Is CCGbank Semantically Valid?
Insights from Negation Scope Analysis

Kentaro Kojima,Yoshihide Kato, Shigeki Matsubara

Proceedings of the 39th Pacific Asia Conference on
Language, Information and Computation (PACLIC 39)

Emmanuele Chersoni, Jong-Bok Kim (eds.)

2025

Is CCGbank Semantically Valid? Insights from Negation Scope Analysis

Kentaro Kojima[1], **Yoshihide Kato**[2], **Shigeki Matsubara**[1,2]
[1]Graduate School of Informatics, Nagoya University
[2]Information & Communications, Nagoya University
kojima.kentaro.a0@s.mail.nagoya-u.ac.jp

Abstract

Combinatory Categorial Grammar (CCG) is a grammatical theory that has been widely used in various semantic analyses. To analyze sentences with CCG, it is essential to construct their derivation trees using a CCG parser. Since many of these parsers are typically trained on CCGbank, evaluating the validity of CCGbank is crucial for ensuring the accuracy of the resulting semantic analyses. In this study, we investigate the validity of CCGbank from a semantic perspective, focusing specifically on the negation scope. Our investigation is based on the assumption that if CCGbank is semantically valid, it must correctly capture negation scopes. We conducted experiments comparing the negation scopes derived from CCGbank with those used in a negation scope resolution task, and confirmed that the scope of the quantifier "no" does not align well. The experimental results show that CCGbank does not capture the semantics of quantifiers correctly.

1 Introduction

Combinatory Categorial Grammar (CCG) (Steedman, 2000) is a grammatical theory that establishes a one-to-one correspondence between syntactic and semantic composition. It has been widely used in various semantic analyses (Mineshima et al., 2015; Abzianidze, 2015; Martínez-Gómez et al., 2017; Beschke and Menzel, 2018). To analyze sentences with CCG, it is essential to construct their derivation trees using a CCG parser, and several such parsers have been made publicly available (Clark and Curran, 2004; Lewis et al., 2016; Yoshikawa et al., 2017; Yamaki et al., 2023) . These parsers are typically trained on the CCGbank (Hockenmaier and Steedman, 2007), and consequently, they inherit and reproduce its characteristics.

However, CCGbank has been criticized for being semantically invalid in certain aspects (Boxwell and White, 2008). A key reason for its invalidity is that it is automatically generated from the Penn Treebank (Marcus et al., 1993). To mitigate the issues, some researchers have modified derivation trees to ensure semantic validity, particularly in the interpretation of noun phrases (Honnibal et al., 2010). As an alternative approach, other studies (Hu and Moss, 2018; Hu et al., 2019) have introduced extra semantic rules, which undermine the advantage of CCG having a transparent relation between syntax and semantics.

The aim of this study is to examine CCGbank from a semantic perspective that has not been previously explored: namely negation scope. Our investigation is based on the following assumption:

- If CCGbank is semantically valid, then it must capture negation scopes correctly.

We conducted experiments comparing the negation scopes derived from CCGbank's derivation trees with those used in a negation scope resolution task. The results show that while the scope of "not" generally aligns well, the scope of the quantifier "no" does not. These findings suggest that CCGbank lacks semantic validity in its annotation of the quantifier "no".

The main contributions of this study are summarized as follows:

- We demonstrate that CCGbank is semantically invalid from the negation scope perspective.
- We conduct a linguistic analysis of its causes and find that it lies not in CCG but in the CCGbank.

The remainder of this paper is organized as follows. Section 2 presents the preliminary definitions necessary for understanding this study. Section 3 outlines the methodology for evaluating the validity of CCGbank. Section 4 reports the experiment for evaluating the validity of CCGbank. Section 5 conducts an additional experiment. Finally, Section 6 presents the conclusion.

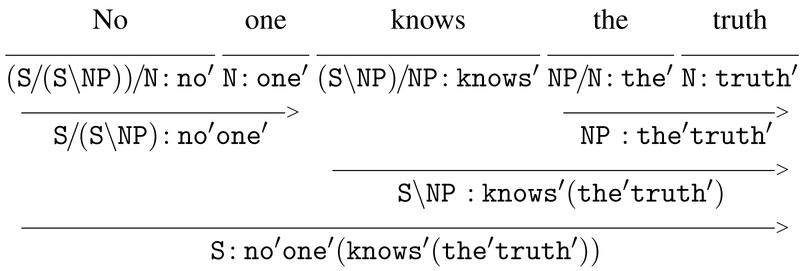

Figure 1: CCG derivation tree of sentence (1)

2 Preliminaries

This section presents preliminary definitions and essential background for understanding this study.

2.1 Negation Scope

Negation is an important phenomenon that frequently appears in natural language. Negation is caused by *negation cue*, such as prefixes (e.g. im-, un-), single words (e.g. not, no), or multiple words (e.g. no more than) and a *negation scope* is a part of a sentence affected by it. For example, in the following sentence (1), "No" functions as the negation cue, whereas "one knows the truth" constitutes the negation scope:

(1) **No** one knows the truth.

Additionally, in the following sentence (2), "not" functions as the negation cue, whereas "I am" and "a student" constitutes the negation scope:

(2) I am **not** a student.

In what follows, negation cues will be indicated in bold, and negation scopes will be marked with underlining. Several datasets annotated with negation cues and scopes have been released, including BioScope (Szarvas et al., 2008) for biomedical texts, SFU Review Corpus (Konstantinova et al., 2012) for product reviews, and the ConanDoyle-neg (Morante and Daelemans, 2012) based on Conan-Doyle's novels. The ConanDoyle-neg is well-known for its ability to capture more complex linguistic phenomena, such as long-range dependencies and discontinuous scope, compared to the other two datasets (Fancellu et al., 2017). Therefore, we adopt the definition of negation scope of the ConanDoyle-neg.

2.2 CCG

CCG is a lexicalized grammatical theory in which each word is assigned a syntactic category. These categories are classified into two types: basic categories (e.g., S for sentence, NP for noun phrase) and complex categories, which are formed by combining categories using the operators / and \. A category of the form X/Y indicates that it expects an expression of category Y to its right in order to form an expression of category X, whereas $X\backslash Y$ expects Y to its left.

In CCG, combining syntactic categories corresponds to one-to-one with that of semantic representations, which can be formalized using lambda calculus. Figure 1 shows the CCG derivation tree for the sentence (1). The syntactic combination illustrated in Figure 1 is based on function application, whose semantic representations are obtained as follows where f and a are λ-terms:

- $X/Y{:}f \quad Y{:}a \Rightarrow X{:}fa$
- $Y{:}a \quad X\backslash Y{:}f \Rightarrow X{:}fa$

CCG also includes another rules[1]:

generalized function composition

- $X/Y{:}f\ Y|_1 Z_1 \cdots |_d Z_d{:}g \Rightarrow X|_1 Z_1 \cdots |_d Z_d{:}\lambda z_d \cdots z_1.f(g z_d \cdots z_1)$
- $Y|_1 Z_1 \cdots |_d Z_d{:}g\ X\backslash Y{:}f \Rightarrow X|_1 Z_1 \cdots |_d Z_d{:}\lambda z_d \cdots z_1.f(g z_d \cdots z_1)$

type raising

- $X{:}a \Rightarrow T/(T\backslash X){:}\lambda f.fa$
- $X{:}a \Rightarrow T\backslash(T/X){:}\lambda f.fa$

Applying the above rules, the CCG derivation tree for sentence (2) is shown in Figure 2.

Here, the notation used in the following sections is defined as follows. For an expression of the form $Y|_1 Z_1 \cdots |_d Z_d$, the sequence $|_1 Z_1 \cdots |_d Z_d$ is referred to as the argument stack and is denoted

[1] Here, $|_i \in \{/, \backslash\}$ and Z_i is a category $(1 \leq i \leq d)$.

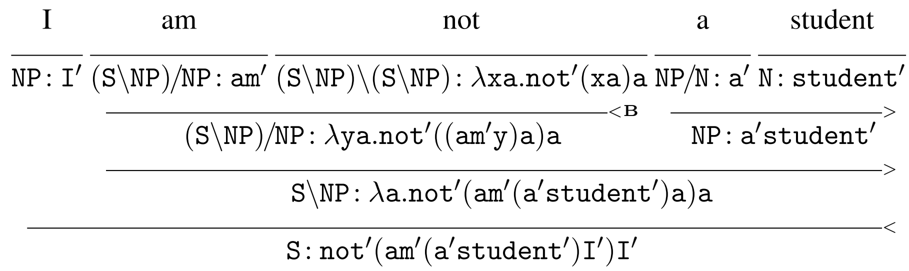

Figure 2: CCG derivation tree of sentence (2)

by a Greek letter, typically α. Specifically, we write $Y|_1 Z_1 \cdots |_d Z_d = Y\alpha$ and define $|\alpha| = d$. When a category X is expressed in the form $X = Y\alpha$, where Y is a basic category, the arity of X is defined as follows:

$$arity(X) = |\alpha|$$

3 Negation Scope on CCGbank

This section outlines the methodology for evaluating the semantic validity of CCGbank. We base our analysis on the following assumption:

- If CCGbank provides semantically valid derivation trees, then the negation scopes derived from them should correspond to those annotated in a general corpus (as described in Section 2.1).

In our analysis, we derive the negation scope from CCGbank as follows[2]:

1. Assign a λ-term to each leaf node in the derivation tree
2. Construct a λ-term according to the CCG rules and obtain its β-normal form M
3. Obtain the negation scope from M

In the following sections, we provide a detailed explanation of 1. and 3., which represent the key components of this evaluation.

3.1 λ-term as Semantic Representation

We do not rely on any specific semantic theory; instead, we use λ-terms that encode only the functional relationships. More specifically, for each word w, a corresponding symbol $\mathtt{w}'$ is introduced and assigned as its semantic representation. By adopting such a primitive semantic representation, it is possible to directly evaluate whether CCGbank correctly captures negation scope without being influenced by any particular semantic theory. In the following sections, we explain special treatment for handling the annotations unique to CCGbank.

3.1.1 Adjunct

Adjuncts are represented by categories of the form $X\alpha|X\alpha$, where X is a basic category. Both instances of $X\alpha$ share the same argument stack α in such cases. For example, in the category (S\NP)\(S\NP), the information about the NP argument expected by the left-hand S\NP must also be passed to the right-hand S\NP. Accordingly, for a word w with the category $(X\alpha|X\alpha)\beta$, the following λ-term is assigned:

$$\lambda b_1 \cdots \lambda b_{|\beta|} \lambda x \lambda a_1 \cdots \lambda a_{|\alpha|}.M$$
$$M \equiv \mathtt{w}' b_1 \cdots b_{|\beta|} (x\ a_1\ \cdots\ a_{|\alpha|}) a_1 \cdots a_{|\alpha|}$$

The λ-terms corresponding to the categories in the argument stack β are sequentially assigned to $b_1, \ldots, b_{|\beta|}$ in the above λ-term. Subsequently, the λ-term associated with the right-hand $X\alpha$ is assigned to x. The variables $a_1, \ldots, a_{|\alpha|}$ are then substituted into this expression, ensuring that the argument stack α on both sides of the adjunction shares the same semantic information.

3.1.2 Coordination

Coordinating conjunctions are represented by a special category called conj, which serves to connect constituents of the same category. It is necessary to distribute the information received by the entire coordinated phrase to each conjunct in coordination structures where the conjuncts are complex categories. For instance, in a coordination structure of verb phrases, the category is S\NP and the conjuncts must receive information about the subject NP. As with adjuncts, this information sharing

[2] McKenna and Steedman (2020) proposes a method that resolves negation scope using CCGbank derivation trees; however, it is not suitable for our purpose because it uses CCG categories as features and does not necessarily conform to the semantic compositionality of CCG.

must be properly represented in the corresponding λ-term. When the category of the coordination structure is X with $arity(X) = n$, the λ-term for the coordinating conjunction is defined as follows:

$$\lambda x_1 \lambda x_2 \lambda y_1 \cdots \lambda y_n.\mathtt{w}'(x_1\, y_1\, \cdots\, y_n)(x_2\, y_1\, \cdots\, y_n)$$

The λ-term corresponding to the right conjunct is assigned to x_1. This term receives the semantic representation passed to the entire coordination structure via the variables $y_1, \ldots, y_n$. Similarly, the variable x_2 is assigned the λ-term corresponding to the left conjunct, which also receives the semantic representation passed to the entire coordination structure via $y_1, \ldots, y_n$.

3.1.3 Type Changing Rule

CCGbank includes type-changing rules that convert a category X into another category Y. We interpret such cases as involving an implicit lexical item with the category Y/X, which applies to the category X via function application to yield the category Y.

3.1.4 Non-local Dependencies

Category information is sometimes shared even outside of adjunct constructions. For example, when the word "which" in the phrase "food which John likes" is assigned the category (NP\NP)/(S\NP), the NP of S\NP and the right-hand NP of NP\NP refer to the same entity (food). To handle such non-local dependencies, we follow the treatment outlined in the CCGbank User's Manual (Hockenmaier and Steedman, 2005) and handle them analogously to adjuncts. Accordingly, we assign the following λ-term to the word "which":

$$\lambda x \lambda y.\mathtt{which}'(xy)y$$

3.2 Negation Scope of CCGbank

Based on the β-normal form λ-term L, the negation scope for the negation cue c is obtained as follows:

- For any subterm of the form fa in L, if $\mathtt{func}(f)$ is the λ-term associated with the negation cue c, and the symbol $\mathtt{w}'$ corresponding to the word w is an element of $\mathtt{Sym}(a)$, then w is included in the scope of c.

func and Sym are defined as follows where M and N are λ-terms:

$$\mathtt{func}(f) = \begin{cases} f & (f \text{ is a symbol}) \\ \mathtt{func}(M) & (f = MN) \\ \text{undefined} & (f = \lambda x.M) \end{cases}$$

	# cues	# sentences
no	618	590
not	4505	4206

Table 1: Statistics of negation cues

	Precision	Recall	F-Score
no	83.35	27.54	41.40
not	84.08	92.50	88.09

Table 2: Results of experiment 1

$$\mathtt{Sym}(a) = \begin{cases} \{a\} & (a \text{ is a symbol}) \\ \mathtt{Sym}(M) \cup \mathtt{Sym}(N) & (a = MN) \\ \mathtt{Sym}(M) & (a = \lambda x.M) \end{cases}$$

4 Experiment 1: Validity of CCGbank

To evaluate the validity of CCGbank from a negation scope perspective, we conducted an experiment using the semantic representation described in Section 3. We used Sections 02-21 of the CCGbank, which are traditionally used as training data.

4.1 Negation Cue

In this experiment, we focus on the negation cues "no" and "not", as they are frequently used in English. Other negation cues, including multi-word constructions such as "by no means" and "no longer", as well as instances where "no" and "not" appear as part of such expressions, are excluded. Table 1 presents the total number of negation cues and sentences.

4.2 Negation Scope

In this experiment, the gold standard negation scopes are provided by NegBERT[3](Khandelwal and Sawant, 2020). This model is selected due to its strong performance, having achieved an F-score of 92.94% on the ConanDoyle-neg. A preliminary experiment confirms that NegBERT demonstrates comparable performance within the CCGbank domain. Specifically, Section 00 of CCGbank is manually annotated for negation scope following the annotation guidelines of the ConanDoyle-neg (Morante et al., 2011), and NegBERT is evaluated against these annotations. The result yields a token-level F-score of 90.90%, supporting the model's suitability for this domain.

[3]https://github.com/adityak6798/Transformers-For-Negation-and-Speculation

negation cue	transformation	Precision	Recall	F-Score
no	None	83.35	27.54	41.40
no	quantifier	71.87	62.06	66.61
no	restrictive post-nominal modification	90.04	49.34	63.75
no	quantifier & restrictive post-nominal modification	77.53	83.80	80.54
not	None	84.08	92.50	88.09
not	quantifier	84.08	92.50	88.09
not	restrictive post-nominal modification	84.97	92.11	88.40
not	quantifier & restrictive post-nominal modification	84.97	92.10	88.39

Table 3: Results for each configuration

4.3 Results of Experiment 1

We evaluated the degree of scope agreement using the token-level F-score. The results are presented in Table 2. As evident from these results, the negation scopes derived from the CCGbank align well with the gold-standard for the negation cue "not", but less so for "no". This means the following:

- The high F-Score with respect to "not" demonstrates that the primitive semantic representation described in Section 3 work well. This is due to the identical treatment of "not" in CCGbank and in CCG.
- Nevertheless, the F-Score with respect to "no" is low. This suggests one possibility: CCGbank does not capture the negation scope of "no".

5 Experiment 2: Linguistic Analysis

Two primary factors may account for the low F-Score for "no". First, the CCGbank does not validly capture the semantics of quantifiers. In Steedman's (2000) analysis, the valid CCG derivation tree for sentence (1) is that shown in Figure 1; however, CCGbank represents it as in Figure 3. Second, in the CCGbank, all post-nominal modifications are uniformly treated as non-restrictive[4] (Hockenmaier and Steedman, 2007; Honnibal et al., 2010). For example, the derivation tree for sentence (3) is shown in Figure 4.

(3) <u>He made</u> **no** <u>remark as to the contents</u>.

These CCGbank's invalid treatment of quantifiers and post-nominal modifications are likely to be the cause of disagreements in negation scope.

[4]In CCG, restrictive and non-restrictive post-nominal modifiers are represented as N\N and NP\NP, respectively.

To investigate the effect of such treatment on CCGbank's negation scope, we modify the CCGbank derivation trees as described in Sections 5.1 and 5.2 and recalculate the F-Score.

5.1 Quantifier

To adopt Steedman's CCG analysis for quantifiers, we transform the category of "no" according to the following:

- When a NP beginning with "no" is an argument of a category of the form $X|_1\texttt{NP}$ (where $X \neq \texttt{NP}$), we replace all occurrences of this NP with the category $X|_2(X|_1\texttt{NP})$ (where $|_2$ denotes the inverse slash of $|_1$).

5.2 Post-nominal Modification

To treat post-nominal modifications as restrictive, we convert CCGbank derivation trees using the conversion rule shown in Figure 5. This rule is identical to the one proposed by Honnibal et al. (2010). Figure 6 shows the modified version of the derivation tree in Figure 4.

5.3 Results of Experiment 2

The experimental results corresponding to each configuration are presented in Table 3. The results show that the negation scope on the modified derivation trees align well with the gold-standards. This indicates the following:

- The current CCGbank (rather than CCG itself) is semantically invalid from the viewpoint of negation scope.

6 Conclusion

This study investigated the validity of CCGbank from the negation scope perspective. Specifically, we compared the negation scopes derived from

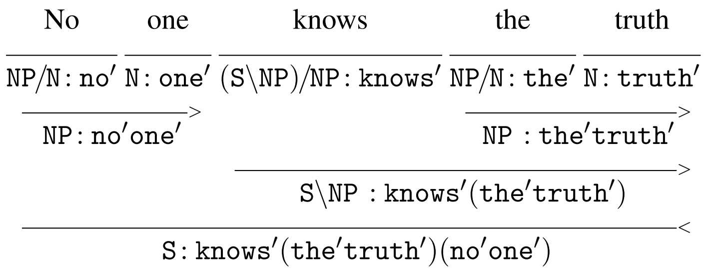

Figure 3: CCGbank derivation tree of sentence (1)

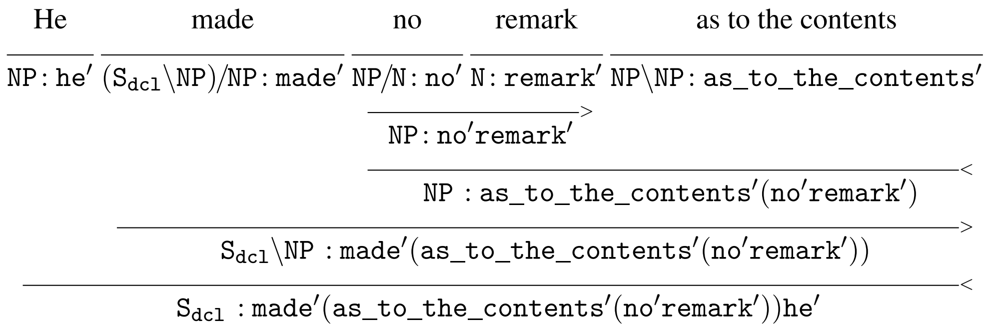

Figure 4: CCGbank derivation tree of sentence (3)

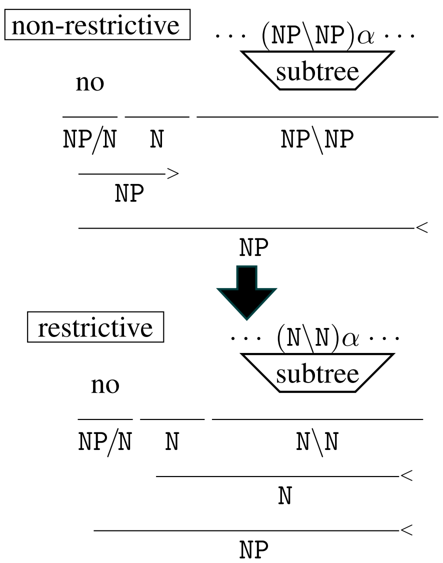

Figure 5: Conversion from non-restrictive to restrictive post-nominal modification.

CCGbank with those defined in the ConanDoyle-neg. The results demonstrated that CCGbank's scopes of the negation cue "not" align well with the ConanDoyle-neg, whereas those of "no" show notable disagreements. We attribute this results to CCGbank's invalid handling of quantifiers and restrictive versus non-restrictive post-nominal modification. However, after modifying CCGbank derivation trees, the negation scopes for "no" align more closely with the gold standards. This finding indicates that there are still challenges in building a semantically valid CCGbank. Therefore, it is essential to address these issues through appropriate corrections on CCGbank to ensure a more valid analysis.

7 Limitation

This study is anchored to NegBERT's performance, given that the gold-standard negation scope was derived from its output. A more rigorous evaluation would therefore require manual annotation of negation scope within CCGbank.

Furthermore, our analysis was limited to the negation cues "no" and "not" and to the phenomena of quantification and restrictive post-nominal modification. Future research should broaden this inquiry by examining additional negation cues and a wider array of linguistic constructions to more comprehensively evaluate the validity of CCGbank.

Acknowledgments

This work was partially supported by the Grant-in-Aid for Scientific Research (B) (No. 25K03418) of JSPS.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of the 2015 Con-*

He | made | no | remark | as to the contents

NP : he′ | (S_dcl\NP)/NP : made′ | (S\NP)\((S\NP)/NP)/N : no′ | N : remark′ | N\N : as_to_the_contents′

(<) N : as_to_the_contents′remark′

(>) (S\NP)\((S\NP)/NP) : no′(as_to_the_contents′remark′)

(<) S_dcl\NP : no′(as_to_the_contents′remark′)made′

(<) S_dcl : no′(as_to_the_contents′remark′)made′he′

Figure 6: Modified version of CCGbank derivation tree of sentence (3)

ference on Empirical Methods in Natural Language Processing, pages 2492–2502.

Sebastian Beschke and Wolfgang Menzel. 2018. Graph algebraic Combinatory Categorial Grammar. In *Proceedings of the 7th Joint Conference on Lexical and Computational Semantics*, pages 54–64.

Stephen Boxwell and Michael White. 2008. Projecting Propbank roles onto the CCGbank. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*.

Stephen Clark and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 103–110.

Federico Fancellu, Adam Lopez, Bonnie Webber, and Hangfeng He. 2017. Detecting negation scope is easy, except when it isn't. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 58–63.

Julia. Hockenmaier and Mark Steedman. 2005. CCGbank: User's Manual.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396.

Matthew Honnibal, James R. Curran, and Johan Bos. 2010. Rebanking CCGbank for improved NP interpretation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 207–215.

Hai Hu, Qi Chen, and Larry Moss. 2019. Natural language inference with monotonicity. In *Proceedings of the 13th International Conference on Computational Semantics*, pages 8–15.

Hai Hu and Larry Moss. 2018. Polarity computations in flexible categorial grammar. In *Proceedings of the 7th Joint Conference on Lexical and Computational Semantics*, pages 124–129.

Aditya Khandelwal and Suraj Sawant. 2020. NegBERT: A transfer learning approach for negation detection and scope resolution. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5739–5748.

Natalia Konstantinova, Sheila C.M. de Sousa, Noa P. Cruz, Manuel J. Maña, Maite Taboada, and Ruslan Mitkov. 2012. A review corpus annotated for negation, speculation and their scope. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3190–3195.

Mike Lewis, Kenton Lee, and Luke Zettlemoyer. 2016. LSTM CCG parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 221–231.

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.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2017. On-demand injection of lexical knowledge for recognising textual entailment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 710–720.

Nick McKenna and Mark Steedman. 2020. Learning negation scope from syntactic structure. In *Proceedings of the 9th Joint Conference on Lexical and Computational Semantics*, pages 137–142.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg: Annotation of negation cues and their scope in conan doyle stories. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 1563–1568.

Roser Morante, Sara Schrauwen, and Walter Daelemans. 2011. Annotation of negation cues and their scope : Guidelines v1.0. Technical report, University of Antwerp.

Mark Steedman. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.

György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. 2008. The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45.

Ryosuke Yamaki, Tadahiro Taniguchi, and Daichi Mochihashi. 2023. Holographic CCG parsing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 262–276.

Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 277–287.