Constructivist Tokenization for English

Allison Fan
Winston Churchill High School
allisonoliviaf@gmail.com

Weiwei Sun
Dept of Computer Science and Technology
University of Cambridge
ws390@cam.ac.uk

Abstract

This paper revisits tokenization from a theoretical perspective, and argues for the necessity of a constructivist approach to tokenization for semantic parsing and modeling language acquisition. We consider two problems: (1) (semi-) automatically converting existing lexicalist annotations, e.g. those of the Penn TreeBank, into constructivist annotations, and (2) automatic tokenization of raw texts. We demonstrate that (1) a heuristic rule-based constructivist tokenizer is able to yield relatively satisfactory accuracy when gold standard Penn TreeBank part-of-speech tags are available, but that some manual annotations are still necessary to obtain gold standard results, and (2) a neural tokenizer is able to provide accurate automatic constructivist tokenization results from raw character sequences. Our research output also includes a set of high-quality morpheme-tokenized corpora, which enable the training of computational models that more closely align with language comprehension and acquisition.

1 Introduction

Although theoretical linguists have been gradually shifting from lexicalism to constructivism, constructivist theories have been barely adapted by computational linguists and psycholinguists. In this paper, we demonstrate the relevance of constructivist approaches to Natural Language Processing (NLP) in the context of tokenization, specifically for English. Though constructivist approaches to text segmentation and treebank annotation have been proposed for some languages such as Hebrew (Tsarfaty and Goldberg, 2008) and Korean (Park, 2017), English tokenization has been viewed as a long-solved problem in NLP. In some NLP tasks, e.g. Neural Machine Translation, it has even been replaced with purely statistics–based approaches, such as Byte Pair Encoding subword tokenization (Sennrich et al., 2016). We, however, argue that existing tokenization methods are not sufficient for at least two subfields — semantic parsing and modeling language acquisition.

We firstly explore the feasibility of (semi-) automatically converting existing lexicalist annotations, such as those in the Penn TreeBank, into constructivist annotations. We demonstrate that simple heuristic rules are able to utilize gold-standard Penn Treebank part of speech tags to produce high-quality constructivist annotations even without manual cleaning, thus substantially increasing efficiency of the constructivist tokenization and tagging process.

Through our rule-based algorithm, we are able to automatically produce a set of silver-standard morpheme-tokenized and tagged corpora from the annotated phrase structure trees of the Penn Treebank (PTB; Marcus et al., 1993) and the CHILDES Treebank (CTB; Pearl and Sprouse, 2013). However, despite the relatively high levels of accuracy of the silver standard corpora, some level of manual annotation is still required to achieve gold standard accuracy.

We then study automatic tokenization for raw, unannotated texts. We built a long short-term memory model (LSTM; Hochreiter and Schmidhuber, 1997) that was able to produce highly accurate tokenization outputs from raw character sequences even when trained with a large portion of silver-standard data. The high performance of our LSTM model is particularly useful in automatically tokenizing texts when previously existing lexicalist annotations are not available.

2 Background–Motivation

2.1 Lexicalist vs Constructivist Approach

There are two main approaches regarding the relationship between morphology and syntax: the lexicalist approach and the constructivist approach. The lexicalist approach was first proposed by Chomsky (1970) and Halle (1973) and states that

Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023), pages 36 - 40

there are two separate and distinct components of grammar: the first component, known as the lexicon, in which complex words, or lexical categories, are formed from morphemes, and the second component, known as syntax, in which lexical categories form phrases and sentences. Lexicalism posits that lexical categories are the basic units of syntactic structure, and the smallest elements which can be manipulated by syntactic processes. The other, newer approach to syntax and morphology, known as anti-lexicalism or constructivism, expresses the view that there is no divide between the formation of words and the formation of phrases, and that therefore there is no significant distinction between morphemes and words at the syntactic level. According to constructivism, morphemes are the basic units of syntactic derivation, and semantic composition starts from morphemes rather than words. See Figure 1 for a comparison of syntactic analyses according to different theories.

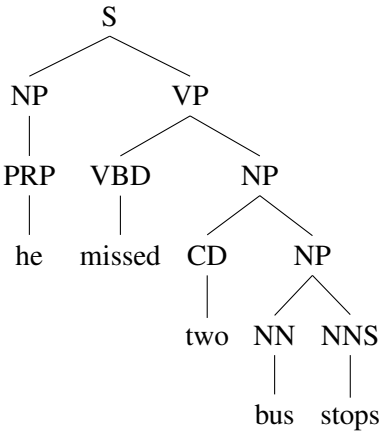
2.2 Relevance to Modeling Child Language Acquisition

A longitudinal study conducted by Brown (1973) of three First Language (L1) American English speaking children found that there was an approximately consistent order in which the children gradually incorporated morphemes into their speech. Table 1 is Brown's order of morpheme acquisition. The work done by Brown, as well as subsequent research on the order morpheme acquisition, demonstrates the importance of modeling morpheme acquisition. We believe that constructivist annotations are necessary to enable quantitative study in this direction.

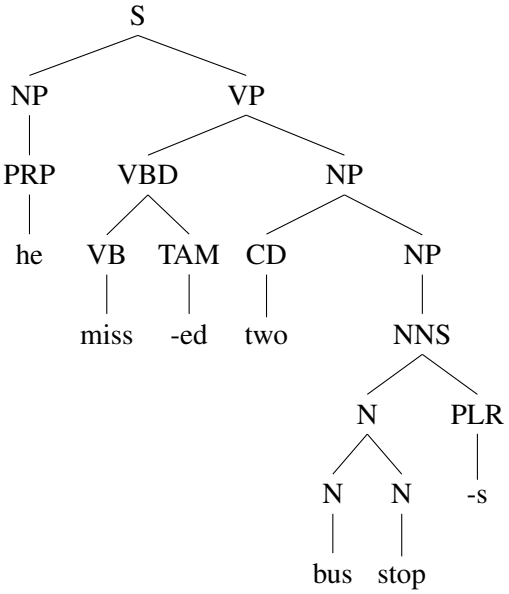
2.3 Relevance to Semantic Parsing

The earliest theory that draws on the ideas of constructivism is Distributed Morphology (DM; Halle, 1990; Halle et al., 1993; Halle, 1997; Harley and Noyer, 2003). One key concept in DM is Syntax All the Way Down — morphological elements can be manipulated by syntactic processes as they enter into the same types of constituent structures. Thus, semantic composition initializes from morphemes.

As seen in Figure 1, constructivist tokenization is able to better support semantic parsing, as morphemes, rather than words, correspond to the elementary units of syntactic-semantic composition. For example, in this case, the past tense -ed and the plural -s both convey additional meaning to their



(a) A lexicalist analysis.



(b) A constructivist analysis.

Figure 1: Contrasting analyses for *he missed two bus stops*. Node labels are practically adapted from PTB annotations.

Rank	Morpheme
1	Present progressive (*-ing*)
2-3	*in*, *on*
4	Plural (*-s*)
5	Past irregular
6	Possessive (*-'s*)
7	Uncontractible copula (*is*, *am*, *are*)
8	Articles (*a*, *the*)
9	Past regular (*-ed*)
10	Third person singular (*-s*)
11	Third person irregular
12	Uncontractible auxiliary (*is*, *am*, *are*)
13	Contractible copula
14	Contractible auxiliary

Table 1: Brown's order of L1 Acquisition of English. Table is from Kwon (2005).

lexical roots that is only able to be distinguished through further parsing to the morpheme level.

3 Rule-Based Tokenization

Dridan and Oepen (2012) presented a rule-based framework for pre-processing text prior to downstream tokenization and demonstrated the effectiveness of a Regular Expression-based approach to tokenization under the lexicalist framework. Inspired by their research, we study the feasibility of introducing a heuristic rule-based tokenizer that works downstream to word-based tokenization to further split PTB-style tokenized and POS-tagged text into functional morphemes, such as the n-categorizer, and lexical roots, following the minimalist theory.

3.1 Data Sources

To gauge our algorithm's accuracy for different types of inputs, our input data sources included both manually annotated gold-standard phrase structure trees as well as unprocessed raw utterance transcripts, detailed as follows:

Penn TreeBank The PTB data inputs consisted of gold-standard annotated phrase structure trees. Since the major parts of PTB have also been annotated with English Resource Semantics (Flickinger, 2000; Flickinger et al., 2014), resulting in DeepBank (Flickinger et al., 2012), the outputs of our system are well aligned to formal semantic annotations.

CHILDES TreeBank CTB is a corpus consisting of manually annotated phrase structure trees derived from child-directed utterance transcriptions in the North American English section of the CHILDES database. The phrase structure tree annotations in the CTB follow the format of PTB, with a few exceptions. CTB provided us with gold-standard child directed speech that more closely resembles the type of language children and infants are exposed to and thus allows us to more accurately model first language acquisition.

CHILDES Raw Texts The final type of input data we used is 'raw', unprocessed and untagged utterances from corpora in the North American English section of the CHILDES database. We separated our raw data inputs into two categories: child-directed speech transcriptions (CDS) and child-produced speech transcriptions (CPS).

3.2 Utilizing PTB POS Tags

Our dataset's tag scheme extracts the PTB-style POS tags and adapts them to label morphemes. We simplify the tags to be a simple POS tag that corresponds with the root of the word (eg VB for verb, N for noun) and an additional tag that marks the function morpheme suffixes of nouns, verbs and adjectives (eg TAM, or tense/aspect/mood for verb function morphemes, PLR for the plural morpheme). It is relatively straightforward to derive labels in regard to cutting-edge Minimalist theories, such as DIV(ide) further.

3.3 Lemmatization

One challenge encountered when tokenizing words into their morphemes was dealing with irregular words, which made it hard to come up with a streamlined set of rules to separate the function morpheme from the lexcial roots of words. Our solution for this issue was to use the WordNet Lemmatizer, which allowed us to get the root forms of nouns, verbs, and adjectives without extraneous morphemes regardless of irregularity. Our algorithm would then add the appropriate functional morphemes to the ends of the words, i.e. *-ed* for past tense verbs, *-s* for plural nouns, based on their original PTB tags.

3.4 Evaluation & Error Analysis

As seen in Table 2, the accuracy of our rule-based system output is largely dependent on the accuracy of the annotations provided in the original input data, as our algorithm bases its tokenization and tagging rules off of the given lexicalist annotations.

	PTB	CTB	CDS	CPS
# tokens	6,069	5,161	3,522	3,588
total # of errors in output	91	51	358	432
# of errors in original	66	29	343	406
# of lemmatization errors	19	12	8	20
# of errors from algorithm	6	10	7	6
% accuracy	97.81	99.01	89.84	87.96

Table 2: Breakdown of errors in the outputs of our rule-based system.

In addition, our algorithm itself introduces very few additional errors. For all four data sources used, the percentage of errors in the output not attributed to annotation errors in the original input data was less than 1% (0.412%, 0.426%, 0.426% and 0.725% for the PTB, CTB, CDS, and CPS data inputs, respectively).

Of the additional errors introduced by our rule-based system, a large portion stemmed from lemmatization. The three types of lemmatization errors observed to occur most frequently include plural nouns not lemmatized to their singular forms, improper lemmatization of present progressive (-ing) verbs, specifically those ending with an *e*, and improper lemmatization of certain irregular verbs that share a spelling with a verb of a different root form, such as *saw*, past tense of *see*, and present tense *saw*, meaning 'to cut'.

However, these cases are very word-specific and, once identified, can easily be fixed through additional, targeted rules to account for these exceptions within the program or through post-processing.

3.5 A Summary of Our Corpora

Gold Data We hand-checked the annotations of approximately **18,340** tokens outputted from our rule-based tokenizer to yield a gold-standard corpus tokenized with the constructivist approach.

# of tokens	Source
5,161	CTB (brown-adam)
3,522	CDS (bloom corpus)
3,588	CPS (bloom corpus)
6,069	PTB (wsj 0001-0018)

Table 3: Token counts, excluding punctuation, of our gold-standard corpora.

Silver Data We also used our rule-based tokenizer to automatically produce silver standard data from annotated CTB & PTB phrase structure trees. The accuracy of our silver-standard data is approximately 99% and 98% for the CTB and PTB respectively, as shown by the evaluation in Table 2.

# of tokens	Source
99,636	CTB (brown-adam)
274,606	CTB (brown-sarah)
108,189	CTB (hslld-hv1-mt)
30,717	PTB (wsj 00)

Table 4: Token counts, excluding punctuation, of our silver-standard corpora.

Bronze Data We also used our rule-based tokenizer to automatically produce bronze standard data from CHILDES raw texts. The accuracy of our silver-standard data is approximately 89% as shown by the evaluation in Table 2.

# of tokens	Source
176,700	CDS (bloom corpus)
126,286	CPS (bloom corpus)

Table 5: Token counts, excluding punctuation, of our bronze-standard corpora.

4 Neural Tokenisation

To fully automate tokenization from raw text inputs, we train a LSTM model with our manually cleaned gold-standard data as well as large-scale silver-standard data derived from the PTB and CTB phrase structure trees using our rule-based system. Our tokenizer is based on character labeling, in which the B(egin), I(nside), and O(utside) labels are used to encode the positional information of each character in an input sentence in regard to its position in its respective token. Experiments indicate that LSTM, together with our data, are effective in building a high performing constructivist tokenizer, which obtained an average accuracy of over 99%.

5 Conclusion and Future Work

This project demonstrated that automatic constructivist tokenization is feasible and can achieve high levels of accuracy, despite the complexity when going beyond words to morphemes. Although one might still need to manually clean resulting corpora to achieve gold standard accuracy, our rule-based tokenizer is able to substantially increase the efficiency of producing constructivist corpora. We also demonstrated that the use of deep learning,

such as LSTM models, can be a promising means of building a tokenizer. It is particularly useful in situations where the complexities / irregularity of a language become too difficult or cumbersome to codify into a rule based algorithm.

In future research, it would be interesting to explore how these two types of constructivist tokenizers perform in more complex, morpheme-heavy languages, such as Turkish. Another area of potential future research is to explore ways to enrich the corpora we produced in this project by adding syntactic and semantic annotations. The new corpora we produced will enable the next phase of research of building computational language acquisition models based on Constructivism. Our corpora will also allow future research in developing new Natural Language Understanding systems.

References

Roger Brown. 1973. Development of the first language in the human species. *American Psychologist*, 28(2):97–106.

N Chomsky. 1970. Remarks on nominalization. ra jacobs & ps rosembaum (eds.), readings in english transformational grammar. *Waltham Mass*.

Rebecca Dridan and Stephan Oepen. 2012. Tokenization: Returning to a long solved problem — a survey, contrastive experiment, recommendations, and toolkit —. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 378–382, Jeju Island, Korea. Association for Computational Linguistics.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Dan Flickinger, Emily M. Bender, and Stephan Oepen. 2014. Towards an encyclopedia of compositional semantics: Documenting the interface of the English Resource Grammar. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 875–881. European Language Resources Association (ELRA).

Daniel Flickinger, Yi Zhang, and Valia Kordoni. 2012. Deepbank: A dynamically annotated treebank of the wall street journal. In *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories*, pages 85–96.

Morris Halle. 1973. Prolegomena to a theory of word formation. *Linguistic inquiry*, 4(1):3–16.

Morris Halle. 1990. An approach to morphology. In *North East Linguistics Society*, volume 20, page 12.

Morris Halle. 1997. Distributed morphology: Impoverishment and fission. *MITWPL*, 30:425–449.

Morris Halle, Alec Marantz, Kenneth Hale, and Samuel Jay Keyser. 1993. Distributed morphology and the pieces of inflection. *1993*, pages 111–176.

Heidi Harley and Rolf Noyer. 2003. Distributed morphology. *The Second Glot International State-of-the-Article Book*, page 463–496.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.

Eun-Young Kwon. 2005. The "natural order" of morpheme acquisition: A historical survey and discussion of three putative determinants.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Jungyeul Park. 2017. Segmentation granularity in dependency representations for korean. In *International Conference on Dependency Linguistics*.

Lisa Pearl and Jon Sprouse. 2013. Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem. *Language Acquisition*, 20(1):23–68.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Reut Tsarfaty and Yoav Goldberg. 2008. Word-based or morpheme-based? annotation strategies for modern hebrew clitics. In *LREC*.