Clausal Modifiers in the Grammar Matrix

Kristen Howell
University of Washinton
Department of Linguistics
kphowell@uw.edu

Olga Zamaraeva
University of Washington
Department of Linguistics
olzama@uw.edu

Abstract

We extend the coverage of an existing grammar customization system to clausal modifiers, also referred to as adverbial clauses. We present an analysis, taking a typologically-driven approach to account for this phenomenon across the world's languages, which we implement in the Grammar Matrix customization system (Bender et al., 2002, 2010). Testing our analysis on testsuites from five genetically and geographically diverse languages that were not considered in development, we achieve 88.4% coverage and 1.5% overgeneration.

Title and Abstract in Russian

Обстоятельственные придаточные в системе LinGO Grammar Matrix

В данной работе мы представляем новую библиотеку для системы LinGO Grammar Matrix. LinGO Grammar Matrix—это платформа для автоматического построения грамматик, где под грамматикой подразумевается программа, которая принимает на вход строку и возвращает соответствующие синтаксическую и семантическую структуру, построенные на основе синтаксической теории HPSG и семантического формализма MRS. Пользователь системы заполняет типологический опросник (отвечает на вопросы о конкретном языке, для которого требуется построить грамматику) и автоматически получает спецификацию, которую затем можно загрузить в различные приложения и получить собственно программу (грамматику). Ранее LinGO Grammar Matrix не поддерживала никаких придаточных предложений; в данной работе мы описываем новую библиотеку для обстоятельственных придаточных. Библиотека включает в себя новую страницу типологического опросника, набор признаковых структур HPSG, которые, с учетом уже имеющихся, позволяют составить корректные грамматики с обстоятельственными придаточными, и, наконец, собственно логику программы, которая и составляет эти грамматики в ответ на конкретный запрос пользователя. В процессе разработки мы проверяем, что система работает корректно в отношении списка возможных (coverage; охват грамматики) и невозможных (overgeneration; избыточные порождения грамматики) предложений из некоторого набора релевантных искусственных псевдоязыков (которые мы определяем как комбинации возможных выборов пользователя), а также из четырех естественных языков, отобранных из типологической литературы, посвященной обстоятельственным придаточным. Для оценки качества нашей библиотеки мы смотрим на охват и избыточные порождения грамматик, полученных нами для пяти естественных языков из разных языковых семей, отобранных уже после окончания разработки.

Proceedings of the 27th International Conference on Computational Linguistics, pages 2939–2952
Santa Fe, New Mexico, USA, August 20-26, 2018.

1 Introduction

The value of large-scale implemented precision grammars to linguists is twofold. First, by including analyses for linguistic phenomena that interact with each other, they are useful for verifying the consistency of linguistic theories (Bender, 2008; Müller, 2015); and second, they facilitate linguistic hypothesis testing by allowing the comparison of multiple analyses for a single phenomenon (Bender, 2010; Fokkens, 2014). Precision grammars parse and generate grammatical strings, emphasizing the linguistic correctness of an analysis by prioritizing *precision* (the number of inputs correctly parsed) over *recall* (the total number of inputs parsed), and in doing so allow linguists to test hypotheses over corpora. While the development of such grammars is time consuming, precision grammar starter kits can speed up the process by using stored analyses to create customized grammars (Bender et al., 2002).

Our present work is situated within the LinGO Grammar Matrix, a precision grammar customization system in which users answer typological questions about their language and a precision grammar fragment in the Head-driven Phrase Structure Grammar formalism (HPSG; Pollard and Sag, 1994) is produced. These starter grammars have given rise to such broad coverage grammars as the Spanish Resource Grammar (Marimon, 2010). We build on previous work in the Grammar Matrix by Trimble (2014), Poulson (2011) and others, following the methodology set forth by Drellishak (2009).

While language-specific analyses for clausal modifiers exist in the English Resource Grammar (ERG; Flickinger, 2000, 2011), the Jacy Grammar of Japanese (Siegel et al., 2016) and the Spanish Resource Grammar (Marimon, 2010), we are not aware of an implemented cross-linguistic analysis that has been applied to a broad range of typologically diverse languages. As subordinate clauses of any kind were not previously supported by the Grammar Matrix customization system and they are common in natural speech, their addition will extend the coverage of grammar fragments produced by this system. Restricting our focus to subordinate clauses that modify verbal projections, we present a library for clausal modifiers (or adverbial clauses) that supports numerous subordination strategies.

Languages typically employ multiple clausal modifier strategies which are marked by a wide range of characteristics. Supporting this variety poses a challenge for customization due to the range of complexity within each clausal modifier strategy, the multiplicity of syntactic phenomena that these strategies interact with and the possibility of multiple strategies within each language. We accomplish this with minimal additions to the Grammar Matrix, adding only two new lexical types and two phrase structure rules and defining strategy-specific subtypes to capture fine-grained complexity.

We begin with a description of related grammar engineering frameworks and annotation schemes (§2) followed by an overview of clausal modifiers, as seen in the typological literature (§3). Next we present an HPSG analysis of the phenomena (§4) and our contribution to the grammar customization system (§5). We describe our development data and evaluation on languages not considered during development, providing error analysis (§6) and conclude with a discussion of the uses for this library (§7).

2 Background

A variety of frameworks have been developed to support grammar engineering. While the Grammar Matrix uses the DELPH-IN Joint Reference Formalism (Copestake, 2002), a Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994) based formalism that balances expressive power with computational efficiency, CoreGram (Müller, 2015) uses another HPSG-based formalism – TRALE (Meurers et al., 2002; Penn, 2004). Other frameworks include ParGram (Butt et al., 2002), the MetaGrammar project (de La Clergerie, 2005) and Grammatical Framework (Ranta, 2004). However, the Grammar Matrix provides a particularly suitable framework for our analyses because it requires its libraries to be typologically robust. As a result, multiple analyses must be developed for a particular phenomenon to account for its cross-linguistic distribution, and these analyses must interact with other phenomena in the customization system in order to produce customized grammars for any given language.

The grammars created by the Grammar Matrix produce syntactic annotations in the HPSG formalism and semantic annotations using Minimal Recursion Semantics (MRS; Copestake et al., 2005) for the strings they parse. These representations can capture information akin that captured by the Universal Dependencies annotation scheme and annotations in PropBank. On the syntactic side, the Universal

Dependencies annotation scheme (McDonald et al., 2013) uses the SCONJ and ADV part of speech tags for subordinators, corresponding to our adposition lexical type in §4.1 and adverb lexical type in §4.2, respectively. These are dependents of the subordinate verb, via a MARK dependency relation, which corresponds to our *basic-head-comp-phrase* in §4.1 and *isect-mod-phrase* in §4.2. In UD, the subordinate verb is a dependent of the matrix verb, via the ADVCL relation, which corresponds to our *scopal-mod-phrase*, described in §4.1. On the semantic side, PropBank (Kingsbury and Palmer, 2002; Palmer et al., 2005) marks different semantic classes of clausal modifiers, such as temporal (ARGM-TMP) and purposive (ARGM-PURP). One key difference is that in PropBank the subordinate clause is a dependent of the matrix verb, whereas in our MRS representations the matrix verb is an argument of the subordinator.

3 Typological Patterns

The typological literature describes a number of strategies for clausal modifiers and, in any given language, multiple strategies may be employed. Drawing on the review in Thompson et al. (2007), we can describe the range of clausal modification strategies in terms of the subordinator on the one hand, and additional characteristics of the subordinate clause on the other. A clausal modifier may be marked by a subordinator,[1] as in (1) from Japanese [jpn]; a subordinator pair comprising a subordinator in the embedded clause and an adverb in the matrix clause, as in (2) from Mandarin [cmn]; or special verbal morphology, as in (3) from Luiseño [lui].

(1) Ame ga agaru **to**, Gon wa hotto shite ana kara haidemashita
rain NOM stop when Gon TOP relief perform.INF hole from sneak.out.PST
'When the rain stopped, Gon got relieved and came out of the hole' [jpn]
(adapted from Thompson et al. 2007)

(2) **Yīnwèi** tiān hēi le, **suǒyǐ** wǒ méi chū - qu
because sky black CRS so 1SG NEG exit - go
'Because it had gotten dark, I didn't go out.' [cmn] (adapted from Li and Thompson 1989)

(3) Yaʔáʃ ŋéŋi ʂuŋá·l kí·ʃ pu-wá·qi-**pi**
man leave.remote woman house.ACC her-sweep-PURP
'The man left in order for the woman to sweep the house' [lui]
(Davis 1973 in Thompson et al. 2007)

Subordinators can occur either before or after the verb phrase or sentence in the subordinate clause. While in some languages their position is restricted to only before or only after the clause, in others the subordinator may attach freely before or after the clause, attaching at either the verb phrase (VP) or sentence (S) level. Clausal modifier strategies with subordinator pairs have an adverb in the matrix clause, which must co-occur with a particular subordinator, as in (2), and the position of that adverb may also be strict or free, attaching before or after the VP or S (Li and Thompson, 1989). Cross-linguistically, the subordinator pair strategy is particularly common for *if ... then* constructions.

The clausal modifier itself shows variation in both its external distribution and internal characteristics. The clausal modifier may attach before or after the matrix clause at the either the VP or S level. This distribution might also be strict or free.

Internal characteristics of the subordinate clause include constraints on the subject, word order and verbal morphology. In some strategies the subject of the modifying clause is shared with the matrix clause. In this case the subject of the subordinate verb and the matrix verb are co-referential and it is unexpressed in the subordinate clause. Additionally, some languages have a distinct word order in the subordinate clause. For example, German is a verb second language with verb final word order in embedded clauses (Thompson et al., 2007).

Finally, the clausal modifier may also be marked by special verbal morphology. Certain subordinators often occur with a particular morphological form or, if the clausal modifier strategy does not involve a subordinator, the morphological form itself may be associated with a particular semantic predication, as in (3). Many Turkic and Austronesian languages, require subordinate clauses to be nominalized.

[1] These are sometimes called subordinating conjunctions or clause linkers.

As described in Noonan (2007), this generally involves a nominalization morpheme on the verb and a change in the clause's internal distribution from that typical of verbal projections to that typical of nominal projections. Furthermore, this often includes an alternate case frame for nominalized verbs.

Our review of the typological literature reveals that clausal modifiers can be marked with a subordinator, a subordinator paired with a matrix adverb or no suborinator at all, illustrated in (1)–(3), and that the clause may bear a number of additional characteristics. We also find that it is common for languages to employ different strategies for different classes of clausal modifiers. In the next section we will develop an analysis to account for these typological patterns.

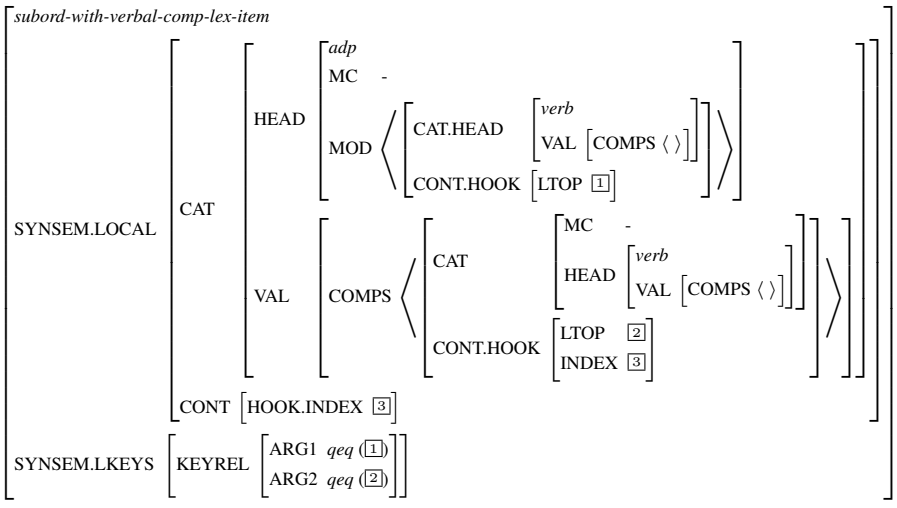
4 Analysis

We present an analysis for clausal modifiers using the HPSG formalism (Pollard and Sag, 1994) for syntactic structure and Minimal Recursion Semantics (MRS; Copestake et al., 2005) for semantic representation. HPSG uses lexical entries, lexical rules and phrase structure rules, encoded as feature structures, to model language. Central to HPSG is a notion of headedness, such that most phrases have a distinguished head daughter and syntactic features are passed up from the head daughter to the mother. Additionally, HPSG describes its features using a multiple inheritance hierarchy, allowing supertypes to capture broad generalizations and subtypes to add fine-grained detail. While there is little theoretical HPSG literature on a cross-linguistic analysis of clausal modifiers, we do find target semantic representations in the ERG (Flickinger, 2000, 2011), and take advantage of existing type definitions in the Grammar Matrix (Bender et al., 2002) on which to build our analysis.

Whereas we described the typology of clausal modifiers in terms the surface forms possible in each strategy, we develop our analysis with respect to underlying lexical type of the subordinator. We posit two types of subordinators: an adposition (§4.1), which is the head of its clause, and therefore must attach to to the subordinate clause at the sentence (S) level, and whose position (before or after the sentence) is strict; and an adverb (§4.2), whose distribution is more free—it may attach to the subordinate clause at either the S or verb phrase (VP) level and may attach either before or after the S or VP constituent.[2] If there is no subordinator, we analyze subordination as morphological (§4.3). For each of these three analyses, we define a new lexical type and/or unary rule for subordination. Finally, we add additional constraints, or feature specifications, to subtypes of the lexical types and unary rules to capture the variation described in §3.

4.1 The Adposition Subordinator

The adposition subordinator is the head of the subordinate clause, defined in (4) as a scopal modifier that takes a clause as its complement.[3] The subordinator is a type of adposition ([HEAD *adp*]) whose comp-

(4)


[2]While we suggest the adposition analysis unless there is evidence that the subordinator is an adverb, the strict S-attachment associated with adpositions is also possible under our adverb analysis.

[3]Due to limited space, we use the notation *qeq*() as a stand-in for the way we build the handle constraint. For a more detailed account of the handle constraint, see Copestake et al. 2005.

lement is [HEAD *verb*].[4] The complement clause is constrained with a boolean feature MC, indicating that it is not compatible with the characteristics of matrix clauses in the language, and has an empty complement list ([COMPS ⟨ ⟩]), requiring that the verb's complement requirements have been satisfied. Similarly, the subordinator modifies a clause (specified on the MOD list), which is [HEAD *verb*] and [COMPS ⟨ ⟩].[5]

Abstracting away from some of the details of semantic composition, the main semantic contribution of this type is a predication (the value of KEYREL), whose first semantic argument (ARG1) is the matrix clause and whose second argument (ARG2) is the subordinate clause. The result of these constraints is an MRS representation, like that in (5) for the sentence *Kim left when Pat arrived*, such that the predication _when_subord_rel[6] has two arguments which point (via *qeq*) to the matrix and subordinate verbs.

(5)
$$\left\langle h_1, e_3, \left| \begin{array}{l} h_4{:}\text{proper\_q}\langle 0:3\rangle(\text{ARG0 } x_6\{\text{PERS 3, NUM sg, IND +}\}, \text{ RSTR } h_5, \text{ BODY } h_7), \\ h_8{:}\text{named}\langle 0:3\rangle(\text{ARG0 } x_6, \text{ CARG } \textit{Kim}), \\ h_9{:}\text{\_leave\_v\_1}\langle 4:8\rangle(\text{ARG0 } e_3\{\text{SF prop, TENSE past}\}, \text{ ARG1 } x_6, \text{ ARG2 } p_{10}), \\ h_2{:}\text{\_when\_x\_subord}\langle 9:13\rangle(\text{ARG0 } e_{11}\{\text{SF prop}\}, \text{ ARG1 } h_{12}, \text{ ARG2 } h_{13}), \\ h_{14}{:}\text{proper\_q}\langle 14:17\rangle(\text{ARG0 } x_{16}\{\text{PERS 3, NUM sg, IND +}\}, \text{ RSTR } h_{15}, \text{ BODY } h_{17}), \\ h_{18}{:}\text{named}\langle 14:17\rangle(\text{ARG0 } x_{16}, \text{ CARG } \textit{Pat}), \\ h_{19}{:}\text{\_arrive\_v\_1}\langle 18:25\rangle(\text{ARG0 } e_{20}\{\text{SF prop, TENSE past}\}, \text{ ARG1 } x_{16}) \end{array} \right| \\ \{\, h_{15}\ qeq\ h_{18}, h_{13}\ qeq\ h_{19}, h_{12}\ qeq\ h_9, h_5\ qeq\ h_8, h_1\ qeq\ h_2 \,\} \right\rangle$$

We take advantage of existing phrase structure rules in the Grammar Matrix for complement and modifier attachment. The *basic-head-comp-phrase* phrase structure rule attaches the adposition to its complement (the subordinate clause) and the *scopal-mod-phrase* attaches the subordinate clause to the constituent it modifies (the matrix clause).

4.2 The Adverb Subordinator

In contrast with the adposition subordinator, the adverb subordinator is not a syntactic head, and therefore has more flexibility with respect to its position in the subordinate clause. As described in this section, because adverbs are not syntactic heads, many features will not be passed up the tree. Thus, rather than adding constraints on the matrix clause to the adverb's lexical type, we add them to the clausal modifier via a non-branching rule and use a new feature, SUBORDINATED to constrain the non-branching rule's daughter to contain the appropriate adverb.

The Lexical Type We define the adverb subordinator type as a non-scopal adverb. It has one element on its MOD list, the subordinate clause, which it can attach to at the VP or S level via the *isect-mod-phrase* rule, already in the Grammar Matrix. The element on the adverb's MOD list, like the complement of the adposition subordinator, is [HEAD *verb*] and [MC −] to prevent the adverb subordinator from occurring in matrix clauses.

The SUBORDINATED Feature Due to the constraints on composition in MRS (as codified, for example, in Copestake et al. 2001), we cannot introduce the subordinate predication on the adverb. MRS composition is done locally at each level in the tree, and though the adverb has access to the semantic head of the subordinate clause (the verb), it doesn't have access to any information about the matrix clause. Therefore, we add the predication and related semantic constraints with a unary rule, after the clause is fully formed. To link each adverb subordinator to a unary rule that contributes the corresponding predication, we introduce the SUBORDINATED feature, which is passed up by the various phrase structure rules. For each adverb, a feature value is created under SUBORDINATED and the unary rule that adds the predication will select for a daughter with the right SUBORDINATED value. To prevent multiple subordinators in the same sentence, all lexical entries for verbs are [SUBORDINATED none] as is the element on the adverb subordinator's MOD list. Once an adverb subordinator attaches, changing the clause's SUBORDINATED value, it will not be compatible with any other adverb subordinator.

[4] Note that this type will not extend to nominalized clauses. A type with a nominalized complement is described in §4.4.

[5] Additional constraints regarding the subjects of these clauses, which will determine whether or not the subject is expressed in the subordinate clause and if the subordinate clause modifies a VP or S, are added to subtypes, as described in §4.4.

[6] The specific predication symbol is specified on the lexical entry for each subordinator.

The Unary Rule We define the *adv-marked-subord-clause-phrase* such that it constrains its daughter to be [HEAD *verb*], [MC −] and, to prevent the rule from applying more than once, [MOD ⟨ ⟩]. The valence features must be fully satisfied on the mother and daughter, with the exception of the SUBJ list, which is identified between daughter and mother in order to accommodate subject sharing.

The main contribution of this rule is the addition of the matrix clause to the subordinate clause's MOD list as well as the subordinating predication and corresponding constraints. The element added to the MOD list is the same as that on the adposition subordinator's MOD list. The unary rule adds an *arg-12-ev-relation*, which has two arguments that are identified with the semantic content of the matrix and subordinate clauses respectively. The unary rule's daughter is identified with ARG2 and the matrix clause is identified with ARG1. Subtypes of this rule include a specific predication value, and we use the SUBORDINATED feature to identify the daughter of the unary rule with a clause marked by the appropriate subordinator for that predication. The resulting MRS is the same as that produced by the adposition subordinator in (5).

4.3 Morphological Subordination

Finally, morphological subordination involves a unary rule that selects a clause with particular morphological features and adds an element to the modifier list and a predication with the appropriate semantic identities. The *morphological-subord-clause-phrase* is identical to the *adv-marked-subord-clause-phrase* with one key difference: It selects a daughter with one or more syntactic or semantic features that are specific to the strategy, as described in §4.4, rather than using the SUBORDINATE feature.

4.4 Additional Constraints

For each clausal modifier strategy, we create a subtype of the appropriate lexical type and/or unary rule described in §4.1–4.3, adding additional constraints that capture the variation described in §3. Table 1 presents the additional features that constrain each phenomenon and indicates whether those features are expressed on the lexical type or unary rule.

Table 1: Features for Clausal Modifier Strategies

Constraints	Adposition Subordinator	Adverb Subordinator	No Subordinator
Clause Position	{POSTHEAD $+, -, bool$} (lexical type)	{POSTHEAD $+, -, bool$} (unary rule)	{POSTHEAD $+, -, bool$} (unary rule)
Clause Attachment	{MOD.SUBJ $\langle\,\rangle, \langle[\,]\rangle, none$} (lexical type)	{MOD.SUBJ $\langle\,\rangle, \langle[\,]\rangle, none$} (unary rule)	{MOD.SUBJ $\langle\,\rangle, \langle[\,]\rangle, none$} (unary rule)
Subordinator Position	{INIT $+, -$} (lexical type)	{POSTHEAD $+, -, bool$} (lexical type)	
Subordinator Attachment	{COMPS.SUBJ $\langle\,\rangle$} (lexical type)	{MOD.SUBJ $\langle\,\rangle, \langle[\,]\rangle, none$} (lexical type)	
Matrix Pair	{SUBPAIR} (lexical type)	{SUBPAIR} (lexical type)	
Special Morphology	{COMPS.FEATURE} (lexical type)	{MOD.FEATURE} (lexical type)	{DTR.FEATURE} (unary rule)
Nominalization	{COMPS.NMZ +} (lexical type)		{DTR.NMZ +} (unary rule)
Shared Subject	{COMPS.SUBJ $\boxed{0}\langle unexpressed\rangle$} {MOD.SUBJ $\boxed{0}$} (lexical type)	{DTR.SUBJ $\boxed{0}\langle unexpressed\rangle$} {MOD.SUBJ $\boxed{0}$} (unary rule)	{DTR.SUBJ $\boxed{0}\langle unexpressed\rangle$} {MOD.SUBJ $\boxed{0}$} (unary rule)

Clause Position and Attachment The first constraints we add govern the external distribution of the clausal modifier. Head-modifier rules are sensitive to the feature POSTHEAD, so if a clausal modifier strategy occurs strictly before the matrix clause, we add the constraint [POSTHEAD −]; if it occurs strictly after, [POSTHEAD +]; and if it can occur in either position, we leave this feature underspecified. We constrain clause attachment with the subject list of the matrix clause. If the modifier attaches to a sentence, the matrix clause must already have a subject, signified by an empty subject list ([SUBJ ⟨ ⟩]). On the other hand, if it attaches to a verb phrase, we constrain this list to be non-empty ([SUBJ ⟨ [ ] ⟩]) and if it can attach to either the VP or S, we leave this constraint underspecified. These constraints go directly on the lexical type of the adposition subordinator and on the unary rule for the adverb subordiantor.

Subordinator Position and Attachment If the subordinator is an adverb, the subordinator position and attachment are constrained the same way as the clausal modifier's, except that these constraints are

specified on the lexical type, so that they will govern the adverb's distribution in the subordinate clause. If the subordinator is an adposition, it is the head of its clause and attaches strictly to a sentence, so the element on its COMPS list is necessarily [SUBJ ⟨ ⟩]. It can attach at the beginning or end of the clause (but only one or the other), which is constrained with the INIT feature. We add INIT to the head-complement rules to constrain the order of the head and complement. If an adposition attaches at the beginning of the clause, it is [INIT +] and if it attaches at the end it is [INIT −].[7]

Subordinator Pairs Both adverbial and adpositional subordinators can require a adverb in the matrix clause (as illustrated in (2)), which we treat as a scopal modifier, constraining its position and attachment the same way as the adverb subordinator. We introduce the SUBPAIR feature, and for each pair of subordinators in the language, we create a unique value, which is added to the matrix adverb's lexical type. The head-modifier rule passes the SUBPAIR value up from the non-head daughter and this feature is propagated up through the head daughter by the other phrase structure rules. In the subordinate clause, this feature is specified on the MOD list of the adposition subordinator, or the lexical type of the adverb subordinator.[8] Finally the *scopal-head-mod-phrase* identifies the SUBPAIR value of the non-head-daughter's MOD list with that of the head-daughter.

Special Morphology and Nominalization Many subordination strategies, whether they include a subordinator or not, require special morphology on the embedded verb. Certain morphological forms can be associated with with features in the morphology library (O'Hara, 2008; Goodman, 2013). For adverb and adposition subordinators, morphological constraints are specified on the lexical type, and if there is no subordinator, these constraints are specified on the unary rule. Supported morhpological features include FORM, ASPECT, MOOD, NOMINALIZATION and user-defined syntactic features.

If NOMINALIZATION is among the specified features, we use a different set of lexical types and unary rules. The nominalized clauses library (Howell et al., to appear) allows users to define nominalization strategies in which nominalization occurs at the verb, verb phrase or sentence level and either adds a nominalized predication or not. This library changes clauses to [HEAD *noun*], adds the feature [NMZ +] and may add a nominalization predication that scopes over the event. Therefore, we define new lexical types and unary rules that select a clause that is [HEAD *noun*] [NMZ +].[9] Lexical nouns are constrained to [NMZ –], preventing the subordinator from selecting a lexical noun rather than a nominalized clause.

Shared Subjects If the subject is shared between the matrix and subordinate clauses, the embedded clause is constrained with an *unexpressed* element on its SUBJ list. The XARG (external argument; Copestake et al., 2005) of the subordinate clause is identified with the XARG of the subordinator's modifier, creating a semantic identity between the subjects of the subordinate and matrix verbs.

Special Word Order We use the analysis of Fokkens (2014) to accommodate verb second word order in the matrix clause and verb final in the subordinate clause. Under this analysis, a clause with subordinate word order is [MC −], which is compatible with our adposition lexical entry and unary rules.

Summary The analyses presented in this section build on the existing customization system, interaction with many different libraries, including word order, nominalization, aspect and mood, among others. We have accounted for the distribution of clausal modifiers, as described in §3, with the addition of two new lexical types and two unary rules to the Grammar Matrix. We create subtypes of those lexical types and unary rules that are further constrained according to Table 1 to account for the diversity of clausal modifier strategies in the typological literature.

[7] These rules are added by the word order library (Bender and Flickinger, 2005). In the rare case that the word order library does not add the necessary rule (for example, an otherwise head-final language has a head-initial subordinator), we add a special head-complement rule that selects for an adposition head daughter. Further detail is provided by Zamaraeva et al. (2018).

[8] This feature is then copied to the modifier list of the clausal modifier by the unary rule.

[9] If the nominalization strategy adds a nominalized_rel predication, the lexical type for adposition subordinators and unary rule for adverb or morphological subordination identify the subordinator's ARG2 with the local top of the _nominalized_rel, rather than the INDEX of the verb.

5 Customized Grammar Development

The Grammar Matrix customization system (Bender and Flickinger, 2005; Bender et al., 2010; Drellishak, 2009) uses a web-based questionnaire to elicit user input regarding the typological characteristics of their language on the front end and a customization script that produces a grammar that handles these phenomena on the back end. The two are linked via a 'choices' file, containing values that correspond to the user's input and can be read by the customization script. Thus to integrate the analysis described in §4, we first developed a clausal modifiers page in the questionnaire[10] to elicit the user's choices and then modified the customization script to add the appropriate analyses to the output grammar.

The clausal modifiers questionnaire uses an iterator so that users can define any number of clausal modifier strategies found in their language. For each strategy the user is asked about the clausal modifier's position and attachment and if the subordinating predication is contributed by a subordinator, a subordinator pair or if it does not involve a subordinator, as illustrated in (1), (2) and (3), respectively. Subsequent questions are based on this first choice.

If the user selects subordinator or subordinator pair, they are asked if the subordinator is an adverb or adposition. If they choose adposition, they are asked if it occurs at the beginning or end of the clause. If it's an adverb, they are asked whether it attaches strictly before, strictly after or freely before or after a VP, S or either a VP or S. If they select subordinator pair, they are also asked about the position of the matrix adverb. The user may enter any number of subordinators or subordinator-matrix adverb pairs, including an orthographic representation and a predicate symbol for each. If there is no subordinator, the user may enter only one predication per strategy. The user may add any number of verbal features associated with the clausal modifier strategy and can check a box to indicate subject sharing.

The choices file generated by the questionnaire goes through a script which validates if the choices are 'legal' and will result in a working grammar. If validation fails, the user is prompted to make changes; otherwise, the customization script adds a basic lexical type and/or unary rule to the grammar, based on the subordinator type, and creates a subtype of that lexical type and/or unary rule that is specific to the strategy. Constraints corresponding to the other choices are added to the subtypes, according to §4.4.

6 Testing, Evaluation and Error Analysis

We use two types of testing during development, which we follow with held out evaluation. Each test involves a testsuite, a choices file and a grammar produced by the customization system. The testsuites are small and designed to be representative of the relevant contrasts in the language, including the full range of possible grammatical sentences for each clausal modifier strategy as well as ungrammatical sentences that are each ungrammatical in one specific way. While our testsuites are small, they are robust in that they contain an example that targets each feature in Table 1 as well as any relevant interacting phenomena individually. We create choices files that define each clausal modifier strategy and account for interacting phenomena and load the resulting grammars into the LKB (Copestake, 2002) to parse each sentence. We inspect the parse trees and semantic representations to verify their correctness. [11]

6.1 Pseudo Languages

During development we tested our analyses on pseudo languages—artificial languages with a minimal lexicon which exhibit each of the phenomena outlined in Table 1 as well as known interacting phenomena. The typological space is large, including 1008 possible combinations for the features in Table 1.[12] Rather than test all combinations exhaustively, we created tests by sampling each constraint from the first column with each subordinator type, rather than each value, and added additional tests for interacting phenomena, including word order, verbal features and nominalization. To test the interaction between different strategies (which may result in overgeneration if under-constrained), we included multiple strategies in some tests. This resulted in 16 pseudo languages, containing a total of 33 distinct

[10] http://matrix.ling.washington.edu/customize/matrix.cgi?subpage=clausalmods

[11] Our code and testsuites can be checked out from svn://lemur.ling.washington.edu/shared/matrix/trunk at revision 41464.

[12] This is a conservative estimate, bundling many features and suppressing interacting phenomena, such as special word order, matrix adverb position, nominalization strategy and different types of verbal features. In reality, the space is much larger.

strategies. We refined our analysis during testing to achieve full coverage over these languages.

6.2 Development Languages

Next we tested our system on natural languages that illustrate particular phenomena in our library. We selected four languages, based on the characteristics their clausal modifiers exhibit and developed a testsuites of ten sentences for each, illustrating which characteristics of clausal modifiers are exhibited in the language and which would result in ungrammatical sentences, according to descriptive grammars.

Mandarin Mandarin [cmn] has a number of subordinator pairs, of which some subordinators can occur without their matrix adverb and some matrix adverbs are homophonous with conjunctions (Li and Thompson, 1989). We illustrated this range with one such pair *yīnwèi...suǒyǐ*, defining two clausal modifier strategies, one with the pair and one with just *yīnwèi* as a subordinator, and one coordination strategy,[13] with *suǒyǐ* as a conjunction.

Wambaya Wambaya [wmb] expresses purposive and prior clauses with a special purposive or prior morpheme on the verb (Nordlinger, 1998). Purposive can also be expressed with the infinitive suffix. These morphological strategies exhibit subject sharing and require dative instead of absolutive case on the object. 'When or because' and 'right after' clauses are finite clauses with no subordinator.[14] 'When or because' clauses attach strictly after the matrix clause, whereas 'right after' clauses attach strictly before. The inherent ambiguity from having both of these strategies in the same language is captured by our grammar. Our typological survey did not reveal case change outside of nominalized clauses, so it was not accounted for in our analysis. For this reason, our Wambaya testsuite has one grammatical sentence that does not parse, a purposive clause with dative case on the object, and one ungrammatical sentence that does parse, a purposive clause with absolutive case on the object.

Rukai We used Rukai [dru], as described by Zeitoun (2007), to test nominalization as a primary strategy for clausal modifiers. We tested two strategies, one in which the nominalization morpheme is also associated with 'reason' and another in which a generic nominalization morpheme is paired with a subordinator, meaning 'reason'. To capture the distinction between these morphemes, we defined a feature in customization that is suitable for both verbs and nouns (so that it will mark the clause both before and after nominalization), and associated it with each morpheme and each clausal modifier strategy.

German Finally German [deu] has verb final word order in the subordinate clause, while matrix clauses are verb second. We used a strategy with a clause initial subordinator from Thompson et al. (2007) to test this word order variation.

Table 2: Development Languages

Language	Family	Test Items	Coverage	Overgeneration
Wambaya [wmb]	Mirndi	10	5/6	1/4
German [deu]	Indo-European	10	2/2	0/8
Rukai [dru]	Austronesian	10	2/2	0/8
Mandarin [cmn]	Sino-Tibetan	10	6/6	0/4

We summarize our results on development languages in Table 2. The two errors are the result of case-frame changes, which are more closely related to verbal morphology than subordination. As a this is an interacting phenomenon and not specific to clausal modifiers we leave this out of scope for our library.

6.3 Evaluation with Held-out Languages

We evaluate on languages that are genetically and geographically diverse from the languages considered in development. These languages are chosen at random from the set of descriptive grammars available at the University of Washington library, and discarded if (a) they come from the same language family as an illustrative language or previous held-out language, (b) they do not contain (or the grammar does

[13] Using the coordination library contributed by (Drellishak and Bender, 2005)

[14] While this could be analyzed as juxtaposition, we illustrate the user's analytic freedom to treat these as subordinate clauses.

not describe) clausal modifiers or (c) the data does not contain a sufficient level of annotation to reliably construct additional examples. From the descriptive grammars, we develop testsuites that capture the range of clausal modifiers according to the linguist's description, constructing grammatical and ungrammatical sentences from examples in the descriptive grammar to contrast each characteristic of the clausal modifier strategy.[15] For each language we include up to four clausal modifier strategies, as described by the author, which may translate to between three and ten strategies in the choices file, depending on the type of variation therein.[16]

Ma'di Ma'di [mhi] has a wide range of clausal modifier strategies, including postposition subordinators, adverb subordinators (some requiring the subordinate clause to occur first, and others allowing it in either position) and subordinate clauses marked by the directive mood (Blackings and Fabb, 2003). Our analysis of adverb subordinators, based on the typological literature and a generalization that adverbials related to time and modality tend to attach higher in the tree (Cinque, 1999), only allows VP and S attachment. In Ma'di, however, adverb subordinators may intervene between the verb and object, suggesting V attachment as well. Since we do not support this, one sentence in our testsuite is not parsed.

Mosetén Strategies in Mosetén [cas], include subordinators in finite clauses, subordinators that require special morphology and strictly morphological subordination. Sakel (2004) states that the clausal modifier can occur in any position that an adverb can, but only shows S attachment in examples. The adverb chapter does not discuss the distribution of adverbs, but shows examples of S and VP attachment, so we infer that clausal modifiers may attach at the S and VP level. This testsuite revealed a bug in the customization code, that was not sampled in pseudo language evaluation. The unary rule supertype for morphological subordination is constrained to be [SUBJ $\langle\ \rangle$], barring VP attachment for morphologically subordinated clauses, and resulting the failure of two strings to parse.

Lavukaleve Subordination in Lavukaleve [lvk] is primarily morphological and various clausal modifier strategies position the subordinate clause differently (Terrill, 2003). Although Lavukaleve is a nominative-accusative language in general, it has a split ergative system in subordinate clauses, such that third person subjects take canonical object marking, but first and second person subjects take the usual subject marking. Alternate case frames in subordinate clauses are not modeled in the Grammar Matrix or by our library, so this results in one ungrammatical string parsing and one grammatical string failing to parse. Another interesting phenomenon in Lavukaleve, as described by Terrill (2003), is an adverb that occurs in the matrix clause when the subordinate clause is marked by special morphology, but no subordinator. This is not captured in our analysis of subordinator pairs, which require a subordinator, so another sentence in our testsuite fails to parse.

Basque Hualde and de Urbina (2003) provide an extremely thorough description of clausal modifiers in Basque [eus], including numerous clausal modifier strategies with different morphological forms that combine with various adpositions and adverbs. We do our best to select a representative set of combinations, including two sets of adpositions that co-occur with different morphemes, one adverb that co-occurs with another morpheme and a nominalization strategy whose meaning differs based on the case of the nominalized clause. This results in 10 clausal modifier strategies in the choices file and a special feature with 7 values (such as purposive, conditional, etc.) that is suitable for both verbal and nominal projections. Our test sentences contained dropped subjects which revealed that the nominalized clauses library does not provide a phrase structure rule for subject dropping in nominalized clauses. This results in three sentences failing to parse. However, upon constructing such a rule, we confirmed that if this rule were in the grammar, those sentences would parse correctly.

Uranina Finally, Uranina [ura] primarily uses subordinators (a clause final clitic and a number of postpositions) for clausal modifiers (Olawsky, 2006). The majority require a finite verb, but in the

[15] This is necessary for robust testing, as the descriptive grammar will often include only grammatical examples, but explain the full paradigm in prose.

[16] For example the choices file requires separate strategies for each morphologically contributed predication, even if they might be grouped under one strategy in the descriptive grammar.

innovative dialect (spoken by younger members of the community), some of these postpositions co-occur with non-finite verbs. There is also a subordinator pair for conditional clauses. We created three clausal modifier strategies, with multiple subordinators to successfully capture this variation.

Table 3: Held-out Language Evaluation

Language	Family	Test Items	Coverage	Overgeneration
Ma'di [mhi]	Central Sudanic	23	16/17	0/6
Mosetén [cas]	Mosetenan	26	13/15	0/11
Lavukaleve [lvk]	Solomons East Papuan	23	8/10	1/13
Basque [eus]	Basque	26	13/16	0/10
Uranina [ura]	Uranina	21	12/12	0/9

The average coverage over 5 test suites was 88.4% and overgeneration was 1.5%, as detailed in Table 3. With the exception of case frame change on embedded verbs in Wambaya and Lavukaleve, our library does a good job of preventing the parsing of ungrammatical strings, upholding our emphasis on precision. Our recall is lower, as we are not able to parse 8 strings, due to 5 errors, which we can classify in 3 groups. The first group includes bugs: the inability to parse VP attachment of clausal modifiers in Mosetén is due to a mistake in the supertype of the non-branching rule, and the lack of a subject dropping rule for nominalized verbs for Basque is due to an oversight in the nominalized clauses library. These bugs are easily fixed, now that they have been identified. The next group is out-of-scope phenomena: valence change (Wambaya and Lavukaleve) is more closely associated with verb form than clausal modifier strategy and should be handled in a library for valence change. The third type of error comes from phenomena that were not brought to light in the typological survey, but were found during held-out evaluation. First, we expected that adverb subordinators would be high-attaching, based on a cross linguistic generalization about adverb classes and the absence of any attested low-attaching adverbs in our literature review. Ma'di brings to light an interesting contradiction to this assumption and we will add the option of V attachment in the matrix clause now that it has been attested. Second, the typological literature did not suggest that matrix "pair" adverbs could occur without a subordinator in the embedded clause. If this is in fact the case in Lavukaleve, this poses a interesting direction for future work.

7 Conclusion

We presented a cross-linguistic HPSG analysis of clausal modifiers, which we implemented in the LinGO Grammar Matrix. We demonstrated the effectiveness of this library over the known typological space in §6.1, tested the application of specific clausal modifier strategies on real languages in §6.2, and evaluated its generalizability on 5 held-out languages in §6.3. Because our analysis is implemented within a larger grammar engineering environment, we were able to test its interaction with relevant phenomena. We contributed all testsuites and choices files back to the project so that they can be added to regression testing to ensure that downstream changes will not impact the coverage of this library and future developers can test their phenomena with clausal modifiers. The grammars produced by the Grammar Matrix with the addition of the clausal modifiers library are useful starting points to linguists who wish to develop broad coverage grammars, as the starter grammar can now include not only matrix clause phenomena, but analyses for subordinate clausal modifiers. In addition to developing broad coverage grammars, the grammars produced by the Grammar Matrix can be used to teach grammar engineering and for linguistic hypothesis testing, as our approach allows user-linguists analytic freedom in their implementation, so that multiple analyses can be explored.

Acknowledgements

We would like to thank Emily Bender for helpful discussion and Jiahui Huang and Paul Buechsenmann for grammaticality judgments for examples in our testsuites.

This material is based upon work supported by the National Science Foundation under Grant No. BCS-1561833 (PI Bender).

References

Emily M. Bender. 2008. Grammar engineering for linguistic hypothesis testing. In *Proceedings of the Texas Linguistics Society X conference: Computational linguistics for less-studied languages*, pages 16–36. Citeseer.

Emily M. Bender. 2010. Reweaving a grammar for Wambaya. *Linguistic Issues in Language Technology*, 3(1).

Emily M. Bender and Dan Flickinger. 2005. Rapid prototyping of scalable grammars: Towards modularity in extensions to a language-independent core. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing IJCNLP-05 (Posters/Demos)*, Jeju Island, Korea, 2005.

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 John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan, 2002.

Emily M. Bender, Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation*, pages 1–50. ISSN 1570-7075. URL `http://dx.doi.org/10.1007/s11168-010-9070-1`. 10.1007/s11168-010-9070-1.

Mairi John Blackings and Nigel Fabb. 2003. *A Grammar of Ma'di*, volume 32. Walter de Gruyter.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The parallel grammar project. In *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Guglielmo Cinque. 1999. *Adverbs and functional heads: A cross-linguistic perspective*. Oxford University Press on Demand.

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

Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 140–147. Association for Computational Linguistics.

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.

John Frederick Davis. 1973. *A partial grammar of simplex and complex sentences in Luiseño*. PhD thesis, University of California.

Eric Villemonte de La Clergerie. 2005. From metagrammars to factorized tag/tig parsers. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 190–191. Association for Computational Linguistics.

Scott Drellishak. 2009. *Widespread but not universal: Improving the typological coverage of the Grammar Matrix*. PhD thesis, University of Washington.

Scott Drellishak and Emily Bender. 2005. A coordination module for a crosslinguistic grammar resource. In *International Conference on Head-Driven Phrase Structure Grammar*, volume 12, pages 108–128. URL `http://web.stanford.edu/group/cslipublications/cslipublications/HPSG/2005/drellishak-bender.pdf`.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1) (Special Issue on Efficient Processing with HPSG):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.

Antske Sibelle Fokkens. 2014. *Enhancing Empirical Research for Linguistically Motivated Precision Grammars*. PhD thesis, Department of Computational Linguistics, Universität des Saarlandes.

Michael Wayne Goodman. 2013. Generation of machine-readable morphological rules from human-readable input. *Seattle: University of Washington Working Papers in Linguistics*, 30.

Kristen Howell, Olga Zamaraeva, and Emily M. Bender. to appear. Nominalized clauses in the Grammar Matrix. In *The 25th International Conference on Head-Driven Phrase Structure Grammar*.

José Ignacio Hualde and Jon Ortiz de Urbina. 2003. *A grammar of Basque*, volume 26. Walter de Gruyter.

Paul Kingsbury and Martha Palmer. 2002. From Treebank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 1989–1993, Las Palmas, Spain, 2002. Citeseer.

Charles N Li and Sandra A Thompson. 1989. *Mandarin Chinese: A functional reference grammar*. University of California Press.

Montserrat Marimon. 2010. The Spanish resource grammar. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 700–704, Valletta, Malta, 2010.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 92–97.

W Detmar Meurers, Gerald Penn, and Frank Richter. 2002. A web-based instructional platform for constraint-based grammar formalisms and parsing. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, pages 19–26. Association for Computational Linguistics.

Stefan Müller. 2015. The CoreGram project: Theoretical linguistics, theory development and verification. *Journal of Language Modelling*, 3(1):21–86. URL `https://hpsg.hu-berlin.de/~stefan/Pub/coregram.html`.

Michael Noonan. 2007. Complementation. In Timothy Shopen, editor, *Language typology and syntactic description*, volume 2: Complex constructions, pages 52–150. Cambridge University Press.

Rachel Nordlinger. 1998. *A Grammar of Wambaya, Northern Australia*. Pacific Linguistics.

Kelly O'Hara. 2008. A morphotactic infrastructure for a grammar customization system. Master's thesis, University of Washington.

Knut J. Olawsky. 2006. *A grammar of Uranina*, volume 37. Walter de Gruyter.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Gerald Penn. 2004. Balancing clarity and efficiency in typed feature logic through delaying. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 239–246. Association for Computational Linguistics.

Carl Pollard and Ivan A Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Laurie Poulson. 2011. Meta-modeling of tense and aspect in a cross-linguistic grammar engineering platform. *University of Washington Working Papers in Linguistics (UWWPL)*, 28.

Aarne Ranta. 2004. Grammatical framework. *Journal of Functional Programming*, 14(2):145–189.

Jeanette Sakel. 2004. *A grammar of Mosetén*, volume 33. Walter de Gruyter.

Melanie Siegel, Emily M Bender, and Francis Bond. 2016. *Jacy: An implemented grammar of Japanese*. CSLI Publications.

Angela Terrill. 2003. *A grammar of Lavukaleve*, volume 30. Walter de Gruyter.

Sandra A Thompson, Robert E Longacre, and Shin Ja J Hwang. 2007. Adverbial clauses. In Timothy Shopen, editor, *Language typology and syntactic description*, volume 2: Complex constructions, pages 237–269. Cambridge University Press.

Thomas James Trimble. 2014. Adjectives in the LinGO Grammar Matrix. Master's thesis, University of Washington.

Olga Zamaraeva, Kristen Howell, and Emily M. Bender. A cross-linguistic account of subordinator and subordinate clause position. Poster to be presented at The 25th International Conference on Head-Driven Phrase Structure Grammar, 2018.

Elizabeth Zeitoun. 2007. *A grammar of Mantauran (Rukai)*. Institute of Linguistics, Academia Sinica Taipei.