An Argument Structure Construction Treebank

Kristopher Kyle and Hakyung Sung
Learner Corpus Research and Applied Data Science Lab
Linguistics Department, University of Oregon
https://lcr-ads-lab.github.io/LCR-ADS-Home/
{kkyle2, hsung}@uoregon.edu

Abstract

In this paper we introduce a freely available treebank that includes argument structure construction (ASC) annotation. We then use the treebank to train probabilistic annotation models that rely on verb lemmas and/ or syntactic frames. We also use the treebank data to train a highly accurate transformer-based annotation model (F1 = 91.8%). Future directions for the development of the treebank and annotation models are discussed.

1 Introduction

In cognitive linguistics, a construction represents a form-meaning pair. In English, for example, the verb form *laughed* prototypically represents a particular action in the past wherein an entity expresses joy, mirth, or scorn "with a chuckle or explosive vocal sound" (Merriam-Webster, n.d.). Constructions exist at all levels of language (e.g., morphological, lexical, syntactic/argument structure, etc.; Goldberg, 2003). Therefore, while we can analyze *laughed* as a particular form-meaning pair, we can also consider the morphological level, wherein the form *laughed* represents a schematic past-tense construction denoting an event that occurred in the past (laugh$_{verb}$ + *-ed*$_{past}$). Constructions also exist at the syntactic/lexicogrammatical level, wherein a verb and its argument structure constitute a form that corresponds to a propositional meaning (e.g., Diessel, 2004; Fillmore, Kay, & O'Connor, 1988; Goldberg, 1995; 2003; 2006; Jackendoff, 2002). These constructions are referred to as argument structure constructions (ASCs). For example, *they*$_{agent}$ *laughed*$_{verb}$ represents an intransitive ASC, and *they*$_{agent}$ *laughed*$_{verb}$ *him*$_{theme}$ *[out of the room]*$_{goal}$ represents a caused-motion construction. Research has suggested that ASCs are psycholinguisticly real and that both the schematic argument structure (e.g., *agent-verb-theme-goal)* and the verb that fills them (e.g., *laugh*) contribute to sentence meaning (e.g., Bencini & Goldberg, 2000; Gries & Wulff, 2005).

ASCs and Language Learning: Analyzing the relationship between ASC use and productive language development and proficiency has been an increasingly important area of investigation in both first (L1) and second (L2) language learning research (e.g., Clark, 1996; Diessel, 2013; Ellis, 2002; Ellis & Ferreira-Junior, 2009a,b; Hwang & Kim, 2022; Kyle, 2016; Kyle & Crossley, 2017; Kyle et al., 2021; Ninio, 1999; Tomasello & Brooks, 1998). Research suggests that individuals first learn fixed form-meaning pairs that occur frequently in their language experiences. Through more [and varied] language experiences, individuals learn that some pieces of a fixed for-meaning-pair is schematic (e.g., the verb slot). They then tend to overgeneralize the items that can fill a particular slot. Through even more language experiences, they tune their linguistic system to the particular items that tend to occur in a particular slot in a particular construction (see, e.g., Ellis, 2002; Ninio, 1999; Tomasello & Brooks, 1998). For later development (at least in L2 contexts), research has shown that more advanced users tend to use a wider range of ASCs (e.g., Hwang & Kim, 2022) and verb-ASC combinations (e.g., Ellis & Ferreira-Junior, 2009a,b) and (on average) more strongly associated verb-ASC combinations (Kyle, 2016; Kyle & Crossley, 2017).

Extracting ASCs from Corpora: An important issue in studies that analyze the characteristics of ASC use is the method used to identify ASCs and their verbs. Many studies use a manual approach to identify ASCs. While this is appropriate for small-scale studies that measure input directly and/or investigate a limited set of ASCs (e.g., Goldberg et al., 2004; Ellis & Ferreira-Junior, 2009a,b) such an approach puts practical limits the amount of data

Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023), pages 51 - 62
March 9-12, 2023 ©2023 Association for Computational Linguistics

that can be analyzed. Given the increase in availability of large datasets of learner data (e.g., Blanchard et al., 2013; Granger et al., 2009; Ishikawa, 2013) and the increased use of reference corpora as a representation of language experiences (e.g., Römer et al., 2014), automatic methods of ASC extraction have been proposed. These have primarily included either the use of syntactic frames as ASCs (e.g., O'Donnell & Ellis, 2010; Kyle, 2016; Römer et al., 2014) or rule-based systems that rely on syntactic frames and explicit lexical information (Hwang & Kim, 2022). To date, however, no approaches have used machine-learning techniques to predict ASCs directly, primarily because no ASC treebank is currently available.

Contributions of this study: In this study, we build on previous related projects such as PropBank (Palmer et al., 2005), FrameNet (Fillmore et al., 2003), VerbNet (Schuler, 2005) and Universal Propositions (Akbik et al., 2015) to create a publicly available treebank of ASCs. We also leverage machine learning algorithms to create a publicly available automated ASC annotation model.

2 Extracting ASCs from Natural Language Data

ASCs have been extracted from corpora for a range of research purposes. These include (among others), investigating alternation (e.g., dative alternation in English; e.g, Gries & Wulff, 2009; Romain, 2022), verb-construction contingencies (e.g., Ellis & Ferreira-Junior, 2009a,b; Kyle, 2016; Kyle & Crossley, 2017), the validity of using corpus data to represent the mental construction of L1 and L2 users (e.g., Römer et al., 2014), and investigating language proficiency and/or development (e.g., Hwang & Kim, 2022; Kyle & Crossley, 2017; Kyle et al., 2021).

2.1 Manual approaches

The default method of ASC extraction has been manual and/or semi-automated annotation of particular ASC structures. This usually involves pre-selecting candidate verb forms and then determining whether each use of the verb form represents a particular construction. For example, Ellis and Ferreira-Junior (2009a, b) annotated a corpus of L1/L2 interview data (Perdue, 1993) for three construction types (verb-locative, verb-object-locative, and double object constructions) using a list of verbs and a follow up manual analysis. Similar procedures have been used in a number of other studies (e.g., Gries & Wulff, 2009; Romain, 2022) While this approach can achieve high accuracies, the manual nature of searches practically limits how much data can be examined. Furthermore, if the goal is to comprehensively examine the relationship between verbs and ASCs (which is the case in some studies), all verbs (and their constructions) in a corpus must be examined.

2.2 Syntactic frame as construction approach

As the availability of large corpora of language use increased and the use of dependency representations gained traction in the field of natural language processing, some scholars began to use dependency-based syntactic frames to identify constructions (e.g., O'Donnell & Ellis, 2010; Kyle, 2016; Kyle & Crossley, 2017). For example, the syntactic frame *subject-verb-object$_{indirect}$-object$_{direct}$* can be used to reliably identify ditransitive constructions. O'Donnell & Ellis (2010) used a dependency-parsed version of the BNC (Andersen et al., 2008) to preliminarily extract constructions for the purposes of examining verb-construction contingencies. Ellis and colleagues used a related approach to explore the relationship between corpus contingencies and online choices in verb-preposition-object constructions (e.g., Römer et al., 2014). However, the relatively low accuracy of the RASP parser (F1 = .763 averaged annotation accuracy) limited the types and specificity of the constructions that could be reliably examined.

As dependency parsers increased in accuracy (and speed) with the introduction of neural-net models (e.g., F1 = .896; Chen & Manning, 2014) and transformer models (e.g., F1 = .951; Honnibal et al., 2020) some researchers (e.g., Kyle, 2016; Kyle & Crossley, 2017; Kyle et al., 2021) explored the contingency of dependency-based syntactic frames and verbs in large corpora such as the Corpus of Contemporary American English (Davies, 2009). These contingencies were then successfully used to model differences in language use across L2 proficiency levels.

While the syntactic frame approach has been useful in a number of contexts, syntactic frames do not directly represent ASCs in all cases. Multiple dependency-based syntactic frames can map onto a single ASC and conversely a single syntactic frame may represent multiple ASCs depending on the context. For example, *subject-verb-object-oblique*$_{prep\_on}$ can represent both a simple transitive construction (*I*$_{subject}$ *found*$_{verb}$ *this*$_{object}$ *[on a bulletin board]*$_{oblique}$) or a caused-motion construction (*I*$_{subject}$ *put*$_{verb}$ *it*$_{object}$ *[on my hand]*$_{oblique}$).

2.3 Rule-based approach

Another approach uses a set of rules written over a dependency representation to identify particular ASCs. For example, Hwang & Kim (2022) identified 11 ASC types (e.g., caused-motion, ditransitive) using a manually derived rule-based system that relies on dependency-based syntactic frames and some lexical items. Although they do not report accuracy on a by-ASC basis, they report an overall F1 score of .82. While this approach represents an interesting preliminary step in identifying particular ASCs, it is not clear how well it can generalize to unseen structures and/or lexical items.

2.4 Other potential approaches

When we convey meaning via a particular form of ASC, a verb interacts with the arguments in the construction. Semantically, the arguments in the construction relate to abstract meanings such as *agent, patient, theme, goal, result,* etc. (Fillmore, 1968; Palmer, Gildea, & Xue, 2010), which are called semantic roles. Semantic roles help encode the general senses that are basic to human experience (Scene Encoding Hypothesis, Goldberg, 1995; Kay & Fillmore, 1999), which in turn are useful for classifying ASCs.

As previously noted, a limitation of the syntactic frame approach is that functional grammatical labels (e.g., subject, direct object, oblique) are not fine-grained enough to determine the semantic role of an argument. Although some preliminary work has been done in the area of automatic semantic role labeling (e.g., Gardner et al., 2018; Shi & Jin, 2019), current state of the art models are not accurate enough to make this a feasible option (though this may change in the future). However, treebanks with manually-annotated semantic role labels present a helpful starting point for a treebank of ASCs.

2.5 Machine-learning approaches

In order for machine-learning models to be used to create automatic ASC annotation models, treebanks that include ASC information are needed. Although some previous work has been done on specific ASC types, such as caused-motion constructions (Hwang, 2014; Hwang et al., 2010), to our knowledge there are currently no publicly available treebanks that are annotated for ASCs. Additionally, although some previous work has trained models to identify a specific ASC type (e.g., Hwang et al., 2010, 2015), there have been no machine-learning based models that annotate a wider range of ASCs. In this study we address these gaps by introducing a publicly available treebank annotated for ASCs. We then introduce a series of automatic ASC annotation models, including a highly accurate transformer-based model.

3 Method

3.1 Creating an ASC treebank

For this project, we used the English portion of the Universal Propositions project (Akbik et al., 2015), which represents a merge of the Universal Dependencies version of the English Web Treebank (EWT; Bies et al., 2012; Silveira et al., 2014) and PropBank (Palmer et al., 2005). The EWT corpus includes sentences sampled from five web registers, including blogs, newsgroups, emails, reviews, and Yahoo! Answers.

We used a semiautomatic approach to annotating the ASC treebank. For each sentence in the training section of the EWT, we first extracted the large-grained argument structures using the default PropBank semantic role labels (e.g., *ARG0-Verbsense-ARG1*). We then converted the large-grained arguments to fine-grained semantic role frames (e.g., *agent-Verb-theme*) using relation mappings from the PropBank frame files (Palmer et al., 2005), which also draw on information in FrameNet (Fillmore et al., 2003) and VerbNet (Schuler, 2005). After a discussion of ASC categorization between the authors that included co-annotation of 100 sentences, the second author (a PhD student with a specialization in construction grammar) manually assigned an ASC to each semantic role frame that occurred at least 5 times in the corpus (*n* = 355) based on the semantics of the frame and its typical use in the treebank sentences. For example, the semantic role frame *theme-Verb-attribute* was annotated as an attributive construction and *agent-Verb-theme* was annotated as a transitive simple construction. In some cases, the corpus analysis indicated that particular semantic role frames could represent multiple

ASC	**Most frequent verbs**	**Total Freq**	**Train**	**Dev**	**Test**
TRAN_S	*have*, *do*, *say*	12,431	9,965	1,213	1,253
ATTR	*be*, *seem*, *look*	6,004	4,723	648	633
INTRAN_S	*go*, *work*, *come*	2,754	2,200	289	265
PASSIVE	*attach*, *do*, *call*	1,818	1,481	167	170
INTRAN_MOT	*go*, *come*, *get*	1,098	915	88	95
TRAN_RES	*let*, *make*, *get*	977	795	90	92
CAUS_MOT	*take*, *put*, *send*	675	546	64	65
DITRAN	*give*, *tell*, *ask*	534	448	40	46
INTRAN_RES	*become*, *go*, *come*	146	121	9	16
Total		**26,437**	**21,194**	**2,608**	**2,635**

Table 1: ASC Representation in Treebank

ASCs. This most often occurred in cases where a fine-grained semantic role for a particular argument of a particular verb was unavailable in PropBank, leading to an underspecified semantic role frame. In these cases, the use of each semantic role frame + verb combination that occurred at least twice in the treebank was checked and each was assigned an ASC. Particularly ambiguous cases were resolved through discussions with the first author. As a final step, we conducted spot checks which led to a small number of corrections. This approach resulted in the categorization of 94.1% of the ASCs in the treebank. Any sentences that included uncategorized ASCs were omitted from further analysis.

In order to evaluate the quality of the semi-automated annotation process, the Authors independently annotated a random sample of 100 sentences from the ASC treebank. The 100 sentences included 189 ASC tokens. The results demonstrated substantial agreement between annotators (*kappa* = .773; *simple agreement rate* = 84.1%; Landis & Koch, 1977). The Authors then adjudicated the annotations until perfect agreement was reached. The annotations generated by the semi-automated process demonstrated excellent agreement with the adjudicated scores (*kappa* = .884, *simple agreement rate* = 92.1%).

In total, 26,437 ASC instances were annotated and included in the analysis (see Table 1 for a summary of the distribution of ASCs in each section of the treebank). The ASC Treebank is freely available at https://github.com/LCR-ADS-Lab/ASC-Treebank and https://osf.io/ncjx8/?view_only=163c81a90eec44fb9ee317ff6fa4d4a6).

3.2 ASCs represented

Though there are many commonalities across ASC types that are investigated, there is currently no definitive set of ASCs that should be included in an ASC tag set, and there are varying levels of specificity that could be represented (e.g., Hwang et al., 2010; 2015). The current study drew on a range of previous literature (e.g., Biber et al., 1999; Goldberg, 1995, 2006; Hwang & Kim, 2022). The nine ASC types included in this study represent an attempt to balance specificity and semantic generalization. Note that all examples in the following subsections come from the training section of the treebank.

3.2.1 Attributive construction

The attributive (ATTR) ASC includes two arguments, namely a *theme* and an *attribute*. The *attribute* is prototypically represented by a noun (e.g., *[it]$_{theme}$ was [an evolution]$_{attribute}$*), an adjective (*[I]$_{theme}$ am [sure]$_{attribute}$*), or a prepositional phrase (*[your dog]$_{theme}$ ... is [in the same room]$_{attribute}$*; Biber et al., 1999). Most commonly, the copular verb *be* is used in this construction.

3.2.2 Intransitive constructions

Intransitive constructions typically include a single argument but can include two arguments if the construction denotes more than a simple action, such as a movement or a state change of a subject argument. We subcategorize intransitive constructions into simple, motion, and resultative ASCs.

Intransitive simple: The intransitive simple (INTRAN_S) ASC includes a single argument and typically denotes either what an *agent* does (e.g., *[I]*$_{agent}$ *am working from our Hong Kong office*) or what happens to a *theme* (e.g., *[Martin's box]*$_{theme}$ *is working wonderfully*)".

Intransitive motion: The intransitive motion (INTRAN_MOT) ASC involves two arguments including a *mover*/*theme* and a *path* (Goldberg, 1995). The path is typically denoted via an adverbial particle (e.g., *[The morbidity rate]*$_{theme}$ *is going [up]*$_{ARGM\text{-}DIR}$) or a prepositional phrase (e.g., *[I]*$_{theme}$ *went [across the bay]*$_{goal}$).

Intransitive resultative: The intransitive resultative (INTRAN_RES) ASC involves two arguments, including a *patient* and a *result*. The construction denotes a patient changing state (e.g., *[The spine]*$_{patient}$ *will become [flexible]*$_{result}$).

3.2.3 Simple transitive construction

The simple transitive construction (TRAN_S) includes two arguments that describe an action done by a subject argument to an object argument. The simple transitive ASC prototypically includes an *agent* and a *theme*/*patient*. The *theme*/*patient* generally represents an entity that is affected by the action denoted by the verb (Biber et al., 1999; e.g., *[They]*$_{agent}$ *are targeting [ambulances]*$_{theme}$). The simple transitive can also denote mental activities (e.g., *[I]*$_{agent}$ *thought [the US government was looking for me]*$_{theme}$) and states (e.g., *[I]*$_{experiencer}$ *love [my gym]*$_{stimulus}$). The simple transitive is also inclusive of communication activities such as speaking or writing (e.g., *[He]*$_{agent}$ *claimed [that they have the means to stage]*$_{topic}$).

3.2.4 Ditransitive Construction

The ditransitive construction (DITRAN) prototypically includes three arguments (e.g., *agent*, *recipient,* and *theme*). It evokes the notion of literal or metaphorical transfer (e.g., *[You]*$_{agent}$ *feed [your rabbits]*$_{recipient}$ *[non-veg items]*$_{theme}$). The ditransitive construction is inclusive of the transfer of a topic during communication (e.g., *[I]*$_{agent}$ *told [the little girl]*$_{recipient}$ *[that she would have to accompany me to school]*$_{topic}$).

3.2.5 Complex Transitive Constructions

Complex transitive constructions include three arguments that describe either a movement or a change in state of an object argument caused by an action of a subject argument. We subcategorize these into caused-motion and transitive resultative constructions as outlined below.

Caused-motion: The caused-motion (CAUS_MOT) ASC includes an *agent* that causes a *theme* to move along a path designated by a directional phrase (Goldberg, 1999). Semantically, caused-motion ASCs are inclusive of both direct causation (e.g., *[I]*$_{agent}$ *took [it]*$_{theme}$ *[there]*$_{destination}$) and indirect causation (e.g., *[The body]*$_{agent}$ *brings [stability]*$_{theme}$ *[to the region]*$_{goal}$).

Transitive resultative: The transitive resultative (TRAN_RES) prototypically includes an *agent*, a *patient*/*theme* and a *result* wherein the *agent* causes the *patient*/*theme* to become the *result* (e.g., ... *[the vessel]*$_{agent}$ *changed [its name]*$_{patient}$ *at sea to [Horizon]*$_{result}$). We also include verb-particle constructions wherein the paired particle has a figurative meaning of the resultative state (e.g., *[No preacher]*$_{agent}$ *has ever blown [himself]*$_{theme}$ *[up]*$_{C\text{-}V}$).

3.2.6 Passive Constructions

Passive (PASSIVE) contains short passive (a form without an expressed agent in *by*-phrase; e.g., *[You]*$_{theme}$ *are invited*$_{\_Vpassive}$ *to join with members of the forum*) and long passive (with an expressed agent; e.g., *coined*$_{\_Vpassive}$ *[by Bill Gates]*$_{agent}$ *to represent the company* (Biber et al., 1999). This also includes past participle pre-modifiers (e.g., *overlooked*$_{\_\text{Vpassive}}$ *[problem]*$_{theme}$) and post-modifiers (e.g., *She guided me through a very difficult period dealing with a family member's suicide, coupled*$_{\_Vpassive}$ *with elder abuse*).

3.2.7 Annotation scheme summary

In total, the corpus is annotated for nine ASC types. Multiple, overlapping ASCs may be present in a particular utterance. For example, a clausal argument of an ASC will represent an additional ASC as in [*But the best way is [to use coupons*]$_{TRAN\_S}$]$_{ATTR}$.

3.2.8 Model Training and Evaluation

We trained three probabilistic models and a transformer model based on RoBERTa (Liu et al., 2019) embeddings. The probabilistic models served two purposes. The first purpose was theoretical in nature (e.g., how well can we predict an ASC based on its verb versus its syntactic frame) and the second was as a set of linguistically-informed baseline models. A transformer model was also used because these models are particularly well suited for the task of ASC identification given that they use a high-featured vector space representation of the context to predict the category of a section of text. The probabilistic models presumed that main verb heads of argument structure constructions were pre-identified (which is relatively trivial using a part of speech tagger and a dependency parser), while the transformer model evaluated all tokens and identified whether a token was the head of an ASC, and which ASC was represented by the token. As such, the annotation task for the probabilistic models was less demanding than the annotation task for the transformer model.

Model 1 (Verb lemmas): The first model calculated the probability that a particular verb lemma token would occur in a particular ASC. While it is likely that better results would be achieved using verb senses instead of verb lemmas, automated verb sense disambiguation is not currently sufficiently accurate to make this approach generalizable for data outside of PropBank. Each main verb lemma that represented the head of an ASC was annotated as the most probable ASC for that verb. For example, in the training data, the verb lemma *put* was most likely to occur in the CAUS_MOT construction, though it also occurred in the TRAN_S and TRAN_RES constructions. Any verb in the development or test set that was not represented in the training data was assigned the most frequent ASC in the training data (TRAN_S).

Model 2 (Syntactic frames): The second model calculated the probability that a particular syntactic frame token would represent an ASC. Drawing on previous research (e.g., Kyle, 2016; Kyle & Crossley, 2017; Kyle et al., 2021; O'Donnell & Ellis, 2010), syntactic frames were operationalized based on the functional grammatical labels included in the dependency representation. In this case, dependency representations followed Universal Dependencies (UD; Nivre et al., 2020). Copular constructions were adapted slightly to allow the copular verb to represent the head of copular constructions. Following previous research (e.g., Kyle & Crossley, 2017; Römer et al., 2014), concrete realizations of prepositions were included in the syntactic frames, and auxiliary verbs were excluded. For example, the syntactic frame *subject_verb_object_on-oblique*, most commonly represented the TRAN_S ASC (e.g., … *[you]*$_{nsubj}$ *have [a bunch of stuff]*$_{object}$ *[on your plate]*$_{obl}$), though it also represented the CAUS_MOT ASC (e.g., *[It]*$_{nsubj}$ *put [hair]*$_{obj}$ *[on my chest]*$_{obl}$). Any syntactic frames in the development or test set that were not represented in the training data were assigned the most frequent ASC in the training data (TRAN_S).

Model 3 (Verb lemma + Syntactic frames): The third model calculated the probability that a particular verb lemma + syntactic frame combination token would represent a particular ASC. As a concrete example, while the verb *put* occurs in multiple ASCs, and the syntactic frame *subject_verb_object_on-oblique* represents at least two ASCs, in the training data the combination of *put* + *subject_verb_object_on-oblique* represented a single ASC (CAUS_MOT). This model used three back-offs. If the verb lemma + syntactic frame was not represented in the training data, the syntactic frame probabilities were used, followed by the verb lemma probabilities and, as a last resort, the most common tag in the training data (TRAN_S).

ASC	**Freq**	**lemma model**	**syntactic frame model**	**lemma + syntactic frame model**	**transformer model**
TRAN_S	1,253	0.821	0.824	0.897	**0.938**
ATTR	633	**0.982**	0.884	0.972	**0.982**
INTRAN_S	265	0.373	0.617	0.713	**0.859**
PASSIVE	170	0.283	0.799	0.809	**0.862**
INTRAN_MOT	95	0.522	0.258	0.540	**0.769**
TRAN_RES	92	0.397	0.723	0.756	**0.798**
CAUS_MOT	65	0.301	0.524	0.557	**0.742**
DITRAN	46	0.536	0.747	0.825	**0.905**
INTRAN_RES	16	0.519	0.105	0.640	**0.759**
Weighted Average		0.735	0.779	0.862	**0.918**

Table 2: F1 scores for each model (test set)

ASC	P	R	F1
TRAN_S	0.927	0.949	0.938
ATTR	0.989	0.975	0.982
INTRAN_S	0.884	0.837	0.859
PASSIVE	0.878	0.847	0.862
INTRAN_MOT	0.750	0.789	0.769
TRAN_RES	0.802	0.793	0.798
CAUS_MOT	0.731	0.754	0.742
DITRAN	0.878	0.935	0.905
INTRAN_RES	0.846	0.688	0.759
Weighted Average	0.917	0.920	0.918

Table 3: Transformer model results in terms of precision, recall, and F1

Model 4 (Transformer model): The fourth model used RoBERTa embeddings to predict whether a word represented the head of a particular ASC. Unlike Models 1-3, which classified an ASC based on a pre-identified main verb, syntactic frame, or verb + syntactic frame combination, Model 4 evaluated each word in a sentence and determined a) whether the word represented the head of an ASC (i.e., was a main verb) and if so, b) the ASC represented by that verb in the sentence. Models were trained using the transformer-based single-class named entities model in Spacy (version 3.4; Honnibal et al., 2020). Models were developed using the training set data, fine-tuned using the development set data, and finally evaluated on the test set data.

4 Results

The results indicated that all models performed well above the simple baseline accuracy (F1 = .307 when all ASCs are tagged as TRAN_S). The transformer model achieved the highest overall classification accuracy (F1 = .918), followed by the verb lemma + syntactic frame model, the syntactic frame model, and the verb lemma model. With regard to individual ASC types, the transformer model also achieved the highest F1 score for each of the 9 ASCs represented in the treebank (inclusive of a tie with the lemma model for the ATTR ASC). The results for the four models (F1 scores) are summarized in Table 2. The full results (precision, recall, and F1 for each ASC type) for the transformer model are included in Table 3.

5 Discussion

In this study, we introduce a treebank with ASC annotations and an automated ASC annotation model. Below, we discuss features of and future directions for the corpus and the prediction models. We also discuss future directions with regard to applied research using the model.

5.1 ASC Treebank

To our knowledge, the ASC Treebank represents the first publicly available and open-source treebank annotated for ASC types. In total, the ASC treebank currently includes 30,664 annotated ASCs across 9 ASC types. When sentences that include uncategorized ASCs are excluded, 26,437 ASCs annotations are represented.

5.1.1 ASC representation

Although some ASCs are well-represented in the treebank (e.g., TRAN_S, ATTR, and INTRAN_S), others are underrepresented (e.g., CAUS_MOT, DITRAN, INTRAN_RES, and TRAN_RES). Instances of the INTRAN_RES ASC, for example, comprises only 0.5% of ASCs instances in the treebank. While this may be representative of the registers included in the EWT (i.e., blogs, newsgroups, emails, reviews, and Yahoo! Answers) the distribution may not be representative of other registers. Regardless, very low representation of INTRAN_RES likely contributed to lower annotation accuracy for this ASC. Future treebank development should include a focus on including more instances of underrepresented ASC types.

5.1.2 Register representation

It is well known that natural language processing models work better on in-domain texts (i.e., texts that share register features) than on out of domain texts (e.g., McClosky et al., 2006). Although the EWT treebank was a convenient context in which to build an ASC treebank, some researchers will be interested in extracting and analyzing texts from registers other than those represented by the EWT. Future treebank development should therefore include a focus on increasing register coverage. Ideally, this would involve adding manual annotations to other publicly available corpora, such as written and spoken L2 corpora that are annotated for universal dependencies (e.g., Berzak et al., 2016; Kyle et al., 2022).

5.1.3 Improved annotation and treebank coverage

The inclusion of verb senses and semantic role labels from Propbank, FrameNet, and VerbNet allowed for the efficient annotation of a relatively large number of ASCs. In total 30,664 (94.1%) of the ASCs in the treebank could be identified using a relatively small set (n = 355) of semantic frame to ASC mappings (plus some verb + semantic frame specific mappings). However, 5.9% of the ASCs in the treebank remain uncategorized. Future treebank development should include a focus on manually annotating the remaining uncategorized ASCs.

One limitation to the approach of using semantic frame (and verb + semantic frame) to ASC mappings is that some semantic role frames in ProbBank (even when augmented with information from VerbNet and FrameNet) may correspond to multiple ASCs. In the EWT data, this was relatively common when one or more elements in semantic frames were underspecified (e.g., *agent-Verb-**ARG2***). In many cases, ambiguous cases could be addressed by looking at how each semantic frame was used in context with a particular verb. However, in some cases, even seemingly unambiguous semantic frames and/or verb sense + semantic frame combinations could be mapped to multiple ASCs. For example, the verb sense *go.08* when used in the semantic frames *(experiencer-)Verb-result* prototypically represents the INTRAN_RES ASC (e.g., *the company went bankrupt*). However, in the EWT, this combination also includes a very few instances that are not representative of the INTRAN_RES ASC, such as *go on your computer*. The small-scale accuracy analysis (100-sentences; 189 ASCs) suggested that agreement was high between the ASC annotations produced by the semi-automated process used in this study and the adjudicated gold-standard ACS annotations (*kappa* = .884; *simple agreement rate* = 92.1%). Although this agreement was higher than between two expert annotators, there is certainly room for improving the quality of the ASC annotations in the treebank. Future treebank development should therefore include a focus on providing additional quality checks and edits in the treebank.

5.2 Prediction models

In this study, three probabilistic models focused on verbs and/or syntactic frames and one transformer model was trained and tested. All models performed well above baseline accuracy. Below we provide a summary of the strengths and weakness of each model, followed by a concrete example of the performance of the most accurate model (transformer model).

5.2.1 Verb lemma model

The verb lemma model (precision = 0.742, recall = 0.758, F1 = 0.735) performed better than baseline, but less well than the other models. Unsurprisingly, the verb lemma model performed well when identifying ATTR (precision = 0.987, recall = .973, F1 = .982), given that the copular verb *be* is very strongly associated with ATTR. The verb model also performed reasonably well when identifying the TRAN_S ASC (precision = 0.755, recall = .900, F1 = .821), but did not perform well (F1 < .600) when identifying other ASCs. These results provide some support for the notion that verbs are not the only (and not necessarily the primary) determinant of the meaning of a sentence/clause (e.g., Bencini & Goldberg, 2000).

5.2.2 Syntactic frame model

The syntactic frame model (precision = 0.793, recall = 0.784, F1 = 0.779) performed better than the verb lemma model, but less well than the remaining two models. The syntactic frame model performed reasonably well (F1 > .700) when annotating 5 of the 9 ASCs (e.g., ATTR, TRAN_S, DITRAN) but performed less well with other four, and in particular those with ambiguous dependency structures (e.g., INTRAN_RES and CAUSE_MOT). These results suggest that although syntactic frames derived from dependency representations are helpful in the identification of some ASCs, dependency syntactic frames should likely not be equated with ASCs.

5.2.3 Verb lemma + syntactic frame model

Unsurprisingly, the verb lemma + syntactic frame model performed much better (precision = 0.866, recall = 0.863, F1 = 0.862) than the models that relied on verb lemmas or syntactic frames only. The model performed reasonably well (F1 > .700) when annotating 6 of the 9 ASCs, but performed less well when annotating CAUS_MOT, INTRAN_MOT, and INTRAN_RES. These structures were particularly difficult to annotate accurately because ambiguity can only be resolved by determining (in the case of CAUS_MOT and INTRAN_RES) whether a predicate phrase such as a prepositional phrase represents a *goal*/*path*/*source* or has a different function. While ambiguities can sometimes be resolved by the preposition used, this is not always the case (leading to low annotation accuracies). This provides further support for the distinction between syntactic frames and ASCs and the need for treebanks annotated for features beyond syntactic dependency representations.

5.2.4 Transformer model

The best performing model was the transformer model (precision = 0.917, recall = 0.920, F1 = 0.918). Unlike the probabilistic models, all ASCs were annotated with an F1 > 0.740. Three ASCs (TRAN_S, ATTR, and DITRAN) were annotated with an F1 > .900. Two more ASCs (INTRAN_S and PASSIVE) were annotated with an F1 > 0.850. These results suggest that transformer models, which rely on a highly-featured vector space representation of a word's context, are particularly well-suited for the automated annotation of ASCs. While these results represent a high degree of accuracy in automated ASC identification, there are still important improvements to be made with regard to the annotation of structures that are less well represented in the ASC treebank (e.g., INTRAN_RES and CAUS_MOT). Future research should focus on improving annotation of these features through model optimization techniques such as oversampling and the addition of sentences to the treebank that include underrepresented ASCs.

5.2.5 Concrete example

To demonstrate the performance of the transformer model in concrete terms, we used the transformer model to identify ASCs in the 16 sentences used in Bencini & Goldberg (2000). In the study, four verbs (*get*, *slice*, *throw, and took*) were each used in four ASCs (TRAN_S, DITRAN, CAUS_MOT, and TRAN_RES). The transformer model from this current study accurately classified all instances of the TRAN_S ASC (*Anita threw the hammer.*, *Michelle got the book*, *Barbara sliced the bread*, and *Audrey took the watch*), the DITRAN ASC (*Chris threw Linda the pencil*, *Beth got Liz an invitation*, *Jennifer sliced Terry an apple*, and *Paula took Sue a message*), and the CAUS_MOT ASC (*Pat threw the keys on the roof*, *Laura got the ball into the net*, *Meg sliced the ham onto the plate*, and *Kim took the rose into the house*). However, the model struggled to classify the TRANS_RES ASCs, and only classified two of the four correctly (*Dana got the mattress inflated* and *Nancy sliced the tire open*). The other two TRAN_RES instances (*Lyn threw the box apart* and *Rachel took the wall down*) were classified as CAUS_MOT, suggesting that more (and more diverse) instances of the TRAN_RES ASC are needed in future iterations of the treebank.

5.3 Applications for future research in linguistics

Previous corpus-based studies of language development and/or proficiency have typically either used manual/semi-automatic approaches to the identification of ASCs (e.g., Ellis & Ferreira-Junior, 2009a; Goldberg et al., 2004). Such approaches are resource intensive and, in most cases, lead to the analysis of a relatively small dataset and/or a limited number of ASCs. Some researchers have leveraged advances in dependency annotation to identify ASCs in larger corpora of both highly proficient language users and language learners using verb + syntactic frame combinations (e.g., Hwang & Kim, 2022; Kyle, 2016; Kyle & Crossley, 2017). The results of this study suggest that while verb + syntactic frames can be used to identify ASCs with a reasonable degree of accuracy (F1 = .862), the transformer-based annotation model introduced in this study is both more accurate overall (F1 = .918) and more stable across ASC types. Future research should investigate the application of the model introduced in this study to corpus-based studies of language learning and in areas such as automatic essay scoring and feedback. This research should include the replication of previous studies that have used less accurate methods of identifying ASCs (e.g., Hwang & Kim, 2022; Kyle & Crossley, 2017).

6 Conclusion

In this study, we introduce publicly available and open-source treebank annotated with ASCs. We also present a highly accurate ASC annotation model, which performs much better (F1 = 0.918) than previously reported rule-based systems (F1 = 0.820; Hwang & Kim, 2021). While improvements can be made with regard to the size and representativeness of the treebank, the results of this study suggest that future treebank annotation efforts would be beneficial to researchers interested in examining ASC use at scale.

References

Akbik, A., Chiticariu, L., Danilevsky, M., Li, Y., Vaithyanathan, S., & Zhu, H. (2015, July). Generating high quality proposition banks for multilingual semantic role labeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1)*, 397-407.

Andersen, Ø. E., Nioche, J., Briscoe, T., & Carroll, J. A. (2008). The BNC parsed with RASP4UIMA. *Proceedings of the Sixth International Language Resources and Evaluation (LREC08)*, 28-30.

Bencini, G. M., & Goldberg, A. E. (2000). The contribution of argument structure constructions to sentence meaning. *Journal of Memory and Language*, *43*(4), 640-651.

Berzak, Y., Kenney, J., Spadine, C., Wang, J. X., Lam, L., Mori, K. S., Garza, S., & Katz, B. (2016). Universal dependencies for learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 737-746.

Biber, D., Johansson, S., Leech, G., Conrad, S., Finegan, E., & Quirk, R. (1999). *Longman grammar of spoken and written English* (Vol. 2). London: Longman.

Bies, A., Mott, J., Warner, C., & Kulick, S. (2012). English web treebank. *Linguistic Data Consortium, Philadelphia, PA.*

Blanchard, D., Tetreault, J., Higgins, D., Cahill, A., & Chodorow, M. (2013). TOEFL11: A corpus of non-native English. *ETS Research Report Series*, *2013*(2), i-15.

Chen, D., & Manning, C. D. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 740-750.

Clark, H. H. (1996). *Using language*. Cambridge university press.

Ninio, A. (1999). Pathbreaking verbs in syntactic development and the question of prototypical transitivity. *Journal of child language*, *26*(3), 619-653.

Davies, M. (2009). The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics, 14*(2), 159–190.

Diessel, H. (2004). *The acquisition of complex sentences*. Cambridge University Press.

Diessel, H. (2013). Construction grammar and first language acquisition. *The Oxford handbook of construction grammar*, *347*, 364.

Ellis, N. C. (2002). Frequency effects in language processing: A review with implications for theories of implicit and explicit language acquisition. *Studies in second language acquisition*, *24*(2), 143-188.

Ellis, N. C., & Ferreira-Junior, F. (2009a). Constructions and their acquisition: Islands and the distinctiveness of their occupancy. *Annual Review of Cognitive Linguistics*, *7*(1), 188-221.

Ellis, N. C., & Ferreira–Junior, F. (2009b). Construction learning as a function of frequency, frequency distribution, and function. *The Modern language journal*, *93*(3), 370-385.

Fillmore, C. J. (1968). The case for case. In E. Bach & R. T. Harms. (Eds.), *Universals in linguistic theory*, 1-88.

Fillmore, C. J., Johnson, C. R., & Petruck, M. R. (2003). Background to Framenet. *International journal of lexicography*, *16*(3), 235-250.

Fillmore, C. J., Kay, P., & O'connor, M. C. (1988). Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, *64*(3), 501-538.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, Mi., & Zettlemoyer, L. (2018). Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 1-6.

Goldberg, A. E. (1995). *Constructions: A construction grammar approach to argument structure*. University of Chicago Press.

Goldberg, A. E. (2003). Constructions: A new theoretical approach to language. *Trends in cognitive sciences*, 7(5), 219-224.

Goldberg, A. E. (2006). *Constructions at work: The nature of generalization in language*. Oxford University Press on Demand.

Goldberg, A. E., Casenhiser, D. M., & Sethuraman, N. (2004). Learning argument structure generalizations. *Cognitive Linguistics*, *15*(3), 289-316.

Granger, S., Dagneaux, E., Meunier, F., & Paquot, M. (Eds.). (2009). *International corpus of learner English* (Vol. 2). Louvain-la-Neuve: Presses universitaires de Louvain.

Gries, S. T., & Wulff, S. (2005). Do foreign language learners also have constructions?. *Annual Review of Cognitive Linguistics*, *3*(1), 182-200.

Gries, S. T., & Wulff, S. (2009). Psycholinguistic and corpus-linguistic evidence for L2 constructions. *Annual Review of Cognitive Linguistics*, *7*(1), 163-186.

Hwang, J. D. (2014). Identification and representation of caused motion constructions (Doctoral dissertation). University of Colorado at Boulder.

Hwang, H., & Kim, H. (2022). Automatic Analysis of Constructional Diversity as a Predictor of EFL Students' Writing Proficiency. *Applied Linguistics*.

Hwang, J. D., Nielsen, R., & Palmer, M. (2010). Towards a domain independent semantics: Enhancing semantic representation with construction grammar. In *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*, 1-8.

Hwang, J. D., & Palmer, M. (2015). Identification of caused motion construction. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, 51-60.

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.

Ishikawa, S. I. (2013). The ICNALE and sophisticated contrastive interlanguage analysis of Asian learners of English. *Learner corpus studies in Asia and the world*, *1*(1), 91-118.

Jackendoff, R. (2002). *Foundations of language*. Oxford University Press.

Kay, P., & Fillmore, C. J. (1999). Grammatical constructions and linguistic generalizations: The What's X doing Y? construction. *Language*, *75*(1), 1–33.

Kyle, K. (2016). Measuring syntactic development in L2 writing: Fine grained indices of syntactic complexity and usage-based indices of syntactic sophistication. (Doctoral dissertation). Georgia State University.

Kyle, K., & Crossley, S. (2017). Assessing syntactic sophistication in L2 writing: A usage-based approach. *Language Testing*, *34*(4), 513-535.

Kyle, K., Crossley, S., & Verspoor, M. (2021). Measuring longitudinal writing development using indices of syntactic complexity and sophistication. *Studies in Second Language Acquisition*, *43*(4), 781–812.

Kyle, K., Eguchi, M., Miller, A., & Sither, T. (2022). A Dependency Treebank of Spoken Second Language English. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications*, 39-45.

Landis, J. R., & Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, 363-374.

Merriam-Webster. (n.d.). Laugh. In *Merriam-Webste.com dictionary*. Retrieved November 10, 2022.

McClosky, D., Charniak, E., & Johnson, M. (2006). Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 337-344.

Nivre, J., Marneffe, M.-C. de, Ginter, F., Hajic, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F., & Zeman, D. (2020). Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 4034–4043.

Ninio, A. (1999). Pathbreaking verbs in syntactic development and the question of prototypical transitivity. *Journal of child language*, *26*(3), 619-653.

O'Donnell, M. B., & Ellis, N. C. (2010). Towards an inventory of English verb argument constructions. In *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*, 9–16.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, *31*(1), 71-106.

Palmer, M., Gildea, D., & Xue, N. (2010). Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, *3*(1), 1-103.

Perdue, C. (Ed.). (1993). *Adult language acquisition: Crosslinguistic perspectives*. Cambridge: Cambridge University Press.

Romain, L. (2022). Putting the argument back into argument structure constructions. *Cognitive Linguistics*, *33*(1), 35-64.

Römer, U., Roberson, A., O'Donnell, M. B., & Ellis, N. C. (2014). Linking learner corpus and experimental data in studying second language learners' knowledge of verb-argument constructions. *ICAME Journal*, *38*(1), 115–135.

Schuler, K. K. (2005). *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Shi, P., & Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *arXiv:1904.05255*.

Silveira, N., Dozat, T., De Marneffe, M.-C., Bowman, S., Connor, M., Bauer, J., & Manning, C. (2014). A Gold Standard Dependency Corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2897–2904.

Tomasello, M., & Brooks, P. J. (1998). Young children's earliest transitive and intransitive constructions. *Cognitive Linguistics*, *9*(4), 379–395.