3PL.PRES.IND famous.MASC.PL
'My grandparents are famous.' [spa]

In Spanish, there are two verbs *to be*: *ser* and *estar*, and they are not interchangeable and convey different senses of being/state. A plausible hypothesis is that modified VP structures like in Figure 5 are possible only with adjectives that occur with the verb *estar* (e.g. *junto*) and not with the ones that occur with *ser* (e.g. *famoso*). But it is not clear whether this distinction is ultimately not pragmatic.[32] The question then is, how/whether to implement this distinction in the grammar and what effect will the changes have on the rest of the grammar, as evaluated not only with respect to (3)-(5) but to the entire corpus treebanked so far.

5. Conclusion and future work

We presented the latest version of the Spanish Resource Grammar (SRG) and its accuracy over a portion of the TIBIDABO treebank. The grammar can be used in linguistic research and in computer-assisted language learning (CALL) applications. The treebank, as it grows in the future, can be used for training high-quality semantic parsers for Spanish.

We argued that learner corpora should be leveraged to study overgeneration in grammars systematically, and presented an example where the grammar and the treebank force us to look at the current SRG analysis of Spanish adjectives in the context of the internal structure of the modificand.

The main avenue for future work apart from general grammar development towards higher coverage and accuracy and lower overgeneration is improving parsing speed. Slow parsing remains a serious problem which requires applying new methods. Recent experiments with training supertaggers for the English Resource Grammar are promising with the speed-up factor of 3 (Zamaraeva and Gómez-Rodríguez, 2023), however, training such a supertagger for the SRG will require that the treebanks grow first. That means continuing the research line presented in this paper.

[31] The semantics is something like "My grandparents are people while being famous."

[32] In principle, while the meaning "The grandparents are people while being famous" is bizarre and does not sound like something anyone would say, perhaps there is a semantic universe in which it makes sense.

6. Limitations

The main limitation of this work is the time cost of grammar engineering and treebanking. Due to the time costs involved, what we present here is work in progress, in the sense that the grammar does not yet cover some syntactic phenomena and some of its existing analyses can be improved: the overgeneration and the ambiguity should be reduced, for example. The results we present are only for sentences up to length 10, and some sentences cannot currently be parsed due to the parser limitations.

7. Acknowledgments

We acknowledge the European Union's Horizon Europe Framework Programme which funded this research under the Marie Skłodowska-Curie postdoctoral fellowship grant HORIZON-MSCA-2021-PF-01 (GAUSS, grant agreement No 101063104); the European Research Council (ERC), which has funded this research under the Horizon Europe research and innovation programme (SALSA, grant agreement No 101100615); Grant SCANNER-UDC (PID2020-113230RB-C21) funded by MICIU/AEI/10.13039/501100011033; Xunta de Galicia (ED431C 2020/11); and Centro de Investigación de Galicia "CITIC", funded by the Xunta de Galicia through the collaboration agreement between the Consellería de Cultura, Educación, Formación Profesional e Universidades and the Galician universities for the reinforcement of the research centres of the Galician University System (CIGUS). We also acknowledge grant GAP (PID2022-139308OA-I00) funded by MCIN/AEI/10.13039/501100011033/ and by ERDF "A way of making Europe".

8. References

Bies, Ann and Justin Mott and Colin Warner. 2015. *English News Text Treebank: Penn Treebank Revised*. Linguistic Data Consortium, 1.0, ISLRN 213-549-616-722-3.

Carreras, Xavier and Chao, Isaac and Padró, Lluís and Padró, Muntsa. 2004. *FreeLing: An Open-Source Suite of Language Analyzers*. Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04), ISLRN 376-309-383-772-9.

De Marneffe, Marie-Catherine and Manning, Christopher D and Nivre, Joakim and Zeman, Daniel. 2021. *Universal dependencies*. MIT Press, 1.0, ISLRN 586-682-285-530-1.

Marimon, Montserrat. 2010. *The Tibidabo Treebank*. ISLRN 051-145-400-668-2.

Padró, Lluís and Stanilovsky, Evgeny. 2012. *Freeling 3.0: Towards wider multilinguality*. LREC2012, ISLRN 376-309-383-772-9.

Taulé, Mariona and Martí, Maria Antónia and Recasens, Marta. 2008. *Ancora: Multilevel annotated corpora for Catalan and Spanish*. LREC, 2.0, ISLRN 252-495-813-736-1.

Emily M. Bender. 2008. Grammar engineering for linguistic hypothesis testing. In *Computational Linguistics For Less-studied Languages*, pages 16–36, Stanford, CA. CSLI Publications.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language and Computation*, 8:1–50.

Emily M Bender and Guy Emerson. 2021. Computational linguistics and grammar engineering. In Stefan Müller, Anne Abeillé, Robert D. Borsley, and Jean-Pierre Koenig, editors, *Head-Driven Phrase Structure Grammar: The handbook*.

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei.

Manfred Bierwisch. 1963. *Grammatik des deutschen Verbs (Grammar of German verbs)*. Akademie Verlag, Berlin.

Miriam Butt and Tracy Holloway King. 2002. Urdu and the Parallel Grammar project. In *Proceedings of the 3rd workshop on Asian language resources and international standardization-Volume 12*, pages 1–3. Association for Computational Linguistics.

Miriam Butt, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond. 1999. *A grammar writer's cookbook*. CSLI Publications, Stanford, CA.

J. Buys and P. Blunsom. 2017. Robust incremental neural semantic graph parsing. In *ACL*, pages 1215–1226.

Ulrich Callmeier. 2000. PET–a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering*, 6(1):99–107.

Bob Carpenter. 1992. *The logic of typed feature structures: with applications to unification grammars, logic programs and constraint resolution*, volume 32. Cambridge University Press.

John Carroll. 1993. *Practical unification-based parsing of natural language*. Ph.D. thesis, University of Cambridge.

Yufei Chen, Weiwei Sun, and Xiaojun Wan. 2018. Accurate SHRG-based semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 408–418.

Noam Chomsky. 1995. *The minimalist program*. MIT press.

Chris Collins and Edward Stabler. 2016. A formalization of minimalist syntax. *Syntax*, 19(1):43–78.

Pablo Contreras Kallens and Morten H Christiansen. 2022. Models of language and multiword expressions. *Frontiers in Artificial Intelligence*, 5:781962.

Ann Copestake. 2002a. Definitions of typed feature structures. In Stephan Oepen, Dan Flickinger, Jun-ichi Tsujii, and Hans Uszkoreit, editors, *Collaborative Language Engineering*, pages 227–230. CSLI Publications, Stanford, CA.

Ann Copestake. 2002b. *Implementing typed feature structure grammars*. CSLI Publications, Stanford, CA.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2-3):281–332.

Berthold Crysmann and Woodley Packard. 2012. Towards efficient HPSG generation for German, a non-configurational language. In *Proceedings of COLING 2012*, pages 695–710.

Luís Morgado da Costa, Francis Bond, and Xiaoling He. 2016. Syntactic well-formedness diagnosis and error-based coaching in computer assisted language learning using machine translation. In *NLPTEA*, pages 107–116.

Rebecca Dridan. 2009. *Using lexical statistics to improve HPSG parsing*. Ph.D. thesis, University of Saarland.

Rebecca Dridan. 2013. Ubertagging: Joint segmentation and supertagging for english. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1201–1212.

Rebecca Dridan, Valia Kordoni, and Jeremy Nicholson. 2008. Enhancing performance of lexicalised grammars. In *Proceedings of ACL-08: HLT*, pages 613–621.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(01):15–28.

Dan Flickinger. 2011. Accuracy v. robustness in grammar engineering. In Emily M. Bender and Jennifer E. Arnold, editors, *Language from a Cognitive Perspective: Grammar, Usage and Processing*, pages 31–50. CSLI Publications, Stanford, CA.

Dan Flickinger and Jiye Yu. 2013. Toward more precision in correction of grammatical errors. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 68–73.

Antske Fokkens. 2014. *Enhancing empirical research for linguistically motivated precision grammars*. Ph.D. thesis, Saarland University.

Ronald M Kaplan, Joan Bresnan, et al. 1981. *Lexical-functional grammar: A formal system for grammatical representation*. Massachusetts Institute Of Technology, Center For Cognitive Science.

Ekaterina Kochmar, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Nitin Madnani, Anaïs Tack, Victoria Yaneva, Zheng Yuan, and Torsten Zesch, editors. 2023. *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. Association for Computational Linguistics, Toronto, Canada.

Zi Lin, Jeremiah Liu, and Jingbo Shang. 2022. Neural-symbolic inference for robust autoregressive graph parsing via compositional uncertainty quantification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4759–4776.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank.

M. Marimon. 2010a. The Spanish Resource Grammar. In *LREC*.

Montserrat Marimon. 2010b. The Tibidabo treebank. *Procesamiento del Lenguaje Natural*, 45:113–119.

Montserrat Marimon, Núria Bel, and Lluís Padró. 2014. Automatic selection of HPSG-parsed sentences for treebank construction. *Computational Linguistics*, 40(3):523–531.

L. Morgado da Costa, R. Winder, Shu Yun Li, Benedict Christopher Lin Tzer Liang, Joseph Mackinnon, and F. Bond. 2020. Automated writing support using deep linguistic parsers. In *LREC*, pages 369–377.

Stefan Müller. 2015. The CoreGram project: Theoretical linguistics, theory development and verification. *Journal of Language Modelling*, 3(1):21–86.

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D Manning. 2004. LinGO Redwoods. *Research on Language and Computation*, 2(4):575–596.

Woodley Packard, Emily M Bender, Jonathon Read, Stephan Oepen, and Rebecca Dridan. 2014. Simple negation scope resolution through deep parsing: A semantic solution to a semantic problem. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–78.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Computational Linguistics and Intelligent Text Processing: Third International Conference, CICLing 2002 Mexico City, Mexico, February 17–23, 2002 Proceedings 3*, pages 1–15. Springer.

Glenn C Slayden. 2012. Array TFS storage for unification grammars. Master's thesis, University of Washington.

Aaron Yamada, Sam Davidson, Paloma Fernández-Mira, Agustina Carando, Kenji Sagae, and Claudia Sánchez-Gutiérrez. 2020. COWS-L2H: A corpus of Spanish learner writing. *Research in Corpus Linguistics*, 8(1):17–32.

Olga Zamaraeva, Chris Curtis, Guy Emerson, Antske Fokkens, Michael Wayne Goodman, Kristen Howell, TJ Trimble, and Emily M Bender. 2022. 20 years of the Grammar Matrix: Cross-linguistic hypothesis testing of increasingly complex interactions. *Journal of Language Modelling*, 10(1):49–137.

Olga Zamaraeva and Carlos Gómez-Rodríguez. 2023. Revisiting supertagging for HPSG.

Olga Zamaraeva, Kristen Howell, and Adam Rhine. 2018. Improving feature extraction for pathology reports with precise negation scope detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3564–3575.