




Limitations of Tokenizers for Building a Neuro-Symbolic Lexicon

Hilton Alers-Valentín¹^a, José D. Maldonado-Torres²^b and J. Fernando Vega-Riveros²^c

¹Linguistics and Cognitive Science, University of Puerto Rico-Mayagüez, Puerto Rico

²Electrical and Computer Engineering, University of Puerto Rico-Mayagüez, Puerto Rico

{hilton.alers, jose.maldonado39, jfernando.vega}@upr.edu

Keywords: Tokenization, Computational Linguistics, Natural Language Processing, Symbolic Computation, Minimalist Syntax, Lexicon, Parser.

Abstract: Tokenization is a critical preprocessing step in natural language processing (NLP), as it determines the units of text that will be analyzed. Conventional tokenization strategies, such as whitespace-based or frequency-based methods, often fail to preserve linguistically meaningful units, including multi-word expressions, phrasal verbs, and morphologically complex tokens. Such failures result in downstream processing errors and hinder parsing performance. This paper examines contemporary tokenization approaches and their limitations in light of foundational concepts in morphology that are relevant for natural language parsing. We then proceed to describe the required features for the cognitive modeling of a human language lexicon and introduce a linguistically aware encoding pipeline. Finally, a preliminary assessment of this system will be presented and major points of the proposed system will be summarized in the conclusions.

1 INTRODUCTION

Before language processing is performed, “meaningful units” (or *tokens*) must be derived from the stream of characters that comprise an utterance (Arppe et al., 2005). This process, known as tokenization, is critical for parsing, as any issues presented in this stage may affect downstream processing tasks (Lu, 2014).


In an ideal scenario, a string of text would be perfectly punctuated, allowing tokenization to consist of splitting the text into tokens solely based on whitespace and punctuation marks (Arppe et al., 2005). On a surface level, this seems like a reasonable assumption for English utterances. However, this approach presents challenges for units that do not conform to conventional delimitations, such as idiomatic expressions, numerical expressions, phrasal verbs, acronyms, and proper nouns, among others. The improper management of these limitations poses concerns for parsing, as the tokens produced may lose their intended meaning when split. It would be unreasonable to analyze an utterance containing the adverb “upside down” as separate tokens. For example, for the phrase “they turned it upside down”:


- (1) Conventional tokenizers: [they, turned, it, upside, down].
- (2) Tokens: [they, turned, it, upside down].


For natural language parsing, lexical items are the real meaningful units. In this paper, we will address concerns posed by conventional tokenization schemes by first presenting foundational concepts in morphology that are relevant for natural language parsing (Section 2). Then, we will describe contemporary tokenization approaches (Section 3). Then we will proceed to describe the required features for the cognitive modeling of a human language lexicon (Section 4). Next, an encoder will be proposed that is designed to meet the previously described challenges (Section 5) and a preliminary assessment of this system will be presented (Section 6). Finally, major points of this proposal will be summarized in the conclusions (Section 7).

2 MORPHOLOGICAL FOUNDATIONS

A common mistake in tokenization is to equate tokens to words as “meaningful units” of natural language. Tokens are strings of textually contiguous characters that may or may not be separated by blank spaces or

^a <https://orcid.org/0000-0001-9057-7732>

^b <https://orcid.org/0009-0005-4454-002X>

^c <https://orcid.org/0009-0008-8523-2543>

some other form of punctuation. However, not all meaningful units are tokens, e.g., the item *cats* contains two distinct, smaller meaningful units: the root *cat* and the suffix *s*. The minimal linguistic sign, i.e., the atomic unit of language that carries meaning or conveys some grammatical feature is the *morpheme*. Generally speaking, morphemes are not free-standing syntactic objects as they have very strict and limited combinatorial capacities. Words, on the other hand, are the minimal free linguistic objects—abstract mental representations that are legible and not subject to orthographic conventions (any speaker has an intuition of what are words in their language, although throughout history, most speakers have been illiterate).

Although “word” seems to be a run-of-the-mill term, it is a very intuitive notion that is not so easy to formally define. In fact, “word” refers to two different concepts: a more concrete, inflected, externalized form and a more abstract mental representation of an atomic unit of language. For example, *forget*, *forgets*, *forgot*, *forgetting*, *forgotten* are, in one sense, five words, but in another sense, they seem to be forms of the same “word”.

A word is the smallest free linguistic unit, which may consist of a single morpheme (simple word) or two or more morphemes (complex word). The abstract (basic) form of a word is called a lexeme. The set of concrete phonetic forms of a lexeme constitutes the paradigm of the lexeme.

The combinatorial component of an I-Language is called a *grammar*, with computational procedures to form new objects, while the *lexicon* (LEX) is the set of lexical items (LI), the primitives or atoms of computation for I-Language. LIs are formatives “in the traditional sense as minimum “meaning-bearing” and functional elements” (Chomsky, 2021). Speakers possess a lexicon or mental dictionary that contains information about the morphosyntactic properties, meaning, and phonological representation of lexical items. It is conjectured that the variety of languages might be completely localized in peripheral aspects of LEX and in externalization (Chomsky, 2021).

3 CONTEMPORARY TOKENIZATION APPROACHES

The Penn Treebank (PTB) (Marcus et al., 1993) is a widely used annotated English corpus and the de facto tokenization standard (Dridan and Oepen, 2012). The tokenization strategy used by the PTB is rule-based and is summarized by (Fares et al., 2013) as follows:

whitespaces are explicit token boundaries, most punctuation marks are split from adjacent tokens, contracted negations are split into separate tokens (e.g., *won't* → *wo n't*), and hyphenated words and ones containing slashes are not split.

Contemporary artificial intelligence approaches, such as bidirectional encoder representations from transformers (BERT) (Devlin et al., 2019) and the generative pre-trained transformer (GPT) (Radford et al., 2018), rely on subword tokenization strategies. These strategies are used to mitigate the effects of rare or out of vocabulary (OOV) words on the model (Wu et al., 2016). To this end, BERT uses WordPiece for tokenization, while GPT uses byte pair encoding (BPE).

In WordPiece and BPE subword splits are determined by frequency-based heuristics: subwords that occur more frequently are maintained as single units, while less frequently occurring subword units are split (Wu et al., 2016). This process is repeated over a dataset until a threshold is reached (Wu et al., 2016). The subword splits created are not linguistically motivated, and thus may not align with the syntactic or morphological structures, specially in the case of less-represented languages or dialects.

4 MODELING A NATURAL LANGUAGE LEXICON

The lexicon is the language module that contains the grammatical information about all lexical items in the utterances to be analyzed by the parser. For a cognitive model of human language following SMT, it is “the heart of the implemented system” (Fong, 2005). Following the Chomsky-Borer hypothesis, Minimalist Grammars (MGs) locate all language-specific variation in the lexicon. Hence, every MG is just a finite set of lexical items. Each lexical item takes the form $A :: \alpha$, where A is the item’s phonetic exponent and α its string of features (Graf, 2021).

4.1 Features

Since the parser has to determine whether a certain combination of words is licensed or grammatical in the language, the lexicon must contain linguistically relevant properties of each lexical unit or features. A (morphosyntactic) feature, the basic unit or formative of syntax, is a property of a lexical item to which syntax is sensitive (such as agreement, which establishes a relationship between features). Features can determine the specific form of a word (through morphological operations such as affixation, metaphony, or

suppletion). They are the fundamental elements of natural languages, linking form and meaning. Features can be interpretable if they affect the semantic interpretation of a word or uninterpretable if they do not. Generally, features affect the morphology (form), semantics (interpretation), and syntax (relationships) of a word (Adger, 2003). Interface rules transduce a syntactic structure consisting of features to a morphological (and eventually phonological) structure on one hand, and to a semantic interpretation on the other, with morphosyntactic features acting as links between sound and meaning. Therefore, to propose the existence of a feature, at least one of the following must be observed:

- (3)(i) there are relationships among morphological forms of words,
- (ii) there is an effect on semantic interpretation, or
- (iii) a syntactic relationship must be established without which incorrect grammatical predictions would occur

Finally, in modeling a human language lexicon, one must consider that the set of features may:

- (4)(i) be universally available,
- (ii) be constructed by each speaker of a language, or
- (iii) be a combination of both.

For being the most restrictive approach, it is assumed that there exists a set of universal features, from which children select the relevant features for the language they are acquiring.

4.2 Feature Systems

Adger (2003) recognizes three possible feature systems:

- (5)(i) *Privative*: Languages have a default feature that appears when this type of feature is unspecified. In this system, features are privative, meaning they have no values and can either be present or absent. This system requires a default rule to supply a feature when it is unspecified. A rule of this type would be: [] → [singular].
- (ii) *Binary*: Features have binary values [+/-]. This system requires two constraints, i.e., that features always appear in bundles ([+sing, -pl]), and that features always appear with some value.
- (iii) *Non-Binary*: Features may take values other than [+/-]. For example, if [number:] were a feature, its values could be [singular] or [plural]. This system is simple, but does not allow

one to express the idea that a value (such as [dual]) is composed of other values ([singular] and [plural]).

For empirical considerations, the privative system would be preferable, as it is the simplest (Adger, 2003).

Features can also be interpretable or uninterpretable, depending on whether they are or not legible at the semantic interface. This distinction is crucial for syntax, as uninterpretable features need to be "checked" or eliminated throughout the construction of syntactic objects so that their derivation "converges" ((Adger, 2003).

4.3 Feature Typology

For each lexical item, according to its syntactic category, features may include agreement or phi-features (grammatical person, number, and gender), grammatical case, argument structure (category-selection features, thematic roles of predicates), and other relevant properties and denotational classes (such as common, proper, animate, personified), quantification, and deixis.

4.3.1 Categorical Features

The syntactic category is the grammatical feature that indicates the equivalence class of a lexical item or syntactic constituent and determines the grammatical properties (morphosyntactic features) and distributional properties (syntactic relationships) of that lexical item. There are two main types of syntactic categories. Lexical categories are lexical items that have referential or extralinguistic meaning, that is, their meaning refers to entities, properties, relationships, or events in some possible world (real or imaginary). Functional categories, on the other hand, do not have extralinguistic reference but have strictly grammatical/functional denotation.

- (6)(i) *Lexical categories*: nouns [N], verbs [V], adjectives [A], prepositions [P], adverbs [Adv]
- (ii) *Functional categories*: little *v*, Inflection [INFL], complementizers [C], coordinating connectors [Con]

4.3.2 Phi-Features

Typically nominal features, the bundle of grammatical person, number, and gender features are known as Phi-features. Other syntactic categories, like adjectives or determiners, may check these features via agreement.

- (7)(i) Person: 1, 2, 3

- (ii) Number: singular [sg], plural [pl]
- (iii) Gender: masculine [m], feminine [f], neuter [n]

4.3.3 Case Features

Case is a licensing feature for nominals. It is purely syntactic in function, with no semantic information, hence uninterpretable. Traditionally, Case is associated with grammatical functions (subject, direct object, indirect object, object of preposition). In many languages, case features are morphologically externalized, which may modify both phonological and orthographical representations. Two major case systems exist cross-linguistically, e.g. nominative-accusative (English, Spanish) and ergative-absolutive (Basque, Mayan) In Spanish pronouns, four cases can be observed:

- (8)(i) Nominative [nom]: *yo, tú, usted, él/ella, nosotros/as, vosotros/as, ustedes, ellos/as*
- (ii) Accusative [acc]: *me, te, lo/la, nos, os, los/las*
- (iii) Dative [dat]: *me, te, le/se, nos, os, les*
- (iv) Oblique [obl]: *mí, ti, usted, él/ella, nosotros/as, vosotros/as, ustedes, ellos/as*

4.3.4 TAM Features

Tense, Aspect, and Mood (TAM) are verbal features, typical of lexical verbs, auxiliaries, and modals.

- (9)(i) Tense: past [past], future [fut]
- (ii) Aspect: progressive [prog], perfective [perf], passive [pas], infinitive []

4.3.5 c-Selectional Features

Subcategorization or c(ategory)-selectional features are uninterpretable features of a lexical item that determine the syntactic category of the constituents that can merge with said lexical item to build a new syntactic object SO. For lexical categories, c-selectional features are linked to their argument structure. For example, adpositions (prepositions and postpositions) normally require a nominal [N] complement in order to form a constituent. Using a simple constituency test, one can observe that in a non-metalinguistic dialog (where participants are not talking about language), the preposition *with* cannot stand as a conversation line, but the phrase *with my friends* can (as an answer to *who were you with yesterday?*). On the other hand, the adverb *yesterday* can be by itself an appropriate contribution in a dialog (as an answer to *when did you visit your friends?*), but the string *yesterday my friends* cannot. This happens because *yesterday*

has no c-selectional features, hence it cannot be properly merged with any other syntactic object to form a constituent. This property of lexical items is represented as uninterpretable c-selectional features which must be checked by merging the lexical item with a complement of the same syntactic category of the lexical item's c-selectional feature. When a lexical item is merged with a constituent not c-selected by it, the derivation crashes and does not produce a legitimate syntactic object.

- (10)(i) with [P, uN] (preposition, selects a nominal complement)
- (ii) yesterday [Adv] (adverb, does not select any complement)

4.3.6 Thematic (Theta) Roles

Some mental concepts that are fossilized into linguistic meaning are said to have been lexicalized and, in a particular language, are called predicates (Adger, 2003). Some predicates need to be combined with other syntactic objects in order to express logical propositions. For example, although "eat" and "devour" have relatively similar meanings, the string "John ate this afternoon" is acceptable while "John devoured this afternoon" is not. This property is deeply linked to a predicate's c-selectional features and the semantic interpretation that the predicate assigns to each one of its arguments. These different semantic interpretations are known in the syntactic literature as *thematic, theta- or Θ -roles*. Some of the theta-roles that commonly appear in the literature are as follows, where the verb (in bold) is the predicate that assigns the theta role to the constituent in italics:

- (11)(i) Agent: *The boy* **kicked** the ball.
- (ii) Patient: The boy **kicked** *the ball*.
- (iii) Goal: The boy **gave** the ball *to his friend*.
- (iv) Location: The boy **put** the ball *in the box*.
- (v) Theme: *The ball* **moved**.

4.4 The Lexicon

The lexicon contains every lexical item appearing in a synthetic corpus of 2000 manually-tagged utterances that were constructed for validation purposes, containing acceptable and non-acceptable utterances, as well as structurally ambiguous and non ambiguous sentences. Lexical items are entered as a string of literals, and features are indicated by means of different data types. All lexical items are labeled with a syntactic category and their specific subset of features and lexical properties.

Since it is necessary to determine whether a certain combination of words is licensed or grammatical in the language, the lexicon should include every possible entry for each ambiguous lexical item. (Alers-Valentín et al., 2019). The property of selection and uninterpretable feature matching will drive the parsing process. In the course of computation, uninterpretable features belonging to analyzed constituents will be eliminated through probe-goal agreement. A (valid) parse is a phrase structure that obeys the selectional properties of the individual lexical items, covers the entire input, and has all uninterpretable features properly valued. (Fong, 2005).

5 PROPOSED ENCODER

To address the limitations of current tokenization strategies and arrive closer to a model of the human lexicon, we propose an encoding scheme designed to preserve linguistic integrity and enhance parsing performance. This encoding scheme consists of a pipeline with the following steps:

- (1) Pre-processing
- (2) Named-entity recognition (NER)
- (3) Tokenization
- (4) Part-of-speech (POS) tagging
- (5) Morphological analysis
- (6) Lemmatization

Each of these components will be discussed in this section.

5.1 Pre-Processing

The pre-processing stage receives an input utterance and converts it into a standardized format for downstream processing. This involves tasks such as lower-casing, removing extraneous whitespace, and normalizing text to account for punctuation, contractions, or informal abbreviations. Such normalization ensures compatibility with the subsequent modules while retaining linguistic content.

5.2 Named-Entity Recognition (NER)

The NER component handles the detection and classification of proper nouns, which are a frequent source of tokenization errors. By identifying entities such as names, locations, dates, and numerical expressions, the system avoids splitting them inappropriately during tokenization. For example, “San Francisco” is

preserved as a single entity rather than being tokenized into two separate units.

5.3 Tokenization

The tokenization process splits input text into individual linguistic units. To this end, the tokenizer would include a rule-based approach, aided by a dictionary that includes certain problematic units such as phrasal verbs, numerical expressions, etc. The tokenizer could be expanded to include statistical models for automatic identification of meaningful units. This facilitates downstream processing by maintaining linguistic coherence.

5.4 Part-of-Speech (POS) Tagging

In the POS tagging phase, each token is assigned a syntactic category, such as noun, verb, adverb, etc. This categorization should use a POS tagging model that incorporates contextual information to disambiguate the most likely categories for each unit. For example, the word “lead” may be tagged as a noun in “lead pipe” but as a verb in “lead the team.” Accurate POS tagging is essential for syntactic parsing and further morphological analysis (Alers-Valentín et al., 2019; Alers-Valentín et al., 2023).

Modern embedding approaches, particularly those based on transformers (e.g. BERT, GPT), achieve context sensitivity through word embeddings. Word embeddings are used to create a dense continuous-valued vectors in a high-dimensional space for each token. In the case of GPT and BERT, the word embedding created for each token integrates positional and contextual information, allowing a single token to have a different representation based on its surrounding context. This is critical for ensuring that tokenized units retain their intended meaning across the categorization, particularly in utterances that contain tokens with polysemy.

5.5 Morphological Analysis

In the morphological analysis step, each token is examined to determine its internal structure, including affixes, roots, and inflections. This step is critical as serves as a method for getting features from a lexicon.

5.6 Lemmatization

The lemmatization phase reduces words to their dictionary forms or lemmas, facilitating generalization in lexicon lookup and parsing tasks. This step ensures

Table 1: A selection of functional heads adapted from (Alers-Valentín and Fong, 2024).

Functional head	uFeatures	Other	Spell-Out (English)
<i>Little v</i>			
v* (transitive) v _{unerg} (unergative) v _{unacc} (unaccusative) v _~ (be)	phi:Person,Number	ef(theta); value acc Case ef(theta) ef check theta ef check theta	be be
<i>Auxiliaries</i>			
prt (participle) prog (progressive) perf (perfective)	phi:Number; Case	ef ef	-ed -ing -en
<i>Inflection (INFL)</i>			
INFL _{fin:nonpast} (finite:non-past) INFL _{fin:past} (finite:past)	phi:Person,Number phi:Person,Number	ef; value nom Case ef; value nom Case	[1,sg]:-m, [2,sg]:-re, [3,sg]:-s, [...pl]:-re [1,sg]:-ed, [1,pl]:-ed, [2,-]:-ed, [3,sg]:-ed, [3,pl]:-ed
INFL _{inf} (non-finite)	phi:Person,Number	ef; value null Case	to
<i>Complementizer</i>			
C (declarative) C _e (decl embedded) C _Q (interrogative) C _{eQ} (int embedded) C _{rel} (relative)	T Wh; T Wh; T Wh; T	Local Extent (LE) head ef; LE head ef; LE head ef; LE head ef(wh); LE head	do do

that semantically equivalent forms are treated consistently, improving the alignment of linguistic data with the model’s representations.

6 PRELIMINARY RESULTS AND ASSESSMENT

A synthetic corpus comprising 1,920 manually annotated and tokenized English utterances was developed, which served as the ground truth for this experiment. The tokens produced by the proposed tokenizer were compared to those generated by the default Punkt tokenizer from the Natural Language Toolkit (NLTK).

Out of the 1,920 utterances, 84 utterances (approximately 4%) exhibited differences between the tokens labeled as ground truth and those produced by Punkt. Mismatches occurred with multi-word tokens, particularly in cases involving phrasal verbs (e.g., “up to,” “fell in love,” etc.) and other multi-word expressions. These results highlight the need for a tokenizer that can preserve the integrity of such expressions.

6.1 Assessment

We propose a set of assessments to evaluate the effectiveness of the tokenizer and encoder system outlined in Section 5. The assessments focus on measuring linguistic integrity, tokenization accuracy, and downstream natural language processing (NLP) task performance. The proposed evaluations include both qualitative and quantitative methods.

6.1.1 Qualitative Assessments

A qualitative evaluation can involve manual analysis of the tokenizer’s outputs to assess its ability to handle linguistically complex cases. Specific focus should be placed on the following:

- (1) **Preservation of Multi-word Expressions:** Test the tokenizer on idiomatic expressions, phrasal verbs, and proper nouns to ensure that these are tokenized as single units.
- (2) **Morphological Coherence:** Evaluate whether the system correctly identifies and retains the morphological structure of words, such as roots, affixes, and inflectional forms.
- (3) **Context Sensitivity:** Assess the performance of the part-of-speech (POS) tagging modules to ensure contextually appropriate tagging (e.g., distinguishing between “lead” as a noun or verb).

6.1.2 Quantitative Assessments

Quantitative evaluations should focus on benchmarking the tokenizer against existing systems using annotated corpora. Suggested metrics include:

- (1) **Tokenization Accuracy:** Compare the system’s token boundaries with a ground truth dataset. This dataset may be manually created and annotated.
- (2) **Parsing Accuracy:** Integrate the tokenizer with dependency and constituency parsers to evaluate parsing accuracy.

- (3) **Named-Entity Recognition (NER) Accuracy:** Measure the system’s ability to preserve named entities as single tokens by testing against a dataset annotated for NER.

7 CONCLUSIONS

In this paper, we have addressed critical limitations in conventional tokenization approaches that fail to preserve the integrity of linguistically meaningful units, such as multi-word expressions, phrasal verbs, and morphologically complex tokens. By analyzing foundational morphological concepts, contemporary tokenization strategies, and the requirements for modeling a human language lexicon, we proposed an encoding pipeline designed to bridge the gap between surface-level text processing and linguistically aware tokenization. The proposed pipeline incorporates pre-processing, named-entity recognition (NER), tokenization, part-of-speech (POS) tagging, morphological analysis, lemmatization, and word embeddings. Preliminary testing demonstrated the need for this approach, as a comparison between a manually annotated corpus and the output of NLTK’s Punkt tokenizer revealed that multi-word expressions, such as phrasal verbs, are a primary source of tokenization errors in existing systems. By preserving such expressions, the tokenizer shows promise in improving parsing performance and downstream natural language processing (NLP) tasks.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 2219712 and 2219713. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

- Adger, D. (2003). *Core Syntax: A Minimalist Approach*. Oxford University Press, Oxford.
- Alers-Valentín, H. and Fong, S. (2024). Towards a biologically-plausible computational model of human language cognition. In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence*, volume 3, pages 1108–1118. SciTePress.
- Alers-Valentín, H., Fong, S., and Vega-Riveros, J. F. (2023). Modeling syntactic knowledge with neuro-symbolic computation. In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence*, volume 3, pages 608–616. SciTePress.
- Alers-Valentín, H., Rivera-Velázquez, C. G., Vega-Riveros, J. F., and Santiago, N. G. (2019). Towards a principled computational system of syntactic ambiguity detection and representation. In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023)*, volume 2, pages 980–987. INSTICC, SciTePress.
- Arppe, A., Carlson, L., Lindén, K., Piitulainen, J., Suominen, M., Vainio, M., Westerlund, H., and Yli-Jyrä, A. (2005). Inquiries into words, constraints and contexts.
- Chomsky, N. (2021). Minimalism: Where are we now, and where can we hope to go. *Gengo Kenkyu*, 160:1–41.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.
- Dridan, R. and Oepen, S. (2012). Tokenization: Returning to a long solved problem — A survey, contrastive experiment, recommendations, and toolkit —. In Li, H., Lin, C.-Y., Osborne, M., Lee, G. G., and Park, J. C., editors, *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.
- Fares, M., Oepen, S., and Zhang, Y. (2013). Machine Learning for High-Quality Tokenization Replicating Variable Tokenization Schemes. In *Computational Linguistics and Intelligent Text Processing - 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part I*, pages 231–244.
- Fong, S. (2005). Computation with probes and goals. In *UG and External Systems: Language, Brain and Computation*, pages 311–334. John Benjamins, Amsterdam.
- Graf, T. (2021). Minimalism and computational linguistics.
- Lu, X. (2014). *Computational Methods for Corpus Annotation and Analysis*. Springer Publishing Company, Incorporated.
- Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.