Automatic Extraction of Clausal Embedding Based on Large-Scale English Text Data

Iona Carslaw[*,1,2], **Sivan Milton**[*,1,2], **Nicolas Navarre**[*,1,2],
Ciyang Qing[2], **Wataru Uegaki**[2]

[1]School of Informatics, University of Edinburgh
[2]School of Philosophy, Psychology & Language Sciences, University of Edinburgh

{ I.C.A.Carslaw, s.milton, n.s.navarre}@sms.ed.ac.uk   {cqing, w.uegaki}@ed.ac.uk

Abstract

For linguists, embedded clauses have been of special interest because of their intricate distribution of syntactic and semantic features. Yet, current research relies on schematically created language examples to investigate these constructions, missing out on statistical information and naturally-occurring examples that can be gained from large language corpora. Thus, we present a methodological approach for detecting and annotating naturally-occurring examples of English embedded clauses in large-scale text data using constituency parsing and a set of parsing heuristics. Our tool has been evaluated on our dataset Golden Embedded Clause Set (GECS), which includes hand-annotated examples of naturally-occurring English embedded clause sentences. Finally, we present a large-scale dataset of naturally-occurring English embedded clauses which we have extracted from the open-source corpus *Dolma* using our extraction tool.

1 Introduction

One of the most popular methods of conducting linguistic research has consisted of handcrafting paradigmatic utterances followed by gathering native speakers' judgements. Yet, it is questionable how much these constructed utterances reflect real-world language use. As a result, plenty of debate has arisen about the legitimacy of paradigmatic utterances as a research tool, with arguments suggesting this particular data collection technique can lead to biased results (Cowart, 1997; Schütze, 2016). Whilst this debate has been happening in linguistics, the advancements of Natural Language Processing (NLP) have led to a significant increase in the amount of freely available language corpora as well as an increase in their size. For example, the open-source dataset *Dolma* consists of 3 trillion English tokens (Soldaini et al., 2023). These datasets provide new opportunities for linguistic research, with the ability to gather statistical data about specific language constructions and naturally-occurring examples beyond handcrafted sentences.

One particular sentence construction that would benefit from such corpus research is that of EMBEDDED CLAUSES. These constructions contain an embedding predicate which selects a clausal complement, as seen in the sentence: *Mary hopes that John likes chocolate*. Here, the predicate *hopes* embeds the declarative clausal complement *that John likes chocolate*. Alongside DECLARATIVE clausal complements, as in (1a), there are also POLAR INTERROGATIVE clausal complements (1b), ALTERNATIVE INTERROGATIVE clausal complements (1c), and CONSTITUENT INTERROGATIVE clausal complements (1d). Crucially, predicates vary with respect to which clausal complement type they are allowed to embed; consider the difference in grammaticality between *wonder*, which can embed interrogative clausal complements, and *hope*, which cannot embed interrogative clausal complements.[1] In addition, it has been observed that emotive factives, such as *be happy (about)*, take declarative and constituent interrogative complements but not polar and alternative interrogative complements (Abels, 2004; Karttunen, 1977; Sæbø, 2007, a.o.).

(1)   a.   Mary {*wondered | hoped | was happy } [that John liked chocolate].

   b.   Mary {wondered | *hoped | *was happy about } [whether John liked chocolate].

   c.   Mary {wondered | *hoped | *was happy about } [whether John liked chocolate or cake].

   d.   Mary {wondered | *hoped | was happy

---

[1]These judgements, commonly reported in the literature, are shared by the 3 native British English and Canadian English speakers among the authors.

about } [which chocolate John ate].

Because of this observation that a predicate selects for particular types of embedded clause in fine-grained ways - partly conditioned by the predicate's lexical semantics - there is a debate amongst syntacticians and semanticists about what roles syntax and semantics play within these constructions (Grimshaw, 1979; Uegaki and Sudo, 2019; White, 2021, a.o.). Extrapolating clausal embeddings from large-scale corpora would help to answer such questions, by providing large-scale statistical evidence for how often these embedding predicates appear in natural language use and what clausal complements they select, as well as the ability to look for natural language examples. Thus, the aim of this paper is to create a tool for linguists to extract English sentences containing embedded clauses from large-scale corpora, whilst also providing the following information: (i) the span of the embedded clause, (ii) the lexeme(s) of the embedding predicate, and (iii) the type of the embedded clause.[2]

This task of extracting embedded clauses is by no means trivial. Firstly, the span of the embedded clause in a sentence has to be correctly identified, excluding any element that belongs to the matrix clause. Secondly, there are constructions that superficially resemble embedded clauses, but are in fact not, as they fail to categorise syntactically as *complements* of an embedding predicate or as *clauses*. To see this, consider the following examples:

(2) a. Mary saw a man [that John mentioned].

    b. Mary ate [what John cooked].

    c. Mary goes to the gym regardless of [whether she is tired or not].

The bracketed clause in (2a) is a RELATIVE CLAUSE and is not a complement of an embedding predicate. In (2b), we have an instance of a FREE RELATIVE, which is considered as primarily a Noun Phrase rather than a clause (Caponigro, 2003; van Riemsdijk, 2006). The bracketed clause in (2c) is an UNCONDITIONAL (Rawlins, 2008), which is a modifier rather than a complement of a matrix predicate. Thirdly, embedded clauses can arise in complex clausal structures such as coordination (3a), which often occurs with ellipsis, nest-

ing (3b), or some combination of both (3c). Consequently, to correctly identify embedded clauses, we need a correct syntactic parse of the sentence, as well as appropriate heuristics to rule out structures such as those in (2) and deal with the structures in (3).

(3) a. Mary knows [that John likes chocolate] and [that Mark does not].

    b. Mary knows [that John thinks [that Mark likes chocolate]].

    c. Mary knows [that John thinks [that Mark likes chocolate]] and [that Mark does not].

Our paper is structured in the following way: Section 2 describes previous attempts at building a large-scale corpora of English embedded clauses (e.g. MegaAcceptability), and additionally examines existing tools designed to extract sentences from language corpora (e.g. linguistic search engines). Section 3 introduces our hand-annotated dataset of English embedded clauses: Golden Embedded Clause Set (GECS). Section 4 describes our extraction tool that uses constituency representations and parsing heuristics, as well as our tool's performance on GECS. Section 5 presents the large English embedded clause dataset that we have extracted from the open-source dataset *Dolma*. Section 6 suggests future research avenues and 7 concludes our work. Overall, we provide three new contributions:

1. A small-scale dataset (GECS) with fine-grained gold standard annotation of English embedded clauses to be used as a benchmark for this task

2. An extraction tool which can be applied to English language corpora to extract and annotate embedded clauses

3. A large-scale extracted set of English embedded clauses from the language corpus *Dolma* for the linguistic community to use

2 Relevant Work

2.1 MegaAcceptability

The only existing attempt at a large dataset of English embedded clauses is the MegaAcceptability dataset (White and Rawlins, 2016, 2020). White and Rawlins selected a list of 1007 English verbs

that are known to select clausal embeddings, and then designed 50 schematic sentences covering a range of syntactic environments in which an embedded clause can occur. They then slotted the 1007 verbs into the 50 schematic sentences to create $\sim 50,000$ entries. Through Amazon MTurk, participants rated the acceptability of the resultant sentences, leading to a large dataset of embedded clause constructions ranked by acceptability, on a 7-point ordinal scale.

Although the MegaAcceptability dataset moves away from the problem of a small set of sentences being used as evidence for linguistic hypotheses, it still utilises non-natural sentences which have been handcrafted. Furthermore, for finite embedded clauses White and Rawlins (2016; 2020) only considered environments without complementisers or with the following complementisers: *that, whether*, and *which*. They also only consider predicates with no prepositions or with the following prepositions: *to* and *about*. They make use of a pre-defined list of verbs which accept clausal complements, which does not account for the full set of embedding verbs nor adjectives and complex predicates which can also accept clausal complements. Therefore, it is unclear if the dataset captures the natural distributions of embedding predicates, embedded clause types, and the types of embedded clauses selected by embedding predicates.

2.2 Linguistic Search Engines

The goal to extract sentences with certain linguistic phenomena from natural language use is not a new concept. There have been several attempts to create *search engines* in which an individual can query annotated natural-language corpora for certain constructions and then be provided with a list of sentences which match the provided query. Prominent tools with this use include the Linguist's Search Engine (Resnik and Elkiss, 2005), SPIKE (Shlain et al., 2020), and the LINDAT/CLARIAH-CZ PML Tree Query (Pajas et al., 2009).

Although these are powerful tools, their query languages are not sufficiently fine-grained to capture the relevant structures of embedded clauses. They rely on annotation of corpora with lemmas, part-of-speech tags, and dependency graph representations. This means that one would need to specify dependency relationships rather than constituency/hierarchical ones to identify the structure of embedded clauses. Such an approach is limiting, as it is difficult to identify clause and predicate
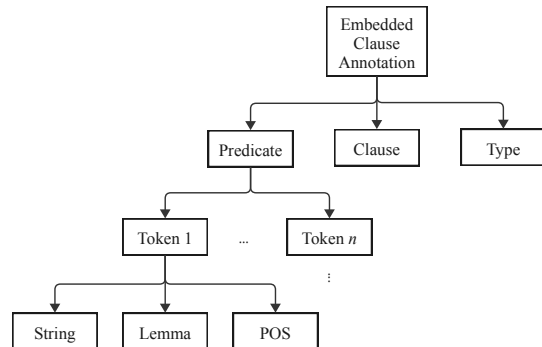

Figure 1: The annotation in GECS for each embedded clause.

spans based on dependency relations or linear structure. There is also less consistency with respect to the relations that identify embedded clauses than with constituency parsers. Moreover, linguistic search engines offer linguists limited flexibility to decide which corpora they want to extract sentences from.

## 3 Golden Embedded Clause Set (GECS)

For the novel task of English embedded clause detection in natural language corpora, we created a hand-annotated dataset (GECS) which can serve as a benchmark for evaluation and be used in its own right for a small-scale analysis of embedded clause constructions. In GECS, each embedded clause is annotated with its embedding predicate, clause span, and clause type (see Figure 1). We provide the embedding predicate as a list of the relevant tokens (i.e. ignoring negation words, adjuncts, and any other tokens which may appear between the first embedding predicate token and the clause).[3]

Annotation Procedure To create our naturally-occurring embedded clause dataset, we selected a subset of $866,538$ sentences from *Dolma*[4] (Soldaini et al., 2023). The data was not cleaned so as to accurately test the robustness of the tool. We then parsed the sentences and filtered them to remove any which necessarily did not contain embedded clauses.[5] To extract the set of polar and

---

[3]To capture the complex constructions of coordinated and nested embedded clauses alluded to in Section 1, we optionally provide a recursive data structure version of GECS which makes the internal clausal structure transparent (see Figure 4 in the Appendix).

[4]Specifically we used the files *cc_en_head-0000*, *cc_en_head-0001*, and *c4-0085*.

[5]We used SBAR from SpaCy's Berkeley Neural Parser (Kitaev et al., 2019; Kitaev and Klein, 2018) as an indicator,
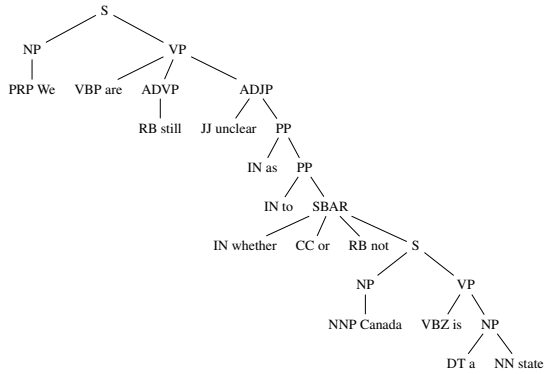
(a) Complex embedding predicate ('are unclear as to')
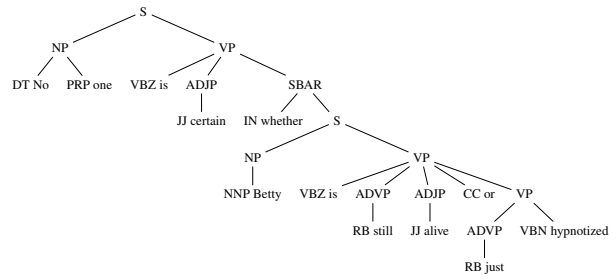
(b) Adjectival predicate ('is certain')

Figure 2: Example parses of embedded clause sentences in GECS.

alternative interrogative embedded clauses, we further filtered out sentences that did not contain the words *whether* or *if*. Finally, to filter for constituent interrogative embedded clauses, we only considered sentences with: *who*, *what*, *when*, *where*, *why*, *how*, or *which*. The next stage of hand annotation consisted of one researcher going through the pre-filtered sentences and confirming (i) if there were embedded clauses and (ii) if so, providing the annotation of predicate tokens, clause span, and type. A second researcher then went through the previous researcher's annotations to confirm agreement.

Overall, GECS contains 147 declarative embedded clauses, 138 polar interrogative embedded clauses, 84 alternative interrogative embedded clauses,[6] and 158 constituent interrogative embedded clauses. In addition, we provide a set of 111 adversarial examples verified to not contain any embedded clauses, but do contain misleading structures such as free relatives and relative clauses. These were created by selecting sentences discarded by the annotators in the final stage of GECS' creation.

4 Parser Tool

Though it is possible to define a set of heuristics based on Regular Expressions or dependency relations, preliminary analysis indicates significant disadvantages to such an approach, as were seen with the linguistic search engines from Section 2. For this reason, we opted for representations from constituency syntactic parsers to extrapolate hierarchical structure. A benefit of this choice is that

linguistic theory is typically given with respect to constituency trees, and we can therefore implement linguistic facts into extraction heuristics more freely than with other representations. While it is possible that a Dependency Parser could be used to achieve equivalent results, it is not clear what improvements it could offer. We leave this question open to a more thorough exploration in future. With the constituency representation we defined a set of heuristics to perform the following tasks:

1. **Detection**: detecting embedded clause(s) in a sentence
2. **Predicate Identification**: identifying each embedding predicate
3. **Clause Identification**: identifying the span of each embedded clause
4. **Typing**: identifying the type of each embedded clause

The syntactic parser that we use is SpaCy's Berkeley Neural Parser, a constituency parser that has an LSTM and self-attentive architecture (Kitaev and Klein, 2018; Kitaev et al., 2019). Other options are available for constituency parsing; however, we decided upon this parser because it is state-of-the-art for constituency parsing.

The SpaCy constituency parser represents each sentence as an n-ary tree structure with several syntactic categories (e.g. S, VP, NP, SBAR) in parent and child hierarchy. This tree structure is particularly helpful in extracting embedded clauses because we can traverse the parent levels and check for particular child nodes in complement positions. We then defined heuristics based on the structures from the parser to perform the aforementioned tasks of embedded clauses detection, predicate identification, clause identification, and typing.

---

assuming that this structure in the parsed representation is a necessary condition for an embedded clause.

[6]There are fewer alternative interrogative embedded clauses due to this type being far sparser in the pre-filtered dataset than the other types.

4

4.1 Methodology

Detection   The first heuristic we deemed necessary for detecting embedded clauses is the existence of an SBAR in the parsed representation. This is a syntactic category for a subordinate clause, which is a superset of embedded clauses, but also includes non-embedded clauses like relative clauses. To check if a subordinate clause is an embedded clause, we assume it needs to be dominated by a VP headed by a predicate. While there may be other syntactic categories immediately above the subordinate clause, we are only interested in the first upstream occurrence of one of two syntactic categories: NP or VP. In the case where the label is VP, the sentence has an embedded clause. If the label is NP, then the sentence does not have an embedded clause—likewise if neither of the two are found until the root node of the tree. We use the hierarchical nature of the constituency parser to distinguish embedded clauses from relative clauses and complements of NPs.

To limit the amount of false positives that would be extracted from the dataset we implemented a few heuristics based on the embedding predicate and the subordinating conjunction of the clauses that are detected. First, if the embedding predicate is empty after the part-of-speech filtering or the only predicate token is 'is/be', then the clause is not considered to be an embedded clause. Secondly, we rule out any clauses beginning with certain subordinating conjunction because they are not indicative of an embedded clause. Specifically, we blacklist the following: *after, although, before, despite, to, for, so, though, unless, until, than, because, since, while, as, even if, in order*.

Predicate Identification   Having identified an embedded clause in a sentence, we can extract the embedding predicate from the sentence by searching for the nearest VP parent of the clause. We iteratively search through the parents of the embedded clause until a VP parent is reached. We then identify the predicate span from this constituent, considering a wider range of possible verbs, adjectives, and prepositions than previous methods (cf. Section 2.1). For each constituent child of the VP (with exception to the final one which contains the embedded clause) we keep every token in the child span as long as the child label is either a PP, NP or SBAR label. For the last child of the VP, we keep every token until the onset of the embedded clause. We then filter these tokens based on their part-of-speech tags. We keep only the tokens that are VERB, ADP or ADJ, with the exception for an auxiliary tag AUX if there is also an adjective in the original token list. This helps us capture adjectival predicates such as 'unclear as to' or 'is certain' (see Figure 2).

Clause Identification   Given that a sentence is detected as having an embedded clause, we can then further use the parsed representation to extract the span of the embedded clause. The constituency parser is advantageous in this regard as we take whatever is under the syntactic label of SBAR to be the embedded clause constituent.

Typing   Having identified the clause span, the heuristics for typing the clause can involve more simple string matching. For alternative interrogative clauses we check the complementiser. If the complementiser *whether* is in the first word of the embedded clause along with the token *or*, then it is an alternative interrogative. If instead, we find *whether* that is not followed by the token *or* or is followed by the explicit string *or not*, then the embedded clause is a polar interrogative. If a unique token of either *which, who, what, when, where, why,* or *how* is the first word of the embedded clause, then it is a constituent interrogative clause. If none of the prior conditions are met, including if the clause begins with *that*, then we type the clause as declarative.

4.2 Evaluation

We evaluate the performance of our tool on the sentence annotations in GECS. With these annotations we can accurately test the tool's ability to detect embedded clauses, embedding predicates, and clause types, allowing us to evaluate how our tool handles messy natural data. We have also built a pattern matching baseline to compare our heuristics against a more linear approach.

Pattern Matching Baseline   The baseline we constructed is a rule-based tool using pre-defined lexical patterns to extract embedded clause annotations from a sentence. This method relies on the SpaCy Matcher, a tool which is similar to Regular Expressions in that it matches a given pattern in a string, but with useful supplemental linguistic information encoded, such as POS and lemma (Honnibal and Montani, 2017)[7]. In order to de-

---

[7]We opted for a Regular Expression matcher baseline due to its straightforward implementation for this task, as com-

tect embedded clauses, the Matcher is provided with the fixed list of (potentially) embedding predicates from MegaAcceptability (White and Rawlins, 2016). It then returns instances of these predicates in a sentence; with an added heuristic ensuring that the predicate is followed by some other verb or auxiliary (i.e. a clause), an embedded clause is identified. Prepositions proceeding the verb are included in the list of predicate tokens and POS and lemmas are also identified. Limited by the linear nature of the Matcher, we define the clause span as the end of the predicate to the end of the sentence, ignoring any adverb/pronoun which may occur between a predicate and clause. For the final goal of typing the embedded clause we again use the Matcher to match the first token of the clause to the associated complementisers for each type. We distinguish between polar and alternative interrogatives by classifying clauses containing the token *or* but not the string *or not* as alternative, and every other instance as polar. If no associated complementiser is found in the clause, the clause is typed as declarative.

Detection	Baseline			Parser Tool		
	Precision	Recall	F1	Precision	Recall	F1
Single	0.54	0.94	**0.69**	0.90	0.94	**0.92**
Multi	0.74	0.85	**0.79**	0.94	0.83	**0.88**
Overall	0.54	0.91	**0.68**	0.90	0.91	**0.91**

Table 1: Precision, Recall, and F1 scores for clause detection in GECS.

Task	Baseline	Parser Tool
Predicate Identification	0.79	0.91
Clause Identification	0.50	0.87
Type Identification	0.94	0.96

Table 2: Identification accuracy scores evaluated on the true positives set of detected clauses from Table 1.

Results We split the evaluation of our tool and the baseline into three detection sections: *Single Clause Evaluation* which evaluates detection performance on sentences in GECS that only included one embedded clause, *Multi Clause Evaluation* which evaluates detection performance on sentences in GECS that had multiple embedded clauses (nested and coordinated clauses), and *Overall* which combines the statistics of single and multi clause evaluation and performance on the adversarials. Table 1 provides the precision and recall for

these metrics. We also evaluated amongst the correctly detected embedded clauses the annotation abilities of our tool and the baseline, by seeing if the selected predicate is correct (*Predicate Identification*), if the selected clause is correct (*Clause Identification*), and if the typing of the clause is correct (*Type Identification*). Table 2 provides the accuracy scores for these metrics.

As Table 1 and 2 shows, we outperform the baseline in every metric, indicating that our method of utilising a constituency based tool is better than a linear based approach. Our tool only slightly degrades in detection recall when given a sentence that had nested and/or coordinated embedded clauses.

Failure analysis In the few cases of our tool's error, we see the following categories: parser errors, unconditionals mistaken as embedded clauses, and incomplete complex predicate detection. The parser error was the biggest issue for failed cases - unfortunately this is an unavoidable error given that any parser will be imperfect. Unconditionals also proved a problem because they are parsed the same as an embedded clause, and therefore are impossible to differentiate from one another. Finally, complex predicates were sometimes incompletely detected so not all of the predicate tokens are placed in the entry. Given that some of these errors are unavoidable, coupled with the tool's high precision and recall, we still take the results to indicate that our tool can be used to create a large-scale dataset of naturally-occurring embedded clauses, as long as researchers propagate the error into their analysis - something which needs to be done with any corpus study.

5 Case Study: Large-Scale Dataset

Having designed a tool which can identify and annotate embedded clauses, we applied it to an English corpus to create a large-scale dataset of annotated embedded clauses. We chose to apply the tool to a subset of *Dolma*[8] (Soldaini et al., 2023). Overall, $28,968,073$ embedded clauses were detected.

5.1 Comparison with MegaAcceptability

In order to compare with MegaAcceptability, we performed a limited case study on our large-scale dataset by only looking at our dataset entries that

included the 1007 verbs that were used in the MegaAcceptability templates. To get the rating of each verb from MegaAcceptability, we selected the maximum normalised rating of that verb's available constructions. We compared the distribution between the acceptability rating of a verb according to MegaAcceptability and its frequency in the large-scale dataset. It would be generally expected that the higher a verb is rated the more frequent it would be. As shown in Figure 3, this is the overall trend that we see. This means that our tool has successfully captured the verbs with the highest acceptability, while the verbs with lower acceptability had a lesser chance of occurring with embedded clauses.

There are some exceptions to the frequency-acceptability distribution, however this provides an interesting exploration point. For instance, the low acceptability outlier which has a high frequency in Figure 3 is the predicate *mean*. Looking at entries with *mean* as the predicate, we see three example types: (i) where it is unclear if the predicate is actually embedding or is acting as some filler (4a), (ii) false positives (4b), and (iii) true embedded clauses (4c). Thus, *mean* could be an outlier because of false positives, or it could be an outlier due to a data-driven approach collecting sentence clause types which a template approach could not.

(4)   a.   It's pretty catchy, I mean who doesn't go ANN ANN and A SORE.

      b.   In Glosa it means "what I've just said".

      c.   This means [...], the ADA applies to you.

5.2   Clauses and Predicates at Scale

With our large scale dataset of embedded clauses we can look beyond the fixed list of predicates as would be provided by a template-driven dataset like MegaAcceptability. With our approach we are able to view the clause-predicate distribution at a grand scale to test and verify linguistic theories. From the nearly 29 million embedded clause examples in the dataset we have the following distribution of clause types: $19,195,112$ declarative clauses, $9,402,868$ constituent interrogative clauses, $261,274$ polar interrogative clauses, and $108,819$ alternative interrogative clauses. This shows us how rare polar and alternative interrogative clauses are. Moreover, we can examine to the distribution of embedding predicates in the dataset. Taking a look at the part-of-
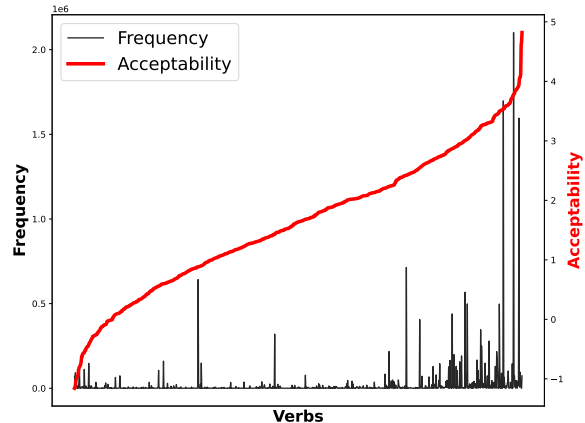

Figure 3: Comparison of natural data frequency and acceptability of the verbs found in MegaAcceptability ranked in increasing order of acceptability

speech tags for each of the embedding predicates in the dataset we can observe the distribution of adjectival and verbal predicates. Adjectival predicates require an accompanying verb or auxiliary (e.g., *be happy*), so we look at complex predicates involving two or more tokens. We find that there are $35,294$ unique adjectives within these complex predicates. Meanwhile, for simple one-word verbal predicates, we find $29,654$ unique predicates. Altogether, this leaves us with a strong set of examples to analyse any clause-predicate distribution of interest.

Here we present an example of how the dataset can be used in linguistic research to further validate and verify linguistic theories, as well as survey new possible sentence constructions that could be of interest.

Emotive Factive Predicates   As mentioned in Section 1, previous analyses have shown that emotive factive predicates, such as *be happy*, or *be glad*, are not able to embed either polar or alternative interrogative clauses (Karttunen, 1977; Abels, 2004; Sæbø, 2007). We can see if the extracted dataset shows this distribution statistically and if there are any counter-examples.

To test this generalisation, we selected a subset of emotive factive predicates to investigate further: *happy, amazed, sad, glad, excited, surprised, incredible, angry, mad, jealous, afraid*. Looking at the clauses that are embedded with these predicates, we get the following distribution of clause types: $175,479$ declarative clauses, $47,877$ constituent interrogative clauses, 159 polar, and 134 alternative interrogative clauses. The statistical breakdown does match the generalisation, with declarative and

constituent interrogative embedded clauses being the more popular embedded clause type. However, more importantly, there are some polar and alternative interrogative, of which we can search through to find potential counter-examples to the generalisation.

In searching through the polar and alternative interrogative embedded clauses, many are false positives, with the following four errors being indicative of the set: unconditionals (5a), wrong predicate span where the emotive factive is not the embedding predicate (5b), real embedded clauses but the sentence does not have the intended meaning required by the generalisation, e.g., *be afraid* is non-factive in (5c), and clausal adjuncts (5d).

(5) a. It's not your problem, because you're happy whether you're with him or doing stuff on your own.

b. I'm not sure how excited to get about this fund and whether he's just piggybacking on the Buffett name.

c. We are afraid whether it will be in Sindhi interest.

d. Meanwhile, people across the state are hair-on-fire mad over whether urban water users should be allowed to buy rural property simply for the water rights, and whether some water users should be allowed to sell their water to others out of state.

Given that we need to propagate the tool's error rate, this is to be expected. However, there appears to be some genuine counter-examples (6), of which at least two of the three native speakers among the authors find grammatical. It is beyond the scope of this paper to provide an analysis of these sentences, so we leave it for future work.

(6) a. In the post you talk about your child's health issues and in the end ask if people are happy with whether they're circumcised or not.

b. You might be surprised about whether there's hope for future shooters.

Although this analysis is not exhaustive in the least, we use these examples to motivate the use of this dataset to further validate and explore linguistic theories through naturally-occurring linguistic data in addition to handcrafted templatic examples.

6 Discussion

As this is the first method at extracting embedded clauses from natural language corpora, we set out some future research avenues to be undertaken. Firstly, clausal embedding extraction should be extended to other languages so that linguistic theories using such large corpora can have crosslinguistic validity. Given that the universal definition of a clausal complement is a complement to VP, we argue that a similar method to what we have described in the paper can be taken with other languages. The main changes would be to the fine-grain heuristics that we used for typing. Of course, our approach is subject to the limitations that follow from any corpus-based research, which introduces its own set of biases pre-existing in the corpora. It is also not always possible to scale this approach crosslinguistically, as the method relies on a given language having large enough corpora (which many do not). Nonetheless, this should not deter people from using the method with an applicable language, in complement with other approaches. Secondly, we recognise that there are other potential methods for extracting English clausal embeddings. One such technique is the use of an LLM, a method which we decided against given that an LLM is a blackbox, meaning a thorough error analysis would not be able to be conducted. To aid future development within this area, we have provided GECS to be used as a benchmark for this task.

7 Conclusion

The availability of large natural-language corpora has led to an opportunity for linguists to conduct large-scale language studies. However, extracting specific language constructions from such large corpora is a difficult endeavour. A particular construction in need of a large-scale corpus study is that of embedded clauses. Thus, we have made three contributions in aid of fulfilling this need. Firstly, we have created GECS, a small-scale dataset with fine-grained gold standard annotation of embedded clauses to be used as a benchmark for embedded clause extraction in English. Secondly, we created a tool which can be applied to English natural language corpora to detect and annotate embedded clauses. And finally, we provided a large-scale extracted set of English embedded clauses from the natural language corpus *Dolma* for the linguistic community to use.

References

Klaus Abels. 2004. Why *surprise*-predicates do not embed polar interrogatives. *Linguistische Arbeitsberichte*, 81:203–222.

Ivano Caponigro. 2003. *Free not to ask: On the semantics of free relatives and wh-words cross-linguistically*. Ph.D. thesis, University of California, Los Angeles.

Wayne Cowart. 1997. *Experimental Syntax: Applying Objective Methods to Sentence Judgments*, 1 edition. SAGE Publications.

Jon Robert Gajewski. 2007. Neg-raising and polarity. *Linguistics and Philosophy*, 30:289–328.

Jane Grimshaw. 1979. Complement selection and the lexicon. *Linguistic inquiry*, 10(2):279–326.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Larry Horn. 1978. Remarks on neg-raising. *Syntax and Semantics*, 9.

Lauri Karttunen. 1977. Syntax and semantics of questions. *Linguistics and philosophy*, 1:3–44.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Clemens Mayr. 2019. Triviality and interrogative embedding: context sensitivity, factivity, and neg-raising. *Natural language semantics*, 27(3):227–278.

Petr Pajas, Jan Štěpánek, and Michal Sedlák. 2009. PML tree query. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Kyle Rawlins. 2008. *(Un)conditionals: An investigation in the syntax and semantics of conditional structures*. Ph.D. thesis, University of California, Santa Cruz.

Phillip Resnik and Aaron Elkiss. 2005. The linguist's search engine: An overview.

Kjell Johan Sæbø. 2007. A whether forecast. In *Logic, Language, and Computation: 6th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2005 Batumi, Georgia, September 12-16, 2005. Revised Selected Papers 6*, pages 189–199.

Carson T. Schütze. 2016. *The empirical base of linguistics : grammaticality judgments and linguistic methodology*. Language Science Press, Berlin.

Micah Shlain, Hillel Taub-Tabib, Shoval Sadde, and Yoav Goldberg. 2020. Syntactic search by example. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 17–23.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Pete Walsh, Hannaneh Hajishirzi, Noah A. Smith, Luke Zettlemoyer, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2023. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

Nadine Theiler, Floris Roelofsen, and Maria Aloni. 2019. Picky predicates: why believe doesn't like interrogative complements, and other puzzles. *Natural language semantics*, 27(2):95–134.

Wataru Uegaki and Yasutada Sudo. 2019. The* hope-wh puzzle. *Natural Language Semantics*, 27(4):323–356.

Henk van Riemsdijk. 2006. Free relatives. In Martin Everaert and Henk van Riemsdijk, editors, *The Blackwell companion to syntax*, volume 2, pages 338–382. Blackwell, Oxford.

Aaron Steven White. 2021. On believing and hoping whether. *Semantics and pragmatics*, 14(6):1–21.

Aaron Steven White and Kyle Rawlins. 2016. A computational model of s-selection. *Proceedings from Semantics and Linguistic Theory*, 26(26):641–663.

Aaron Steven White and Kyle Rawlins. 2020. Frequency, acceptability, and selection: A case study of clause-embedding. *Glossa (London)*, 5(1):1–40.

Hedde Zeijlstra. 2018. Does neg-raising involve neg-raising? *Topoi*, 37(3):417–433.

Richard Zuber. 1982. Semantic restrictions on certain complementizers. In *Proceedings of the 13th International Congress of Linguists*, pages 434–436. Tokyo.
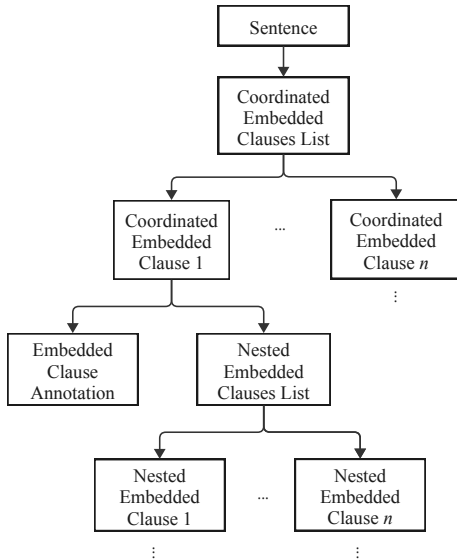
A   Data Structure

Figure 4: The data structure in GECS for coordination and nesting of embedded clauses.

B   Examples of Parser Errors

We have provided some examples of parser errors which are discussed in section 4.2. The sentence in Figure 5 is usually interpreted as a conditional sentence, however the parser represented it as an embedded alternative interrogative clause. On the other hand, the sentence in Figure 6 is an embedded clause, but the parser represented it as a relative clause.

C   Neg-Raising Generalisation

Another linguistic generalisation that concerns embedded clauses is Neg-Raising. When syntactically negated, Neg-Raising predicates can take semantically higher scope than negation (Horn, 1978; Gajewski, 2007; Zeijlstra, 2018). Consider (7a), which can have the reading as in (7a) but can also have the interpretation in (7b). Accordingly, *believe* is described as a Neg-Raising predicate.

(7) a. I don't believe John is nice

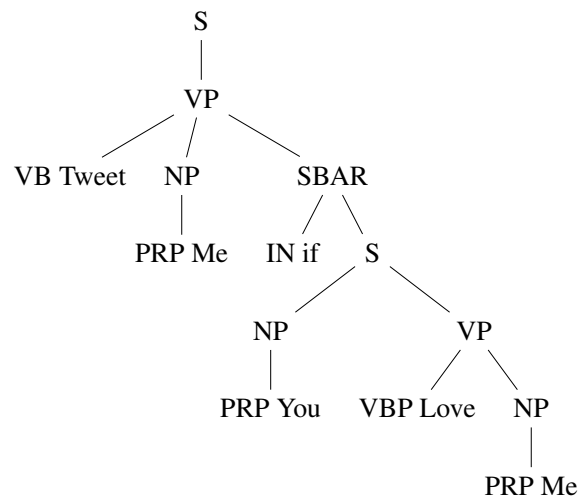   b. I believe that John isn't nice

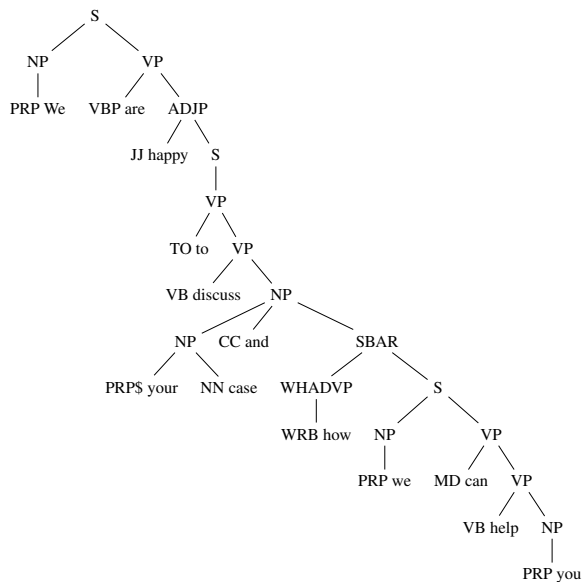Figure 5: False-positive parse of a sentence with a conditional

Figure 6: False-negative parse of a constituent embedded clause

However, many other embedding predicates do not allow this reading. Consider *predict* in (8a), which does not have the reading in (8b), so that *predict* is a non-Neg-Raising predicate

(8)   a.   I don't predict it will rain

     b.   I predict it will not rain

A generalisation of Neg-Raising predicates that has come from the literature is that they do not select for interrogative embedded clauses (Zuber, 1982; Mayr, 2019; Theiler et al., 2019). Thus, we can see in our dataset if there are genuine counter-examples to this generalisation, by searching for the Neg-Raising predicates that have interrogative embedded clauses. We did this with the predicate *believe* which occurred $497,684$ amount of times in the dataset, of which $19,787$ were interrogative embedded clauses. Looking at a subset of these interrogative embedded clauses, the majority of the instances were not true embedded clauses, which is to be expected as we need to propagate the error of the tool into our analysis. However, there were a few potential true counter-examples like (9), which two of the three native speakers amongst the authors found grammatical. However, we leave their analysis for future work.

(9)   a.   As we are not omniscient, we can't validate/believe (absolutely) whether God's existence is true/absolute.

     b.   Ernst was also asked if she believed whether or not the Russia investigation was warranted.