Dependency Annotation for Learner Corpora

Markus Dickinson

Indiana University
Bloomington, IN 47405
E-mail: md7@indiana.edu

Marwa Ragheb

Indiana University
Bloomington, IN 47405
E-mail: mragheb@indiana.edu

Abstract

Building from the CHILDES dependency annotation scheme and on interlanguage POS annotation, we describe a syntactic annotation scheme developed for the data of second language learners. We encode subcategorization frames and underlying dependencies, in addition to the usual surface dependencies. The annotation scheme is relatively independent of language and can be mapped to learner errors.

1 Introduction and Motivation

A prominent area of research in the collection of corpora containing the language of second language learners has been in annotating so-called errors (see, e.g., Granger, 2003). This helps further studies in computer-assisted error analysis and develop technology for detecting learner errors, but does not address more general properties of learner language, such as fluency and complexity (e.g., Pendar and Chapelle, 2008), stage of acquisition (e.g., Pienemman, 1998), or even basic features such as tense and aspect (e.g., Wulff et al., 2009). There is thus a need for annotated data which more generally supports second language acquisition (SLA) research.

What needs to be described is *interlanguage*, the in-progress language of learners which is a linguistic system in its own right, without focusing on errors. Díaz-Negrillo et al. (2009) annotate interlanguage part-of-speech (POS) properties, but, as far as we know, no one has syntactically annotated interlanguage. This is despite the fact that there is much work on automatically detecting learner syntactic errors (see, e.g., Vandeventer Faltin, 2003).

At the same time, any such annotation effort would ideally also be useful to the computational linguistics community. To accurately parse learner data, it is desirable to have a collection of sentences with proper annotation for evaluation. To that end, we describe work in developing an annotation scheme to encode syntactic relations in learner language.

2 Background

2.1 POS annotation of learner data

Díaz-Negrillo et al. (2009) address the notion of part-of-speech (POS) annotation for interlanguage, distinguishing three types of evidence needed to capture it: 1) stem/lexicon lookup, 2) morphological cues, and 3) word distribution. Because a learner does not always use a word in a native way, these pieces of evidence can conflict, and thus a POS annotation scheme which conflates all three, such as one developed for native language, cannot properly annotate learner language. For instance, in (1) (=(15) in Díaz-Negrillo et al. (2009)), there is a mismatch in the word's morphology (past tense) and its distribution (past participle).

(1) it has **grew** up a lot specially after 1996

While this serves as an excellent starting point, only POS annotation is explored, and this is done by examining an error-annotated corpus and seeing what POS information is necessary to account for the different error types. We start in the opposite direction, assuming no error annotation, and we develop a syntactic dependency annotation scheme for learner language.

2.2 Parsing of learner data

Although learner corpora have not generally been annotated with syntax, there is much work on syntactically parsing learner data (e.g., Vandeventer Faltin, 2003), to detect and diagnose ill-formed structures, such as non-agreement between a subject and a verb. For example, Menzel and Schröder (1999) use weighted constraints to derive dependency structures for learner language, and constraints which are violated are used to recover error information about the sentence. The sentence is robustly parsed, so the resulting dependency structure in some sense captures interlanguage.

However, it is not clear what exactly the surface syntax is encoding, as the parse is based on a model of native language. Further, it is unlikely that surface dependencies (or constituencies) capture the full set of syntactic facts employed by a learner. We return to these issues in section 4.1 after outlining general properties of our annotation scheme, including POS.

3 Annotation Scheme

For our pilot annotation, we use a collection of essays, from learners of different levels, from the early 1990s. The learners watched a short cartoon (*Tin Toy*) and were asked to discuss what happened. Examining this data has revealed several major theoretical issues for annotating second language data, which we outline in this paper.

3.1 General principles

In developing an annotation scheme, we followed several general guiding principles. The first principle is to annotate the language *as is*, i.e., annotate only what is there. We want to make as few claims as possible about what the intended meaning of the learner is, aiming only at an adequate description of the learners' interlanguage, from which researchers can draw their own conclusions. Thus, we do not posit empty elements or corrected forms, and errors do not exist directly in the corpus (though, see section 5). Further, since we build from previous annotation schemes, if a particular tag requires us to infer properties about the learner which we cannot possibly know, we prefer to adjust the annotation scheme.

Secondly, we try to give the learner the benefit of the doubt. For example, in (2), it is not entirely clear how the learner is trying to use *fraid*. We mark it as being a predicate, with *toy* as its subject, and do not attempt to posit any particular non-native properties, such as marking *(a)fraid* as a verb.[1]

(2) The toy **fraid**.

Since we stay as close to the original text as possible, we do not resegment run-on sentences. However, since these sentences have multiple syntactic roots (see section 4.2), they can easily be divided into smaller units.

3.2 Basic annotation

To make the corpus maximally useful for searching, we include lemma information for each word. This normalizes a word across all its realizations, including spelling mistakes. As we will see, it also allows us to capture properties akin to the stem information in Díaz-Negrillo et al. (2009).

We then annotate the corpus using the SUSANNE tagset (Sampson, 1995), as it distinguishes properties potentially of interest for SLA research, e.g., transitive and intransitive verbs, countable and uncountable nouns, and definite and indefinite articles. We have split the POS annotation into two parts, however. Namely, we have one POS tag to refer to the linguistic *form* of a word, which generally refers to its morphological features, and another POS tag to refer to the syntactic *use* of a word. This is essentially the same as the distinction between morphological and

[1] We still capture the fact that the text is non-native by encoding an empty subcategorization list, as with any other predicative (with or without a copula); see section 4.3.

distributional evidence in Díaz-Negrillo et al. (2009). In example (3), for instance, *makes* has the morphological form of 3rd person singular present tense (VVZt), but, in its position following *can*, its use is as a baseform verb (VV0t).

(3) Tin Toy can **makes** different music **sound**.

We leave a tag underspecified if it is not clear how the token is being used, as we can see for the word *sound* in (3). The form is clearly singular (NN1c), but the learner may be using this form as either a singular or plural noun (NN1c or NN2). Therefore, the *use* tag is underspecified (NN).

Where we differ from Díaz-Negrillo et al. (2009) is in not encoding a *stem* tag. This is potentially problematic, as the discrepancies between the inherent stem properties and the morphology account for realizations such as *choiced* in (4) (=(9) in Díaz-Negrillo et al. (2009)). However, we have recourse to the lemma information, which is essentially a stem lookup. One possibile solution in this particular case is to make the lemma *choose* (instead of *choice*), with the form and use tags as participles.

(4) ... to be **choiced** for a job ...

New & redefined tags We have had to make some changes to the tagset to account for learner language. Firstly, we use compound tags for words which have been merged. In (5), for instance, we have the POS use tag of AT1+NN1c for *adram*, which seems to be a blend of *a drum*.

(5) The tin toy had **adram** and a acordion.

Secondly, in the interest of underspecification, we have added some tags. With verbs, for instance, there are contexts where more than one tense is possible, as in (6). In this case, we do not know the specific POS use of *follow*; we only know that it is tensed.[2] The SUSANNE tagset specifies particular tense properties (e.g., D for past tense); to bypass making this decision, we define an underspecified VVTt, where T stands for a tensed verb.

(6) The child **follow** him.

3.3 Annotation format

We annotate the corpus with the standard CoNLL format, as in figure 1, which allows for dependency relations (Buchholz and Marsi, 2006). Thus far, we have described columns 1-5 (id, word, lemma, form POS, use POS); the remaining columns are described in the next section.

[2] The context of the video can help disambiguate some learner ambiguities, but the annotation scheme is necessary in cases where it cannot.

```
1 Tin       tin       NP1x NP1x _                 2 MOD   _ _
2 Toy       toy       NP1x NP1x _                 4 SUBJ  _ _
3 can       can       VMo  VMo  _                 4 AUX   _ _
4 makes     make      VVZt VV0t <SUBJ, AUX, OBJ>  0 ROOT  _ _
5 different different JJ   JJ   _                 7 MOD   _ _
6 music     music     NN1u JJ   _                 7 MOD   _ _
7 sound     sound     NN1c NN   _                 4 OBJ   _ _
8 .         .         YF   YF   _                 4 PUNCT _ _
```

Figure 1: Example CoNLL format

4 Dependency Relations

4.1 Capturing interlanguage syntax

We mark dependency relations between words, as this annotation captures many grammatical properties relevant to language acquisition, such as agreement of morphosyntactic features (e.g., Parodi, 2000) and argument structure (e.g., Mellow, 2008). It is also multi-lingual, making the scheme adaptable for use with other languages and readily available for dependency parsing (see, e.g., Buchholz and Marsi, 2006). Furthermore, encoding dependencies can be done more quickly than with constituencies.

Orthogonal to the decision to encode dependency relations, however, we must ask what types of evidence need to be brought to bear on learner language. What evidence determines the syntactic usage of a word, especially when that usage may be non-native? Consider, for instance, the constructed example (7). We know *He* in (7a) is the subject because of the morphological case marking of nominative, and the syntactic distribution indicates that nominals preceding verbs are possible subjects.

(7) a. **He** wants to save his life.
b. **Him** wants to save his life.

We thus want to consider **distributional** evidence in encoding syntactic properties. Languages vary in to what degree distribution is determined by word order and to what degree by morphological markings. For English, we place a greater emphasis on word order, or positional, information for determining grammatical relations, for a few reasons. First, English case marking is very deficient, found only on pronouns. Secondly, with a solid POS platform, morphological discrepancies are largely already accounted for: in (7b), for instance, we have a accusative POS form with a nominative POS use. Thirdly, as we will see below, when the morphology conflicts with the word order, this can be accounted for via other means. The syntactic distributional evidence we propose to use is essentially what is encoded by a surface dependency annotation scheme.

As a second piece of evidence, we need to consider inherent lexical properties of a word. Consider the difference between the constructed examples (8a) and (8b): in both cases, *couch* is acting as an object of the verb. The difference is in the **subcategorization** properties of the two verbs: *owns* selects for a subject and object (<SUBJ,OBJ>), while *snores* only selects for a subject (<SUBJ>). While words can be ambiguous in their subcategorization frames, there is generally only one acceptable for a given sentence.

(8) a. He **owns** a couch.
    b. He **snores** a couch.

Consider now the real learner example in (9). The word *dull* (assuming its intended form *doll*) is ambiguous: it could be an object of *escape* (with a missing subject), or it could be the subject in the wrong location. By biasing the distributional evidence towards positional information, we can say that *dull* is an object, but it seems like the learner simply misplaced the subject.

(9) ...escape the dull [doll]

One could underspecify the dependency labels (e.g., Carroll et al., 2003)—e.g., ARG instead of SUBJ/OBJ—but the issue is really that of putting together a complete argument structure for this verb. Giving the learner the benefit of the doubt, they have all the elements to form a well-formed semantic meaning from this sentence (cf. *the doll escaped*). There is a notion of what we will call **underlying dependencies** to account for. Using the argument structures and semantics of the words in the sentence as evidence, all the argument slots can reasonably be filled. This is in line with approaches mapping to underlying functor-argument structures (e.g., Sgall et al., 2004; Kromann, 2003).

In example (9), for instance, the underlying dependency of *dull* is as a subject of *escape*, even though the surface dependency is an object. A *dull* (*doll*) is an acceptable syntactic and semantic argument of *escape*, and so this is the relation we want if the goal is to be able to create a syntactically and semantically well-formed structure.

Underlying dependencies are useful not only for encoding learner ambiguities, but also long-distance dependencies. Given the generally-held assumption that each word has only one head in a dependency graph, surface dependencies cannot capture every relationship. Consider the constructed example (10a), for instance. Here, the verb *owns* is in a relative clause, with its subject immediately on the left. The verb is transitive, however, with its object left-dislocated, and surface dependencies do not account for this. This is a crucial piece of information, when one considers an intransitive verb such as *snores*, as in (10b), where both verbs have a long-distance object (*couch*).

(10) a. The couch that he **owns** burned.
     b. The couch that he **snores** burned.

One final piece of evidence is that of **collocations**, where one word selects another. In (11), for example, *interest* collocationally restricts the range of following prepositions, making the string *interest with* sound non-native.

(11) The baby had no more interest **with** the tin toy.

4.2 Encoding distributional (surface) dependencies

With the CoNLL format, it is straightforward to encode surface dependencies: columns 7 and 8 mark the head of each word and the relationship. The surface dependency tree for figure 1, for example, is shown in figure 2.[3]

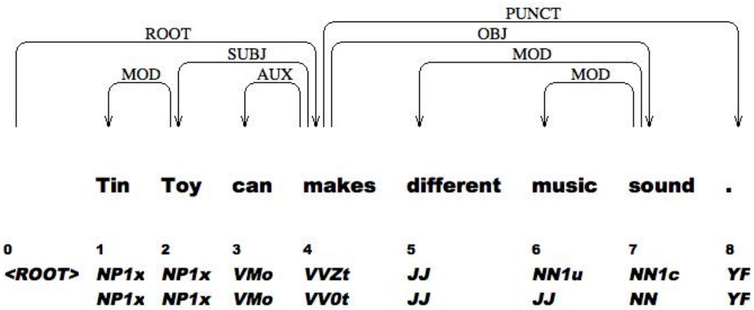
PUNCT
ROOT
OBJ
SUBJ
MOD
MOD
AUX
MOD
Tin Toy can makes different music sound .
0 1 2 3 4 5 6 7 8
<ROOT> NP1x NP1x VMo VVZt JJ NN1u NN1c YF
NP1x NP1x VMo VV0t JJ JJ NN YF

Figure 2: Example dependency tree

The scheme used to mark these relations is the one used for the CHILDES database of first language learners (Sagae et al., 2004, 2007). This was chosen because it was developed for language "in progress," and it encodes fairly specific grammatical relations, a total of 37 relations. Furthermore, if needed, the scheme can be mapped to a scheme which allows for underspecification of grammatical relations (Carroll et al., 2003).

New & redefined labels Despite the usefulness of the CHILDES annotation scheme, we have had to make some minor alterations. For example, we have added a label APPOS, to account for appositions, such as in (12), where *and* is an APPOS dependent of *things*.[4]

(12) thir are two *things* besaid the Toy, raings **and** string of beads.

[3] We use MaltEval (Nilsson and Nivre, 2008) for displaying trees.

[4] In coordinate structures, *and* is the head, a decision we follow for convenience.

Secondly, we have clarified the use of X and C before various labels (SUBJ, COMP, PRED, JCT). According to the documentation,[5] C indicates a finite clause and X a non-finite clause, which usually corresponds to C clauses having a subject and X not. In our data, however, a subject sometimes accompanies a non-finite verb or there might be no verb, so neither label technically fits the data we have. In (13), for example, *save* is some type of COMP of *want*, but it is not clear what its finiteness properties are.

(13) ... he *want* **save** his life.

We therefore re-define X and C categories, to focus on the presence or absence of subjects: C indicates the presence of a subject and X the absence. This is more straightforward to determine in English learner data than verb tense, and our goal is to make as few decisions as possible. This definition also seems to be in line with the intention of the annotation scheme (see the discussion of XCOMP parser errors in Sagae et al., 2007, sec. 5).[6]

Underspecifying information A full decision cannot always be made for the dependency graph. In (14), for example, we find the extraneous word *at*, and it is not clear where to put it in the dependency tree. In such cases, we attach the word to the virtual root and assign it no dependency label.

(14) He begins to walk and **at** to run.

4.3 Encoding subcategorization

We encode subcategorization frames in slot 6 of the CoNLL files, a spot for "other syntactic and morphological features." This is encoded as a list of dependency relations, e.g., <SUBJ, OBJ>. The only encoded elements are selected arguments and not adjunct dependents. If a word selects for nothing, we use '_' to indicate an empty list (<>).

Since cases are ambiguous, we encode the closest possible match. For *the toys*, for example, the subcategorization value of *toys* is <DET>, while in the bare NP *toys*, it is _. When a case is not clear, we give the learner the benefit of the doubt. For instance, in (15) (=(3)), the subcategorization for *sound* could be either <DET> or _, depending on if the learner intended to use a singular or plural noun. We label it _, with the number discrepancy already accounted for with POS (form=NN1c, use=NN).

(15) Tin Toy can makes different music **sound**.

[5] `http://www.cs.cmu.edu/~sagae/childesparser/childes-annotation.pdf`

[6] This is a fairly English-specific decision and would need to be adapted for languages which drop subjects; similar, consistent solutions can likely be found.

4.4 Encoding underlying dependencies

To encode underlying dependencies, we have decided only to encode those relations which are not already encoded at the surface level. Any relation already encoded as a surface relation is also interpreted as an underlying relation; if a relation covers the same pair of words as the surface syntax does, then the underlying relation overwrites the surface one (see (9)).

We encode this information in column 10 of the CoNLL files, using a list of pairs of head positions and relations. We cannot encode a single-headed dependency tree structure, as with surface relations, since a word may have several heads. For example, in (16), *I* serves as the subject of three different verbs, illustrated in figure 3: the surface subject of *hope* (7) and the underlying subject of *do* (9) and *enjoy* (12). The effect of this is also that the subcategorization lists for those verbs can be seen to be saturated.

(16) Now the only thing **I** *hope* to *do* is to *enjoy* him very well . . .

```
1  Now   now   RTo    RTo    _                  10 JCT   _ _
2  the   the   AT     AT     _                  4  DET   _ _
3  only  only  JBy    JBy    _                  4  MOD   _ _
4  thing thing NN1c   NN1c   <DET>              10 SUBJ  _ <9,OBJ>
5  that  that  CST    CST    _                  7  CPZR  _ _
6  I     I     PPIS1  PPIS1  _                  7  SUBJ  _ <(9,SUBJ),(12,SUBJ)>
7  hope  hope  VV0t   VV0t   <SUBJ, XCOMP>      4  CMOD  _ _
8  to    to    TO     TO     _                  9  INF   _ _
9  do    do    VD0    VD0    <SUBJ, INF, OBJ> 7  XCOMP _ _
10 is    be    VBZ    VBZ    <SUBJ, XPRED>      0  ROOT  _ _
11 to    to    TO     TO     _                  12 INF   _ _
12 enjoy enjoy VV0t   VV0t   <SUBJ, INF, OBJ> 10 XPRED _ _
13 him   he    PPHO1m PPHO1m _                  12 OBJ   _ _
14 very  very  RG     RG     _                  15 JCT   _ _
15 well  well  RR     RR     _                  12 JCT*  _ _
```

Figure 3: A CoNLL example with underlying dependencies

Encoding all underlying dependencies in a single list preserves the CoNLL format. However, this can easily be mapped to a more principled representation, such as DeccaXML (Boyd et al., 2007), to allow multiple heads.

4.5 Encoding collocations

Collocational knowledge is encoded by affixing an asterisk to the relevant dependency relation when words are used non-natively. This allows us simply to mark awkward collocations without having to develop a robust theory of them. In figure 3 above, for example, we can see a slightly unnatural-sounding collocation of *enjoy well*, marked as a JCT* relation.

5 Mapping to errors

Our goal has to been to annotate learner language as it appears, but, to maximize the utility of the corpus, we want to be able to map our representation to one which is also useful for developing error detection systems or the study of learner errors (see, e.g., Ellis, 1994). We thus sketch here how mismatches between annotation levels point to errors.[7]

Form tag $\neq$ Use tag As mentioned previously, discrepancies between form and use POS tags can point to non-native usage. In (17), for example, *crawl* is form-annotated as VV0i (baseform verb) and, in this context, use-annotated as VVDi (past tense).

(17) A baby **crawl** around the room . . .

Surface $\neq$ Underlying dependencies With both surface and underlying dependencies, we can identify those positions which conflict. For example, in (18), *baby* is marked as the surface object (OBJ) of *apparer*; the same positions have an underlying dependency of subject (SUBJ), however, pointing to some type of misuse. The issue here might be with word order or with argument structure; by annotating the layers separately, we point to the error without having to draw a particular conclusion about its nature.

(18) Then to *apparer* a **baby**.

Subcategorization $\neq$ Underlying dependencies Since the subcategorization lists indicate which arguments are selected for by a word, we can compare the list of potential arguments against those that are actually realized. We must check against the underlying dependents, not the surface dependents, because only in the underlying dependents do we find long-distance relations. This will capture sentences with missing, extra, or wrong arguments, as all are mismatches. An example of a missing argument is given in (19) (=(13)), where the infinitival *to* is missing. In this case, the subcategorization value of *save* is <SUBJ, INF, OBJ>, yet only the SUBJ and OBJ are realized (*he* and *life*, respectively).

(19) ... *he* want **save** his *life*.

To fully detect these errors, we have to know which types of dependents are arguments and which are adjuncts. For example, a JCT (adjunct) relation is a legitimate dependent of verbs, but is not selected. This has been clear-cut so far and can be ensured by splitting any ambiguous categories into separate argument and adjunct subcategories.

[7] We use the term *error* for any non-native usage, but ascribe it no theoretical status.

Limitations Currently, there are some aspects of learner language that we do not deal with, or in only limited ways. Anomalous word orders of adjuncts, for example, are not treated by our scheme.[8] In (20), for instance, the placement of *now* is odd. Currently, we mark such cases very cursorily, simply by adding a + sign on the dependency label (e.g., JCT+).

(20) He can't see **now** nothing.

Secondly, we do not handle semantic and pragmatic anomalies. In (21) (=(16)), for instance, *enjoy* should have probably been something like *entertain*. If it results in an anomalous argument structure, we will catch some of these errors indirectly, but otherwise we do nothing to mark them.

(21) Now the only thing I hope to do is to **enjoy** him very well …

6 Summary & Outlook

We have introduced a new annotation scheme capturing dependency relations for learner language. Specifically, we saw the need for encoding subcategorization frames and what we have termed underlying dependencies, in addition to the usual surface dependencies. We also needed to make clear exactly what distributional properties surface dependencies relate to. This scheme allows us to encode learner language as it appears, but also allows the annotation to be mapped to errors.

The next steps are straightforward, starting with continuing to annotate our pilot data, refining the annotation scheme as needed, e.g., splitting up argument and adjunct labels and fleshing out word order issues. This process will be well-documented, leading to a set of guidelines, and we will also test inter-annotator agreement. We are now collecting new data which can be released publicly (so researchers can study, e.g., the most frequent mistakes).

Acknowledgements

We wish to thank Kathleen Bardovi-Harlig, Rex Sprouse, and Detmar Meurers for useful discussion, as well as three anonymous reviewers and the IU Computational Linguistics discussion group.

References

Boyd, A., Dickinson, M., and Meurers, D. (2007). On representing dependency relations – insights from converting the german tigerdb. In *Proceedings of TLT 2007*, pages 31–42, Bergen, Norway.

Buchholz, S. and Marsi, E. (2006). Conll-x shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X*, pages 149–164, New York City.

[8] For argument word order, see the discussion on surface and underlying dependencies.

Carroll, J., Minnen, G., and Briscoe, T. (2003). Parser evaluation: Using a grammatical relation annotation scheme. In Abeillé, A., editor, *Treebanks: Building and using syntactically annoted corpora*, chapter 17, pages 299–316. Kluwer Academic Publishers, Dordrecht.

Díaz-Negrillo, A., Meurers, D., Valera, S., and Wunsch, H. (submitted, 2009). Towards interlanguage POS annotation for effective learner corpora in SLA and FLT.

Ellis, R. (1994). *The Study of Second Language Acquisition*. Oxford University Press, Oxford.

Granger, S. (2003). Error-tagged learner corpora and CALL: A promising synergy. *CALICO Journal*, 20(3):465–480.

Kromann, M. T. (2003). The danish dependency treebank and the underlying linguistic theory. In *Proceedings of TLT-03*.

Mellow, J. D. (2008). The emergence of complex syntax: A longitudinal case study of the esl development of dependency resolution. *Lingua*, 118(4):499 – 521.

Menzel, W. and Schröder, I. (1999). Error diagnosis for language learning systems. *ReCALL*, pages 20–30.

Nilsson, J. and Nivre, J. (2008). MaltEval: An evaluation and visualization tool for dependency parsing. In *Proceedings of LREC-08*, Marrakech.

Parodi, T. (2000). Finiteness and verb placement in second language acquisition. *Second Language Research*, 16(4):355 – 381.

Pendar, N. and Chapelle, C. (2008). Investigating the promise of learner corpora: Methodological issues. *CALICO Journal*, 25(2):189–206.

Pienemman, M. (1998). *Language Processing and Second Language Development: Processability theory*. John Benjamins.

Sagae, K., Davis, E., Lavie, A., MacWhinney, B., and Wintner, S. (2007). High-accuracy annotation and parsing of childes transcripts. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 25–32, Prague.

Sagae, K., MacWhinney, B., and Lavie, A. (2004). Adding syntactic annotations to transcripts of parent-child dialogs. In *Proceedings of LREC-04*, Lisbon.

Sampson, G. (1995). *English for the Computer: The SUSANNE Corpus and Analytic Scheme*. Clarendon Press, Oxford.

Sgall, P., Panevová, J., and Hajičová, E. (2004). Deep syntactic annotation: Tectogrammatical representation and beyond. In *Proceedings of the HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 32–38, Boston.

Vandeventer Faltin, A. (2003). *Syntactic error diagnosis in the context of computer assisted language learning*. Thèse de doctorat, Université de Genève, Genève.

Wulff, S., Ellis, N. C., Roemer, U., Bardovi-Harlig, K., and LeBlanc, C. (2009). The acquisition of tense-aspect: Converging evidence from corpora and telicity ratings. *The Modern Language Journal*, 93(3):354–369.