Assessing Relative Sentence Complexity using an Incremental CCG Parser

Assessing Relative Sentence Complexity using an Incremental CCG Parser

Bharat Ram Ambati and **Siva Reddy** and **Mark Steedman**
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
bharat.ambati@ed.ac.uk, siva.reddy@ed.ac.uk, steedman@inf.ed.ac.uk

Abstract

Given a pair of sentences, we present computational models to assess if one sentence is simpler to read than the other. While existing models explored the usage of phrase structure features using a non-incremental parser, experimental evidence suggests that the human language processor works incrementally. We empirically evaluate if syntactic features from an incremental CCG parser are more useful than features from a non-incremental phrase structure parser. Our evaluation on Simple and Standard Wikipedia sentence pairs suggests that incremental CCG features are indeed more useful than phrase structure features achieving 0.44 points gain in performance. Incremental CCG parser also gives significant improvements in speed (12 times faster) in comparison to the phrase structure parser. Furthermore, with the addition of psycholinguistic features, we achieve the strongest result to date reported on this task. Our code and data can be downloaded from https://github.com/bharatambati/sent-compl.

1 Introduction

The task of assessing text readability aims to classify text into different levels of difficulty, e.g., text comprehensible by a particular age group or second language learners (Petersen and Ostendorf, 2009; Feng, 2010; Vajjala and Meurers, 2014). There have been efforts to automatically simplify Wikipedia to cater its content for children and English language learners (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011; Wubben et al., 2012; Siddharthan and Mandya, 2014). A related attempt of Vajjala and Meurers (2016) studied the usage of linguistic features for automatic classification of a pair of sentences – one from Standard Wikipedia and the other its corresponding simplification from Simple Wikipedia – into COMPLEX and SIMPLE. As syntactic features, they use information from phrase structure trees produced by a non-incremental parser, and found them useful.

However, psycholinguistic theories suggest that humans process text incrementally, i.e., humans build syntactic analysis interactively by enhancing current analysis or choosing an alternative analysis on the basis of the plausibility with respect to context (Marslen-Wilson, 1973; Altmann and Steedman, 1988; Tanenhaus et al., 1995). Besides being cognitively possible, incremental parsing has shown to be useful for many real-time applications such as language modeling for speech recognition (Chelba and Jelinek, 2000; Roark, 2001), modeling text reading time (Demberg and Keller, 2008), dialogue systems (Stoness et al., 2004) and machine translation (Schwartz et al., 2011). Furthermore, incremental parsers offer linear time speed. Here we explore the usefulness of incremental parsing for predicting relative sentence readability.

Given a pair of sentences – one sentence a simplified version of the other – we aim to classify the sentences into SIMPLE or COMPLEX. We use the sentences from Standard Wikipedia (WIKI) paired with their corresponding simplifications in Simple Wikipedia (SIMPLEWIKI) as training and evaluation data. We pose this problem as a pairwise classification problem (Section 2). For feature extraction,

Proceedings of NAACL-HLT 2016, pages 1051–1057,
San Diego, California, June 12-17, 2016.

we use an incremental CCG parser which provides a trace of each step of the parse derivation (Section 3). Our evaluation results show that incremental parse features are more useful than non-incremental parse features (Section 5). With the addition of psycholinguistic features, we attain the best reported results on this task. We make our system available for public usage.

2 Problem Formulation

Initially Vajjala and Meurers (2014) trained a binary classifier to classify sentences in SIMPLEWIKI to the class SIMPLE, and sentences in WIKI to the class COMPLEX. This model performed poorly on relative readability assessment. Noting that not all SIMPLEWIKI sentences are simpler than every other sentence in WIKI, Vajjala and Meurers (2016) reframed the problem as a ranking problem according to which given a pair of parallel SIMPLEWIKI and WIKI sentences, the former must be ranked better than the latter in terms of readability. Inspired by Vajjala and Meurers (2016), we also treat each pair together, and model relative readability assessment as a pairwise classification problem. Let a, b be a pair of parallel sentences. Let **a**, **b** represent their corresponding feature vectors. We define our classifier Φ as

$$\begin{aligned}\Phi(\mathbf{a}-\mathbf{b}) &= 1 \quad \text{if} \;\; a \in \text{SIMPLE} \;\; \& \; b \in \text{COMPLEX} \\ &= -1 \quad \text{if} \;\; b \in \text{SIMPLE} \;\; \& \; a \in \text{COMPLEX}\end{aligned}$$

The motivation for our modelling is that relative features (difference) are more useful than absolute features, e.g., intuitively shorter sentences are simple to read, but length can only be defined in comparison with another sentence.

3 Incremental CCG Parse Features

Below we provide necessary background, and then present the features.

3.1 Combinatory Categorial Grammar (CCG)

CCG (Steedman, 2000) is a lexicalized formalism in which words are assigned syntactic types encoding subcategorization information. Figure 1 displays an incremental CCG derivation. Here, the syntactic type (category) $(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP}$ on *ate* indicates that it is a transitive verb looking for a NP

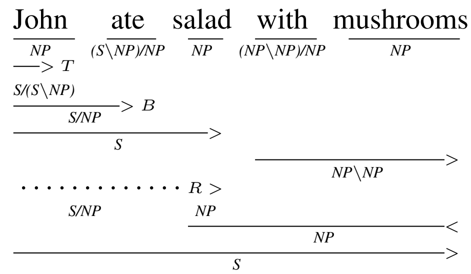

Figure 1: Incremental CCG derivation tree.

(object) on the righthand side and a NP (subject) on the lefthand side. Due to its lexicalized and strongly typed nature, the formalism offers attractive properties like elegant composition mechanisms which impose context-sensitive constraints, efficient parsing algorithms, and a synchronous syntax-semantics interface. In Figure 1, the category of *with* $(\mathrm{NP}\backslash\mathrm{NP})/\mathrm{NP}$ combines with the category of *mushrooms* NP on its righthand side using the combinatory rule of *forward application* (indicated by $>$), to form the category $\mathrm{NP}\backslash\mathrm{NP}$ representing the phrase *with mushrooms*. This phrase in turn combines with other contextual categories using CCG combinators to form new categories representing larger phrases.

In contrast to phrase structure trees, CCG derivation trees encode a richer notion of syntactic type and constituency. For example, in a phrase structure tree, the category (constituency tag) of *ate* would be *VBD* irrespective of whether it is transitive or intransitive, whereas the CCG category distinguishes these types. As the linguistic complexity increases, the complexity of the CCG category may increase, e.g., the relative pronoun has the category $(\mathrm{NP}\backslash\mathrm{NP})/(\mathrm{S}\backslash\mathrm{NP})$ in relative clause constructions. In addition, CCG derivation trees have combinators annotated at each level which indicate the way in which the category is derived, e.g., in Figure 1 the category S/NP of *John ate* is formed by first *type-raising* (indicated by >T) *John* and then applying *forward composition* (indicated by >B) with *ate*. CCG combinators can throw light into the linguistic complexity of the construction, e.g., *crossed composition* is an indicator of long-range dependency. Phrase structure trees do not have this additional information encoded on their nodes.

3.2 Incremental CCG

Ambati et al. (2015) introduced a shift-reduce in-

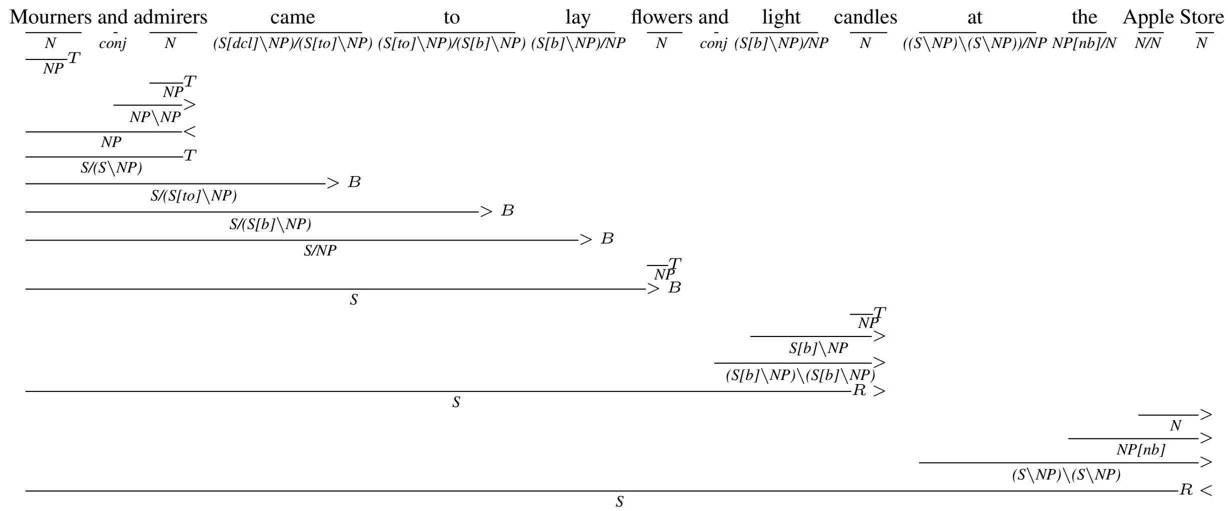

Figure 2: Incremental Derivation for a relatively complex sentence.

cremental CCG parser for English.[1] The main difference between this incremental version and standard non-incremental CCG parsers such as Zhang and Clark (2011) is that as soon as the grammar allows two types to combine, they are greedily combined. For example, in Figure 1, first *John* is pushed on the stack but is immediately reduced when its head *ate* appears on the stack (i.e., *John*'s category combines with *ate*'s category to form a new category), and similarly when *salad* is seen, it is reduced with *ate*. When *with* appears it waits to be reduced until its head *mushrooms* appears on the stack, and later *mushrooms* is reduced with *salad* via *ate* using a special *revealing* operation (indicated by R>) followed by a sequence of operations. The *revealing* operation is performed when a category has greedily consumed a head in advance of a subsequently encountered post-modifier to regenerate the head. In the non-incremental version, *salad* is not reduced with *ate* until *with mushrooms* is reduced with it.

Consider the following sentences (A) and (B) where (B) is a simpler version of (A).

(A)	Mourners and admirers came to lay flowers and light candles at the Apple Store.
(B)	People went to the Apple Store with flowers and candles.

Figures 2 and 3 present the incremental derivations for both these sentences. Consider the CCG category for *to* in both the sentences. In (A), the category of *to* is `(S[dcl]\NP)/(S[to]\NP)` which is more complex compared to the category of *to* in (B) which is `PP/NP`. Both the derivations have one right reveal action (indicated by $R >$). In (A), the depth of this action is two since it is a VP coordination.[2] Whereas in (B) the depth is only one. Such information can be useful in predicting the complexity of a sentence.

3.3 Features

As discussed above, as the complexity of a sentence increases, the complexity of CCG categories, combinators and the number of revealing operations increase in the incremental analysis. We exploit this information to assess the readability of a sentence. For each sentence, we build a feature vector using the features defined below extracted from its incremental CCG derivation.

Sentence Level Features. These features include sentence length, height of the CCG derivation, and the final number of constituents. A CCG derivation may have multiple constituents if none of the combinators allow the constituents to combine. This happens mainly in ungrammatical sentences.

CCG Rule Counts. These features include the number of applications, forward applications, back-

[1] This parser is not word by word (strictly) incremental but is incremental with respect to CCG derivational constituents following Strict Competence Hypothesis (Steedman, 2000).

[2] Please see Ambati et al. (2015) for additional information on the *depth* of revealing operations.

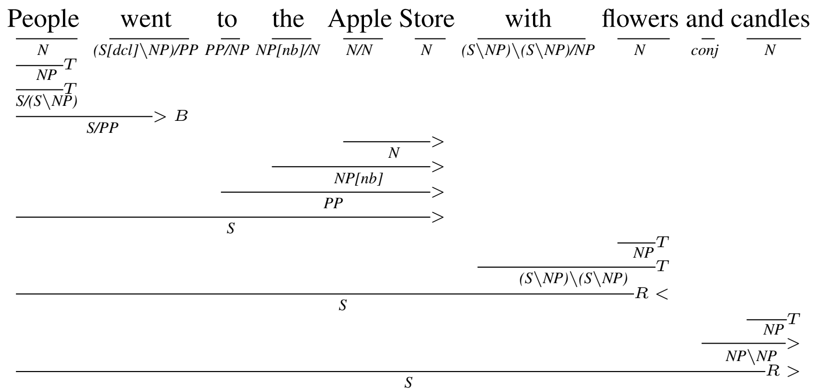

Figure 3: Incremental Derivation for a relatively simple sentence.

ward applications, compositions, forward compositions, backward compositions, left punctuations, right punctuations, coordinations, type-raisings, type-changing, left revealing, right revealing operations used in the CCG derivation. Each combinator is treated as a different feature dimension with its count as the feature value. For the revealing operations, we also add additional features which indicate the depth of the revealing which is analogous to surprisal (Hale, 2001).

CCG Categories. We define the complexity of a CCG category as the number of basic syntactic types used in the category, e.g., the complexity of $(\mathrm{S[pss]}\backslash\mathrm{NP})/(\mathrm{S[to]}\backslash\mathrm{NP})$ is 4 since it has one S[pss], one $S[to]$, and two NPs. Note that CCG type S[pss] indicates a *sentence* but of the subtype *passive*. We use average complexity of all the CCG categories used in the derivation as a real valued feature. In addition, we define integer-valued features representing the frequency of specific subtypes (we have 21 subtypes each defined as a different dimension) and the frequency of the top 8 syntactic types (each as a different dimension).

4 Experimental Setup

4.1 Evaluation Data

As evaluation data, we use WIKI and SIMPLEWIKI parallel sentence pairs collected by Hwang et al. (2015), a newer and larger version compared to Zhu et al. (2010)'s collection. We only use the pairs from the section GOOD consisting of 150K pairs. We further removed pairs containing identical sentences which resulted in 117K clean pairs. We randomly divided the data into training (60%), development (20%) and test (20%) splits.

4.2 Implementation details

As our classifier (see Section 2) we use SVM with Sequential Minimal Optimization in Weka toolkit (Hall et al., 2009) following its popularity in readability literature (Feng, 2010; Hancke et al., 2012; Vajjala and Meurers, 2014).[3] We use Ambati et al. (2015)'s CCG parser for extracting CCG derivations. This parser requires a CCG supertagger to limit its search space for which we use EasyCCG tagger (Lewis and Steedman, 2014).

4.3 Baseline

NON-INCREMENTAL PST. Following Vajjala and Meurers (2016), we use features extracted from Phrase Structure Trees (PST) produced by the Stanford parser (Klein and Manning, 2003), a non-incremental parser. We use the exact code used by Vajjala and Meurers (2016) to extract these features which include part-of-speech tags, constituency features like the number of noun phrases, verb phrases and preposition phrases, and the average size of the constituent trees. Vajjala and Meurers (2016) used a total of 57 features.[4]

5 Results

First we analyze the impact of incremental CCG features (and so the name **INCREMENTAL CCG**).

[3] We also experimented with Naive Bayes and Logistic Regression and observed similar pattern in the results. But, SVM gave the best results among the classifiers we explored.

[4] Details of the features can be found in Vajjala and Meurers (2016).

Model	Accuracy
NON-INCREMENTAL PST	71.68
INCREMENTAL CCG	**72.12**

Table 1: Impact of different syntactic features.

Table 1 presents the results of predicting relative readability on the test data.[5] INCREMENTAL CCG achieves 72.12% accuracy, a significant[6] improvement of 0.44 points over NON-INCREMENTAL PST (71.68%) indicating that incremental CCG features are empirically more useful than non-incremental phrase structure features. We also evaluate if this result holds for incremental vs. non-incremental CCG parse features. Ambati et al. (2015) can also produce non-incremental CCG parses by turning off a flag. Note that in the non-incremental version, revealing features are absent. This version achieves an accuracy of 72.02%, around 0.1% lower than the winner INCREMENTAL CCG, yet higher than NON-INCREMENTAL PST showing that CCG derivation trees offer richer syntactic information than phrase structure trees. POS taggers used for Stanford and CCG parsers gave similar accuracy. This shows that the improvements are indeed due to the incremental CCG parse features rather than the POS features.

Apart from the syntactic features, Vajjala and Meurers (2016) have also used psycholinguistic features such as age of acquisition of words, word imagery ratings, word familiarity ratings, and ambiguity of a word, collected from the psycholinguistic repositories Celex (Baayen et al., 1995), MRC (Wilson, 1988), AoA (Kuperman et al., 2012) and WordNet (Fellbaum, 1998). These features are found to be highly predictive for assessing readability. We enhance our syntactic models NON-INCREMENTAL PST and INCREMENTAL CCG by adding these psycholinguistic features to build NON-INCREMENTAL PST++ and INCREMENTAL CCG++ respectively. Table 2 presents the final results along with the previous state-of-the-art results of Vajjala and Meurers (2016).[7] Psycholinguistic features gave a boost of around 6.75 points on the syntactic models.[8] Additionally the performance gap between our models decrease (from 0.44 to 0.19) showing some of the psycholinguistic features also model a subset of the syntactic features. INCREMENTAL CCG++ achieves an accuracy of 78.77% outperforming the previous best system of Vajjala and Meurers (2016) by a wide margin.

[5]All feature engineering is done on the development data.

[6]Numbers in bold indicate significant results, significance measured using McNemar's test.

[7]We ran Vajjala and Meurers (2016)'s code on our dataset and get similar results reported on Zhu et al. (2010)'s dataset.

Model	Accuracy
Vajjala and Meurers (2016)	74.58
NON-INCREMENTAL PST++	78.68
INCREMENTAL CCG++	**78.87**

Table 2: Performance of models with both syntactic and psycholinguistic features.

Speed. In addition to accuracy, parsing speed is important in real-time applications. The Stanford parser took 204 minutes to parse the test data with a speed of 3.8 sentences per second. The incremental CCG parser took 16 minutes with an average speed of 47.5 sentences per second, a 12X improvement over the Stanford parser. These numbers include POS tagging time for the Stanford parser, and POS tagging and supertagging time for the incremental CCG parser. All the systems are run on the same hardware (Intel i5-2400 CPU @ 3.10GHz).

6 Conclusion

Our empirical evaluation on assessing relative sentence complexity suggests that syntactic features extracted from an incremental CCG parser are more useful than from a non-incremental phrase structure parser. This result aligns with psycholinguistic findings that human sentence processor is incremental. Our incremental model enhanced with psycholinguistic features achieves the best reported results on predicting relative sentence readability. We experimented with Simple Wikipedia and Wikipedia data from Hwang et al. (2015). We can explore the usefulness of our system on other datasets like OneStopEnglish (OSE) corpus (Vajjala and Meurers, 2016) or the dataset from Xu et al. (2015). We are also currently exploring the usefulness of incremental analysis for psycholinguistic data by switching off the lookahead feature.

[8]Non-incremental CCG achieves an accuracy of 78.77%.

Acknowledgments

We thank Sowmya Vajjala and Dave Howcroft for providing data and settings for the baseline. We also thank the three anonymous reviewers for their useful suggestions. This work was supported by ERC Advanced Fellowship 249520 GRAMPLUS, EU IST Cognitive Systems IP Xperience and a Google PhD Fellowship for the second author.

References

[Altmann and Steedman1988] Gerry Altmann and Mark Steedman. 1988. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

[Ambati et al.2015] Bharat Ram Ambati, Tejaswini Deoskar, Mark Johnson, and Mark Steedman. 2015. An Incremental Algorithm for Transition-based CCG Parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 53–63, Denver, Colorado, May–June. Association for Computational Linguistics.

[Baayen et al.1995] R. H. Baayen, R. Piepenbrock, and L. Gulikers. 1995. *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

[Chelba and Jelinek2000] Ciprian Chelba and Frederick Jelinek. 2000. Structured language modeling. *Computer Speech & Language*, 14(4):283–332.

[Coster and Kauchak2011] William Coster and David Kauchak. 2011. Simple English Wikipedia: A New Text Simplification Task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA, June. Association for Computational Linguistics.

[Demberg and Keller2008] Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

[Fellbaum1998] Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

[Feng2010] Lijun Feng. 2010. *Automatic readability assessment*. Ph.D. thesis, City University of New York.

[Hale2001] John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 159–166. Association for Computational Linguistics.

[Hall et al.2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

[Hancke et al.2012] Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability Classification for German using Lexical, Syntactic, and Morphological Features. In *Proceedings of COLING 2012*, pages 1063–1080, Mumbai, India, December. The COLING 2012 Organizing Committee.

[Hwang et al.2015] William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning Sentences from Standard Wikipedia to Simple Wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 211–217, Denver, Colorado, May–June. Association for Computational Linguistics.

[Klein and Manning2003] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan, July. Association for Computational Linguistics.

[Kuperman et al.2012] Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 English words. *Behavior Research Methods*, 44(4):978–990.

[Lewis and Steedman2014] Mike Lewis and Mark Steedman. 2014. Improved CCG parsing with Semi-supervised Supertagging. *Transactions of the Association for Computational Linguistics (TACL)*, 2:327–338.

[Marslen-Wilson1973] W. Marslen-Wilson. 1973. Linguistic structure and speech shadowing at very short latencies. *Nature*, 244:522–533.

[Petersen and Ostendorf2009] Sarah E Petersen and Mari Ostendorf. 2009. A machine learning approach to reading level assessment. *Computer speech & language*, 23(1):89–106.

[Roark2001] Brian Roark. 2001. Probabilistic Top-Down Parsing and Language Modeling. *Computational Linguistics*, 27:249–276.

[Schwartz et al.2011] Lane Schwartz, Chris Callison-Burch, William Schuler, and Stephen Wu. 2011. Incremental Syntactic Language Models for Phrase-based Translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 620–631, Portland, Oregon, USA, June. Association for Computational Linguistics.

[Siddharthan and Mandya2014] Advaith Siddharthan and Angrosh Mandya. 2014. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731, Gothenburg, Sweden, April. Association for Computational Linguistics.

[Steedman2000] Mark Steedman. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA, USA.

[Stoness et al.2004] Scott C Stoness, Joel Tetreault, and James Allen. 2004. Incremental Parsing with Reference Interaction. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 18–25.

[Tanenhaus et al.1995] MK Tanenhaus, MJ Spivey-Knowlton, KM Eberhard, and JC Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268(5217):1632–1634.

[Vajjala and Meurers2014] Sowmya Vajjala and Detmar Meurers. 2014. Assessing the relative reading level of sentence pairs for text simplification. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 288–297, Gothenburg, Sweden, April. Association for Computational Linguistics.

[Vajjala and Meurers2016] Sowmya Vajjala and Detmar Meurers. 2016. Readability-based Sentence Ranking for Evaluating Text Simplification. In *arXiv preprint*.

[Wilson1988] Michael Wilson. 1988. MRC Psycholinguistic Database: Machine-usable dictionary, version 2.00. *Behavior Research Methods, Instruments, & Computers*, 20(1):6–10.

[Woodsend and Lapata2011] Kristian Woodsend and Mirella Lapata. 2011. WikiSimple: Automatic Simplification of Wikipedia Articles. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, pages 927–932, San Francisco, California, USA.

[Wubben et al.2012] Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence Simplification by Monolingual Machine Translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea, July. Association for Computational Linguistics.

[Xu et al.2015] Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

[Zhang and Clark2011] Yue Zhang and Stephen Clark. 2011. Shift-Reduce CCG Parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 683–692, Portland, Oregon, USA, June. Association for Computational Linguistics.

[Zhu et al.2010] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China, August. Coling 2010 Organizing Committee.