Consistent CCG Parsing over Multiple Sentences for Improved Logical Reasoning

Masashi Yoshikawa[1]
yoshikawa.masashi.yh8@is.naist.jp

Koji Mineshima[2]
mineshima.koji@ocha.ac.jp

Hiroshi Noji[3]
hiroshi.noji@aist.go.jp

Daisuke Bekki[2]
bekki@is.ocha.ac.jp

[1]Nara Institute of Science and Technology, Nara, Japan
[2]Ochanomizu University, Tokyo, Japan
[3]Artificial Intelligence Research Center, AIST, Tokyo, Japan

Abstract

In formal logic-based approaches to Recognizing Textual Entailment (RTE), a Combinatory Categorial Grammar (CCG) parser is used to parse input premises and hypotheses to obtain their logical formulas. Here, it is important that the parser processes the sentences consistently; failing to recognize a similar syntactic structure results in inconsistent predicate argument structures among them, in which case the succeeding theorem proving is doomed to failure. In this work, we present a simple method to extend an existing CCG parser to parse a set of sentences consistently, which is achieved with an inter-sentence modeling with Markov Random Fields (MRF). When combined with existing logic-based systems, our method always shows improvement in the RTE experiments on English and Japanese languages.

1 Introduction

While today's neural network-based syntactic parsers (Dyer et al., 2016; Dozat and Manning, 2017; Yoshikawa et al., 2017) have proven successful on sentence level modeling, it is still challenging to accurately process texts that go beyond a single sentence (e.g. coreference resolution, discourse structure analysis). In this work we focus, among others, on the consistent analysis of multiple sentences in a document. This is as an important problem in reasoning tasks as other document analysis.

RTE is an elemental technology for semantic analysis of multiple sentences, where, given a text (T) and a hypothesis (H), a system determines if T entails H. Existing methods based on formal logic (Bos, 2008; Martínez-Gómez et al., 2017; Abzianidze, 2017) obtain logical formulas for T and H using an off-the-shelf CCG parser, and then feed them to a theorem prover. The standard approach to mapping CCG trees onto logical formulas is to assign λ-terms to the words in a sentence and combine them in a bottom-up fashion (Figure 1a). Here, when the parser fails to make consistent analyses for T and H, the succeeding inference component is also doomed to failure. In Figure 1b, when the parser wrongly analyzes "*man exercising*" in H as "*man*" modifying "*exercising*", the entailment relation cannot be established, due to the different argument structures of `exercise` in the resulting formulas.

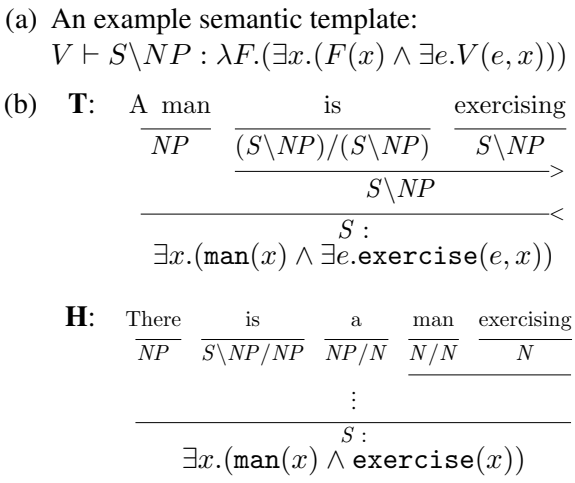

Figure 1: (a) An example semantic template for verbs V that associates a CCG category $S\backslash NP$ with a λ-term. (b) A logical formula of a sentence is obtained at the root of a tree by composing λ-terms of all words following CCG combinatory rules. In this Figure, hypothesis H is wrongly parsed (See the text for details).

While it is ideal to enhance the overall performance of a parser, it is not cheaply obtainable. Additionally, neural network-based parsers are susceptible to subtle changes in the input and thus hard to inspect and modify its parameters to change its prediction. Due to this, we cannot expect that a particular pair of words across multiple sentences be always analyzed in a consistent manner.

In this work, we solve the inconsistency prob-

lem above by adapting the inter-sentence model of Rush et al. (2012) to CCG parsing. Their motivation is to exploit the similarities among test sentences to overcome situations where the amount of the training data is scarce or its domain is different from the test data. The method based on dual decomposition tries to find parse trees for a set of sentences that agree with an MRF, which encourages the assignment of a similar structure to similar contexts.

In our approach, we aim to eliminate wrong logical formulas such as in Figure 1 by rewarding consistent CCG parses across sentences. This, in turn, is achieved by rewarding the consistent assignment of categories to the terminals. This works for CCG parsing, as its derivation is mostly determined by the terminal categories. The key of our approach is that by combining A* parsing of Yoshikawa et al. (2017) with dual decomposition, we can keep small the latency incurred by the use of the iterative algorithm.

We conducted experiments using two state-of-the-art logic-based systems (Martínez-Gómez et al., 2017; Abzianidze, 2017) and two RTE datasets for English and Japanese languages. Our method always shows improvement compared to the baselines.

2 Method

We describe our approach of modeling the inter-consistencies among CCG trees $Y = \langle \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N \rangle$ for sentences $X = \langle \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \rangle$ (§2.1), [1] A* parsing method for each $\boldsymbol{y}_i$ (§2.2) and joint decoding of the MRF and A* parsing using dual decomposition (§2.3).

2.1 Document Consistencies with MRF

To model inter-consistencies among CCG parses, we adapt the global MRF model of Rush et al. (2012). See Figure 2 for an example MRF. Our MRF encourages the assignment of similar categories to the words appearing in similar contexts.

Firstly we construct a graphical representation of an MRF. For each context (unigram surface form in the case of Figure 2) $c \in C$, we have a set W_c of indices $\langle s, t \rangle$ that appear in c, where s is a sentence index and t a word index on sentence s. Let $W = \bigcup_{c \in C} W_c$. We define an undirected graph $G = \langle V, E \rangle$, whose vertices are $V = C \cup W$ and edges $E = \{\langle w, c \rangle : c \in C, w \in W_c\}$. See Figure 2 for an MRF graph constructed for an example RTE problem.

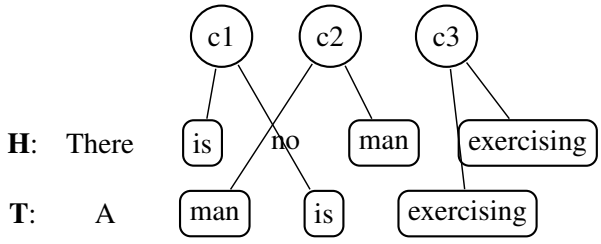

Figure 2: An MRF graph is made up of cliques each consisting of one *context node* ($\in C$; circles) and *word nodes* ($\in W$; rectangles) instantiating that context. As such, each clique expresses the interdependencies among words appearing across sentences.

We assign to each node in the graph a label from a set of CCG categories $\mathcal{T}$, so as to maximize the global consistency score g. By combining g with local CCG parsing for each $\boldsymbol{y}$, we aim to obtain globally consistent trees Y (§2.3). We define label assignment $\boldsymbol{z}$ to nodes in V as $\boldsymbol{z} = \langle z_1, ..., z_{|W|}, z'_1, ..., z'_{|C|} \rangle \in \mathcal{T}^{|W|} \times \mathcal{T}'^{|C|}$, where $\mathcal{T}' = \mathcal{T} \cup \{NULL\}$. In the following, z_w denotes the element in $\boldsymbol{z}$ at the index corresponding to $w \in W$ (similarly z'_c for $c \in C$). Following Rush et al. (2012), we allow $NULL$ label for context nodes. This works as a switch to "turn off" the consistency constraints to the connected nodes. Then, in the set $\mathcal{Z}(X)$ of all possible $\boldsymbol{z}$s for X, we look for $\boldsymbol{z}^* = \arg\max_{\boldsymbol{z} \in \mathcal{Z}(X)} g(\boldsymbol{z})$, where $g(\boldsymbol{z})$ is[2]:

$$g(\boldsymbol{z}) = \sum_{w \in W} f_w(z_w) + \sum_{(w,c) \in E} f_{w,c}(z_w, z'_c).$$

To reward the consistent assignment of categories among connected nodes, $f_{w,c}$ is defined as follow:

$$f_{w,c}(z_w, z'_c) = \begin{cases} \delta_1 & \text{if } z_w = z'_c \\ \delta_2 & \text{if } \texttt{simpl}(z_w) = \texttt{simpl}(z'_c) \\ \delta_3 & \text{if } z'_c = NULL \\ 0 & \text{otherwise,} \end{cases}$$

where $\delta_1 \geq \delta_2 \geq \delta_3$ and `simpl` removes feature values from a category (e.g. $\texttt{simpl}(S_{dcl} \backslash NP) = S \backslash NP$). for f_w, we use $\log P_{tag}$ obtained by CCG parser (§2.2). We tune δ_is based on the RTE performance on the development set.

[1] In this work, we focus on the inconsistency problem of premises and hypotheses of RTE task, and thus X does not contain sentences from any "training data", as was done in Rush et al. (2012). Exploiting external resources in the same manner is also an interesting future direction.

[2] We omit unary terms f_c for $c \in C$, as we set them 0.

Since the above MRF $g(\boldsymbol{z})$ has a simple naïve Bayes structure, we can compute $argmax$ using dynamic programming.

2.2 A* CCG Parsing

To parse a sentence, we use the state-of-the-art A* parsing method of Yoshikawa et al. (2017), which treats a CCG tree $\boldsymbol{y}$ as a tuple $\langle \boldsymbol{c}, \boldsymbol{h} \rangle$ of categories $\boldsymbol{c} = \langle c_1, \ldots, c_M \rangle$ and dependency structure $\boldsymbol{h} = \langle h_1, \ldots, h_M \rangle$, where each h_i is a head index. They model a tree with a locally factored model; the probability of a CCG tree is the product of the probabilities of the categories p_{tag} and the dependency heads p_{dep} of all words in $\boldsymbol{x}$:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i \in [1,M]} p_{tag}(c_i|\boldsymbol{x}) \prod_{i \in [1,M]} p_{dep}(h_i|\boldsymbol{x}).$$

Note that the most computationally heavy part of their method is the calculation of $P_{tag|dep}$, which needs to be done only once in our extension with dual decomposition. The additional computational cost of our method is rather small, as it depends on the number of times to run A* algorithm on the precomputed $P_{tag|dep}$, which is quite efficient.[3]

The probability $P(Y|X)$ of parses Y for X under this model is simply the product of all $\boldsymbol{y}_i$s:

$$\begin{aligned} Y^* &= \arg\max_{Y \in \mathcal{Y}(X)} P(Y|X) \\ &= \arg\max_{Y \in \mathcal{Y}(X)} \sum_{\boldsymbol{y}_i \in Y} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i), \end{aligned}$$

where $\mathcal{Y}(X)$ is the space of all possible parses for X.

2.3 Dual Decomposition

To obtain CCG parses Y for sentences X that are optimal in terms of both the global consistency model (§2.1) and the local parsing model (§2.2), we solve the following problem using dual decomposition:

$$\begin{aligned} (Y^*, \boldsymbol{z}^*) = \arg\max_{Y \in \mathcal{Y}(X), \boldsymbol{z} \in \mathcal{Z}(X)} & P(Y|X) + g(\boldsymbol{z}) \\ \text{s.t. } \forall \langle s, t \rangle \in W \; & z_{s,t} = c_{s,t}, \end{aligned}$$

where $c_{s,t}$ is the category assigned on t'th word in $\boldsymbol{y}_s$. The condition in the equation states that the decoded Y^* and $\boldsymbol{z}^*$ must agree in the category assignment to word nodes in the MRF. Alg. 1 shows the pseudocode for dual decomposition applied to our method. Note that all the decoding subproblems can be kept intact even when added the Lagrangian multiplier u of dual decomposition.

Algorithm 1 Joint CCG parsing and global MRF decoding

▷ J: a set of pairs of word nodes and categories in MRF
▷ α: step size ($0.0 < \alpha \leq 1.0$)
Let $J = \{\langle w, c \rangle | w \in W, c \in \mathcal{T}\}$
Let $\mathbb{1}_c(z) = 1$ if z equals to c else 0
$u^{(1)}_{w,c} \leftarrow 0 \;\; \forall \langle w, c \rangle \in J$
for $k = 1, \ldots, K$ **do**
  $\boldsymbol{z}^{(k)} \leftarrow \arg\max_{\boldsymbol{z} \in \mathcal{Z}(X)} g(\boldsymbol{z}) + \sum_{\langle w,c \rangle \in J} u^{(k)}_{w,c} \mathbb{1}_c(z_w)$
  $Y^{(k)} \leftarrow \arg\max_{Y \in \mathcal{Y}(X)} P(Y|X) - \sum_{\langle w,c \rangle \in J} u^{(k)}_{w,c} \mathbb{1}_c(c_w)$
  if $z^{(k)}_w = c^{(k)}_w$ for all $w \in W$ **then**
    return $\langle \boldsymbol{z}^{(k)}, Y^{(k)} \rangle$
  $u^{(k+1)}_{w,c} \leftarrow u^{(k)}_{w,c} + \alpha(\mathbb{1}_c(z^{(k)}_w) - \mathbb{1}_c(c^{(k)}_w)) \; \forall \langle w, c \rangle \in J$
return $\langle \boldsymbol{z}^{(K)}, Y^{(K)} \rangle$

3 Experiments

3.1 Experimental Settings

English In English experiment, we test the performance of ccg2lambda (Martínez-Gómez et al., 2017) and LangPro (Abzianidze, 2017) on SICK dataset (Marelli et al., 2014)[4]. As mentioned earlier, these systems try to prove whether T entails H, by applying a theorem prover to the logical formulas converted from the CCG trees. We report results for ccg2lambda with the default settings (with SPSA abduction; Martínez-Gómez et al. (2017)) and results for two versions of LangPro, one which is described in Abzianidze (2015) (henceforth we refer to it as LangPro15) and the other in Abzianidze (2017) (LangPro17).[5] Briefly, the difference between the two versions is that LangPro17 is more robust to parse errors. See the paper for the detail. For the CCG parser in §2.2, we use depccg[6] with an MRF in §2.1. We compare our results with depccg without the MRF and baselines reported in the above papers that use

[3] The supertagger of depccg processes 54 sentences per second while its A* decoder 2463 sentences per second. This is measured on SICK test set consisting of 9854 sentences using 2.20 GHz Intel Xeon CPUs with 16 cores.

[4] We also conducted experiments on FraCaS dataset (Cooper et al., 1996). For ccg2lambda, we found no improvements in RTE performance with our MRF, while for LangPro, we found that MRF guides to solve additional two problems.

[5] We report the scores for LangPro improved from the reviewed version, which we obtained from the author through the personal communication after the acceptance.

[6] https://github.com/masashi-y/depccg

Method	Accuracy	Precision	Recall
LangPro15 (Abzianidze, 2015)			
EasyCCG	79.05	**98.00**	52.67
depccg	80.37	97.94	55.81
depccg + MRF	**80.88**	97.91	**57.03**
LangPro17 (Abzianidze, 2017)			
EasyCCG	81.04	97.47	57.69
depccg	81.53	97.51	58.81
depccg + MRF	**81.61**	**97.52**	**59.00**
ccg2lambda (Martínez-Gómez et al., 2017)			
EasyCCG	81.59	**97.73**	58.48
depccg	81.95	97.19	59.98
depccg + MRF	**82.86**	97.14	**62.18**

Table 1: RTE results on test section of SICK

EasyCCG (Lewis and Steedman, 2014).

In MRF, a context node is constructed when two or more words from both T and H share the same surface form. Exceptionally, some pairs of categories are allowed to be aligned with score δ_1: a pair of noun modifier (N/N) and verb tense ($S_{ng}\backslash NP$), which are categories for present participles, and a pair of nominal modifier (N/N) and noun (N). In the experiment using ccg2lambda the pairs of categories of transitive and intransitive verbs, $((S_X\backslash NP)/NP, S_X\backslash NP)$ and $((S_X\backslash NP)/PP, S_X\backslash NP)$, for any feature X are also allowed with δ_1.

For the hyperparamters, we conducted grid search over $[0.0, 0.1, \ldots, 0.9]$ for each δ_i in the MRF s.t. $\delta_1 \geq \delta_2 \geq \delta_3$ and found that $\delta_1 = 0.9, \delta_2 = 0.1, \delta_3 = 0.0$ works the best on SICK trial set. We set $\alpha = 0.0002$ and $K = 500$ in Alg. 1. We decay α by 0.9 in every iteration.

Japanese In Japanese experiment, we evaluate ccg2lambda's performance on JSeM dataset (Kawazoe et al., 2017). To construct an MRF graph, we processed RTE problems with kuromoji[7] and made a context node for a noun or a verb followed by an adverb. The reason why we use bigram POS tag-based context is that the graph construction based on the surface form has resulted in poor RTE performance, by overgenerating MRF constraints. This may be due to the fact that Japanese sentences are usually tokenized into smaller units. We used depccg and the same hyperparameters as English experiment.

[7] http://www.atilika.org/

Method	Accuracy	Precision	Recall
jigg	75.0	92.7	65.4
depccg	67.87	88.34	56.77
depccg + MRF	71.31	88.88	62.24

Table 2: RTE results using ccg2lambda on JSeM

3.2 Results and Error Analysis

We show the results on SICK in Table 1. Our MRF consistently contributes to the improvement of the accuracies for both ccg2lambda and LangPro. We observe the same tendency in the scores for all systems; with MRF, both the accuracy and recall for the systems moderately improve and the systems using depccg have higher recall and lower precision compared to the ones with EasyCCG (with LangPro17 it marks higher precision as well).

In SICK, there are many instances of the construction shown in Figure 1 ("*There is no man exercising*", "*There is no dog barking*", etc.), whose correct reading is that the last verb (e.g. *exercising*) is a present participle modifying a noun (e.g. *man*). EasyCCG and default depccg wrongly parse the last phrase (*man exercising*) as $N/N\ N$, where *man* modifies *exercising*. Our method correctly predicts $N\ S_{ng}\backslash NP$, by utilizing the paired sentence (e.g. "*A man is exercising*"), in which the role of *exercising* is less ambiguous.

Given that the strength of LangPro17 is its robustness to parse errors such as PP-attachment, the larger gain in the accuracy for LangPro15 (roughly 0.5 versus 0.1 point up) indicates that our method is also robust in handling well-known difficult parsing problems. The example (a) in Table 3 is a case of coordinate construction. Baseline depccg wrongly coordinates *crocheting* with a noun *sofa*, while our method successfully resolves the correct coordinate structure by assigning $S_{ng}\backslash NP$ to the word (hence attaching it to *sitting*). Example (b) is one of the cases of PP-attachment that our method successfully resolved. Our method relocates the two PPs in T in their correct places. As in the example in Figure 1, our method corrects cases like (a) and (b) by using the structure of the less ambiguous counterpart as a guide. In the case of (c), the existing parsers misclassify *outdoors* in T as a noun and turns the verb *run* into a transitive verb. With our method, intransitive verb *run* in H works as a soft constraint on the verb in T and corrects its structure successfully. However, there are some cases where using only surface forms as a cue forces the assignment of categories which is

	Sentences
(a)	T: The girl is sitting on the couch and is [$_{S_{ng}\backslash NP}$ crocheting] H: The girl is sitting on the sofa and **crocheting** **crocheting**: ✗ $N \rightsquigarrow$ ✓ $S_{ng}\backslash NP$
(b)	T: A veteran is showing different things **from** a war **to** some people H: Different things [$_{(NP\backslash NP)/NP}$ from] a war are being shown [$_{((S\backslash NP)\backslash(S\backslash NP))/NP}$ to] some people by a veteran **from**: ✗ $((S\backslash NP)\backslash(S\backslash NP))/NP \rightsquigarrow$ ✓ $(NP\backslash NP)/NP$ **to**: ✗ $(NP\backslash NP)/NP \rightsquigarrow$ ✓ $((S\backslash NP)\backslash(S\backslash NP))/NP$
(c)	T: A few man in a competition are [$_{S_{ng}\backslash NP}$ running] outside H: A few man in a competition are **running** outdoors **running**: ✗ $(S_{ng}\backslash NP)/NP \rightsquigarrow$ ✓$S_{ng}\backslash NP$
(d)	T: A man is [$_{(S_{ng}\backslash NP)/NP}$ eating] some food H: The person is **eating** **eating**: ✓ $S_{ng}\backslash NP \rightsquigarrow$ ✗ $(S_{ng}\backslash NP)/NP$

Table 3: Example parse results in SICK test set. (a), (b), (c) With the global MRF model, words in bold font previously assigned a wrong category (✗) have been assigned a correct one (✓). (d) is a case where the MRF is too strict and leads to the wrong assignment.

consistent but not desirable. In example (d), *eat* is used as a transitive verb in T and as an intransitive verb in H; thus it should have different categories.

We show the results on JSeM in Table 2. The RTE performance for Japanese language has improved consistently across all the scores when we add an MRF. However all the scores with depccg (with or without MRF) lag behind the scores reported in Mineshima et al. (2016), which uses a CCG parser implemented in Jigg (Noji and Miyao, 2016). We hypothesize that this is due to the fact that the previous work created the semantic templates for this language by analyzing parse outputs by Jigg and this resulted in a kind of "overfitting" in the templates.

In the above experiments, our method worked well, mainly due to the fact that the sentences in these datasets have comparably simple structure. However, in other datasets, there are naturally more complex cases as in Table 3 (d), where we want different syntactic analyses for occurences of words with the same surface form. We can counter these cases by simply extending the definition of "context" by N-grams or the use of POS tag as we did in the Japanese experiment. Developing a machine learning-based method that selects which contexts to use and set δ_is automatically is also an important future work.

4 Conclusion and Future Work

In this work, by modeling the inter-consistencies of multiple sentences in CCG parsing, we have successfully improved the performance of the formal logic-based methods to RTE. Still, there can be pairs of words in more complex RTE problems that should not have the same category but that our method wrongly force them to. This is mainly due to the fact that we hand-tuned rules to construct context nodes. In future work, we extend the method so that it learns when to set an MRF constraint.

Acknowledgments

First of all, we thank the three anonymous reviewers for their insightful comments. We are also grateful to Lasha Abzianidze for conducting in-depth experiments and for detailed discussion about LangPro. This work was supported by JST CREST Grant Number JPMJCR1301, Japan.

References

Lasha Abzianidze. 2015. A tableau prover for natural logic and language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 2492–2502. http://aclweb.org/anthology/D15-1296.

Lasha Abzianidze. 2017. LangPro: Natural Language Theorem Prover. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Copenhagen, Denmark, pages 115–120. http://www.aclweb.org/anthology/D17-2020.

Johan Bos. 2008. Wide-coverage Semantic Analysis with Boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, STEP '08, pages 277–286.

http://dl.acm.org/citation.cfm?id=1626481.1626503.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. FraCaS: A Framework for Computational Semantics. Deliverable D16.

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. *In Proc. of ICLR* https://arxiv.org/abs/1611.01734.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 199–209. http://www.aclweb.org/anthology/N16-1024.

Ai Kawazoe, Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. 2017. An inference problem set for evaluating semantic theories and semantic processing systems for japanese. In Mihoko Otake, Setsuya Kurahashi, Yuiko Ota, Ken Satoh, and Daisuke Bekki, editors, *New Frontiers in Artificial Intelligence: JSAI-isAI 2015 Workshops, LENLS, JURISIN, AAA, HAT-MASH, TSDAA, ASD-HR, and SKL, Kanagawa, Japan, November 16-18, 2015, Revised Selected Papers*. Springer International Publishing, Cham, pages 58–65. https://doi.org/10.1007/978-3-319-50953-2_5.

Mike Lewis and Mark Steedman. 2014. A* CCG Parsing with a Supertag-factored Model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 990–1000. https://doi.org/10.3115/v1/D14-1107.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, pages 216–223. ACL Anthology Identifier: L14-1314. http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf.

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: Volume 1, Long Papers*. Association for Computational Linguistics, Valencia, Spain, pages 710–720. http://www.aclweb.org/anthology/E17-1067.

Koji Mineshima, Ribeka Tanaka, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2016. Building compositional semantics and higher-order inference system for a wide-coverage Japanese CCG parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 2236–2242. https://aclweb.org/anthology/D16-1242.

Hiroshi Noji and Yusuke Miyao. 2016. Jigg: A Framework for an Easy Natural Language Processing Pipeline. In *Proceedings of ACL-2016 System Demonstrations*. Association for Computational Linguistics, pages 103–108. https://doi.org/10.18653/v1/P16-4018.

Alexander Rush, Roi Reichart, Michael Collins, and Amir Globerson. 2012. Improved Parsing and POS Tagging Using Inter-Sentence Consistency Constraints. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Jeju Island, Korea, pages 1434–1444. http://www.aclweb.org/anthology/D12-1131.

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 (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 277–287. http://aclweb.org/anthology/P17-1026.