

OSU_CHGCG at SemEval-2016 Task 9: Chinese Semantic Dependency Parsing with Generalized Categorical Grammar

Manjuan Duan Lifeng Jin William Schuler

Department of Linguistics

The Ohio State University

{duan, jin, schuler}@ling.osu.edu

Abstract

This paper introduces our Chinese semantic dependency parsing system for Task 9 of SemEval 2016. Our system has two components: a parser trained using the Berkeley Grammar Trainer on the Penn Chinese Treebank reannotated in a Generalized Categorical Grammar, and a multinomial logistic regression classifier. We first parse the data with the automatic parser to obtain predicate-argument dependencies and then we use the classifier to predict the semantic dependency labels for the predicate-argument dependency relations extracted. Although our parser is not trained directly on the task training data, our system yields the best performance for the non-local dependency recovery for the news data and comparable overall results.

1 Introduction

Semantic dependency parsing is an important language processing task which is useful in information extraction and question answering. In this paper, we introduce a Chinese semantic dependency parsing system which is built upon a categorial analysis of the Chinese language.

Categorial grammar annotations are attractive because they have a transparent syntactic-semantic interface and provide a natural account of long-distance dependencies in a language. For this system, we adopt a Generalized Categorical Grammar framework, GCG, (Bach, 1981), for our language

analysis. GCG annotations, compared with other Categorial Grammars, have a larger set of language-specific rules and a smaller set of lexical categories, which on the one hand retains the desirable features of a categorial grammar, such as straightforward compositionality of its syntactic derivations and elegant analysis of filler-gap phenomenon, and on the other hand, mitigates the sparse data problem faced by any heavily lexicalized annotations. Parsers trained with English GCG annotations have been shown to have state-of-the-art parsing performance and better long-distance dependency recovery (Rimell et al., 2009; Nguyen et al., 2012). Parsers trained with Chinese GCG annotations have been shown to achieve better parsing accuracy than the parser trained with Chinese Combinatory Categorial Grammar, CCG, (Steedman, 2000; Steedman, 2012) annotations (Tse and Curran, 2010; Tse and Curran, 2012) for those trees which both grammar annotations assign the same tree structures (Duan and Schuler, 2015).

The current experiment is our first experiment with dependency relations generated from the Chinese GCG annotations. We evaluate them against the manually annotated semantic dependencies in the current SemEval task. Since the purpose of the system is to verify the semantic dependencies generated by the Chinese GCG parser are reasonable, we adopt a minimalist machine learning scheme for this system to accomplish the evaluation. We first train the Berkeley parser (Petrov and Klein, 2007) with GCG annotations converted from the currently

reannotated portion of around 71% of the Chinese Treebank (Xue et al., 2005) sentences. With this parser, we parse the sentences from training sets of the SemEval task and extract dependencies from the parses. Since the dependency labels are more fine-grained in the SemEval task, we train a multinomial logistic regression classifier to predict the dependency labels for the extracted dependencies, using lexical, POS and other position features.

Even though the parser is not directly trained on the gold GCG annotations of the training sentences, this system still yields respectable results compared with other more task dependent systems. Also the official evaluation of the task shows that the current system yields the best non-local dependency parsing accuracy for the newspaper corpus, which supports the findings in English that GCG annotations yield superior performance in long-distance dependency recovery (Nguyen et al., 2012).

2 Chinese GCG Parser

This experiment used the Berkeley parser trained on Chinese GCG reannotated trees.

2.1 Chinese GCG framework

A generalized categorial grammar (Bach, 1981; Nguyen et al., 2012)¹ is a tuple $\langle P, O, R, W, M \rangle$ (Oehrle, 1994) consisting of a set P of primitive category types, a set O of type-constructing operators, a set R of inference rules, a set W of vocabulary items, and a mapping M from vocabulary items to syntactic categories. A set of complex syntactic categories C may then be defined as: $P \subset C$; $C \times O \times C \subset C$; nothing else is in C .

The reannotation of Chinese Treebank into GCG annotations is still an on-going project. So far, we have identified the following set of primitive syntactic categories P for Mandarin Chinese:

- V**: verb-headed clause
- N**: noun-headed phrase or clause
- D**: *de*-clause headed by 的
- C**: cardinal number
- Q**: quantificational phrase
- A**: adjectival phrase or nominal modifier
- R**: adverbial phrase or verbal modifier
- B**: verbal complement of in *ba*-construction
- E**: verbal complement of in passive voice

The set of type-constructing operators O for Mandarin Chinese includes **-a** and **-b** operators for unsatisfied requirements of preceding or succeeding arguments, **-c** and **-d** operators for unsatisfied requirements of preceding or succeeding conjuncts, and a **-g** operator for unsatisfied requirements of gap categories.²

A GCG category consists of a primitive category followed by one or more unsatisfied dependencies, each consisting of an operator followed by another category. For example the category for a transitive verb is ‘**V-aN-bN**’, since it is headed by a verb and has unsatisfied dependencies for noun phrases preceding and following it, i.e., the subject and direct object respectively.

As in other categorial grammars, inference rules for local argument attachment apply functors of category $\varphi_{1..n-1}$ -**ac** or $p\phi_{1..n-1}$ -**bc** to initial or final arguments of category c , where $c \in C$, $p \in P$ and each $\varphi \in \{\mathbf{-a}, \mathbf{-b}\} \times C$:

$$c:g \ p\varphi_{1..n-1}\text{-}\mathbf{ac}:h \Rightarrow p\varphi_{1..n-1}:\lambda_x g(\mathbf{f}_n x) \wedge (h x) \quad (\text{Aa})$$

$$p\varphi_{1..n-1}\text{-}\mathbf{bc}:g \ c:h \Rightarrow p\varphi_{1..n-1}:\lambda_x (g x) \wedge h(\mathbf{f}_n x) \quad (\text{Ab})$$

These two inference rules stipulate the argument of category c is the n -th argument of the head. Inference rules for modifier composition apply preceding or succeeding modifiers of category p -**bd** to modificands of category c , where $p \in \{\mathbf{A}, \mathbf{R}\}$, $d \in \{\mathbf{N}, \mathbf{V}\}$:

$$p\text{-}\mathbf{bd}:g \ c:h \Rightarrow c:\lambda_x \exists_y (g y) \wedge (h x) \wedge (\mathbf{f}_1 y)=x \quad (\text{Ma})$$

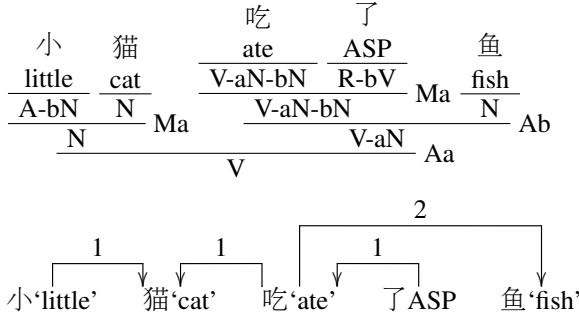
$$c:g \ p\text{-}\mathbf{bd}:h \Rightarrow c:\lambda_x \exists_y (g x) \wedge (h y) \wedge (\mathbf{f}_1 y)=x \quad (\text{Mb})$$

Separate modifier composition rules makes it possible to assign the same syntactic category to the modifier regardless of its occurring position. The modifier composition rules Ma and Mb establish a ‘1’-labeled dependency from the modifier to the modificand. For example, for the sentence ‘小猫吃了’

¹Nguyen et al (2012) notate the ‘//’ and ‘\’ operators of Bach (1981) as **-g** and **-h**, mnemonic for ‘gap’ and ‘heavy shift’.

²Following Nguyen et al (2012), directional operators such as forward and backward slashes (‘\’ and ‘/’) are not used because some operators, such as gap operators in tough constructions, are undirected.

鱼 *The little cat ate the fish*, we have the derivation and the dependencies extracted from the derivation as follows.



Inference rules for gap composition are:

$$p\varphi_{1..n-1}oc \Rightarrow p\varphi_{1..n-1}\text{-}\mathbf{g}c:\lambda_{vx}(gx) \wedge (\mathbf{f}_n x)=v \quad (\text{Ga})$$

$$c:g \Rightarrow c\text{-}\mathbf{g}d:\lambda_{vx}(gx) \wedge (\mathbf{f}_1 v)=x \quad (\text{Gb})$$

$$\mathbf{N}:g \Rightarrow \mathbf{N}\text{-}\mathbf{g}\mathbf{N}:\lambda_{vx}(gx) \wedge \exists_e (\mathbf{de}\text{-}\mathbf{asso} \ e \ x \ v) \quad (\text{Gc})$$

where $p \in P$, $o \in \{-\mathbf{a}, -\mathbf{b}\}$, $c \in C$, $d \in \{\mathbf{A}\text{-}\mathbf{b}\mathbf{N}, \mathbf{R}\text{-}\mathbf{b}\mathbf{V}\}$ and $\varphi \in \{-\mathbf{a}, -\mathbf{b}\} \times C$.

Rule Ga hypothesizes a gap as a preceding or succeeding argument, rule Gb hypothesizes an adjectival or adverbial modifier gap and rule Gc hypothesizes a gap which is associated with the subject in the topicalization construction which does not involve movement.

Non-local arguments, each consisting of a non-local operator and argument category $\psi \in \{-\mathbf{g}\} \times C$, are then propagated to consequents from all possible combinations of antecedents. For $d:g \ e:h \Rightarrow c:(f \ g \ h) \in \{\mathbf{Aa}\text{-}\mathbf{b}, \mathbf{Ma}\text{-}\mathbf{b}\}$:

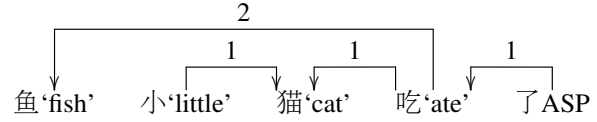
$$\begin{aligned} d\psi_{1..m}:g \ e\psi_{m+1..n}:h \\ \Rightarrow c\psi_{1..n}:\lambda_{v_{1..n}} f(g \ v_{1..m})(h \ v_{m+1..n}) \\ (\text{Ac-d}, \text{Mc-d}) \end{aligned}$$

Rules Ac-d and Mc-d stipulate non-local propagation through argument and modifier composition.

Inference rules for filler attachment apply gapped clauses to topicalized phrases as fillers. For $c \in C$, and $p \in P$:

$$p:g \ c\text{-}\mathbf{g}p:h \Rightarrow c:\lambda_x \exists_y (g \ y) \wedge (h \ y \ x) \quad (\text{Fa})$$

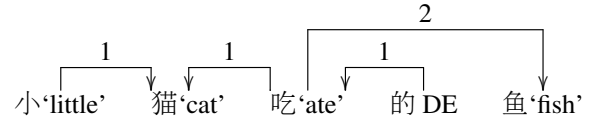
For example, for a topicalized sentence such as “鱼，小猫吃了。 *The fish, the little cat ate.*”, we can extract the dependencies as follows.



In Mandarin relative clauses, the particle 的 ‘de’ takes a preceding clause containing a gap to form a relative clause modifying a succeeding noun. The modified noun is the filler of the gap in the relative clause. The inference rules for relative clauses apply the gapped *de*-clause to the modificand as a filler. For $c \in C$:

$$\mathbf{D}\text{-}\mathbf{g}c:g \ \mathbf{N}:h \Rightarrow \mathbf{N}:\lambda_x (h \ x) \wedge \exists_y (g \ x \ y) \quad (\text{R})$$

For a noun phrase such as “小猫吃的鱼 *the fish that the little cat ate*”, we can extract the following dependencies.



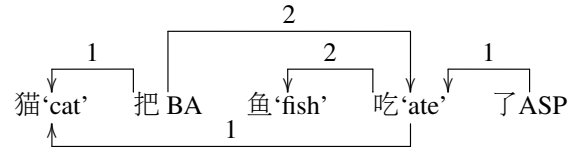
Ba constructions in Mandarin Chinese require the affected patients of certain verbs to occur before the verb. In our analysis, we propose that the particle *ba* takes a *ba*-verb as its complement. *Ba*-verbs can be derived from transitive verbs with the type conversion rule, or some verbs, such as some resultative verbs, have *Ba*-verb categories.

The particle 把 *ba* is assigned the category $\mathbf{V}\text{-}\mathbf{a}\mathbf{N}\text{-}\mathbf{b}(\mathbf{B}\text{-}\mathbf{a}\mathbf{N})$, with coindexation between the referent of its subject ($\mathbf{f}_1 \ x$) and the referent of the subject of its complement ($\mathbf{f}_1 (\mathbf{f}_2 \ x)$).

把 ‘ba’ $\mapsto \mathbf{V}\text{-}\mathbf{a}\mathbf{N}\text{-}\mathbf{b}(\mathbf{B}\text{-}\mathbf{a}\mathbf{N})$:

$$\lambda_x (\mathbf{f}_0 \ x)=\mathbf{ba} \wedge (\mathbf{f}_1 \ x)=(\mathbf{f}_1 (\mathbf{f}_2 \ x))$$

The lexical entry of *ba* gives the following dependencies for the sentence ‘猫把鱼吃了 *The cat ate the fish.*’



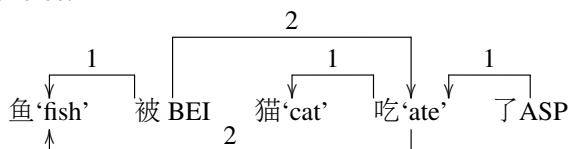
Mandarin Chinese uses the particle 被 *bei* to construct passive sentences. In *bei* constructions, the

patient argument of a verb, usually the second argument of a transitive verb or a *ba*-verb, is moved to the subject position of the clause. We propose the particle 被 *bei* takes a *bei*-verb as its complement, which is converted from a transitive verb with a missing object. Here is the lexical entry we propose for the *bei* particle.

被 ‘bei’ \mapsto V-aN-b(E-aN-gN)-bN:

$$\lambda_x (\mathbf{f}_0 x) = \text{bei} \wedge (\mathbf{f}_3 x) = (\mathbf{f}_1 (\mathbf{f}_2 x))$$

For example, the sentence ‘鱼被猫吃了’ *The fish was eaten by the cat* gives us the following dependencies.



2.2 Training Chinese GCG Parser

The syntactic parser we used for the current semantic dependency parsing is trained by the latent variable PCFG trainer (Petrov and Klein, 2007) on Chinese GCG annotations. Converting Penn Chinese Treebank 5 and 6 into GCG annotations is still an on-going project. We use a set of reannotation rules similar to those described by Nguyen et al. (2012) to reannotate the Penn Chinese Treebank into GCG trees. We currently have fully annotated 71% of sentences (18,505 out of 26,062 sentences) from the Penn Chinese Treebank 5 and 6.³ With these 18,505, we have trained the Chinese GCG parser which is used for the current semantic dependency parsing task. From the parses, we extracted the raw dependencies as the input for the multinomial classifier in the next stage.

3 Multinomial Dependency Label Classifier

As shown in the examples above, the predicate-argument dependencies extracted from GCG derivations do not have fine-grained labels as those dependencies annotated for the task. Also the dependencies identified by the parser sometimes have different directions than those annotated in the task.

³The reannotation rules for generating the training set for current parser are available at <http://www.sourceforge.net/projects/modelblocks>.

Therefore, in order to increase the coverage, for each dependency identified by the parser, we also add a dependency which has the reverse direction. For example, if the parser predicts that a dependency such as *I(eat, cat)*, in which the head is *eat*, the dependent is *cat*, the dependency label is ‘1’, we would add another dependency with inversed direction: *I-inv(cat, eat)*. By doing so, we can increase the coverage of the annotated dependencies to around 83%. However, it also doubles the dependencies predicted by the parser and potentially hurts the recall of the accuracy later on.

There are totally 157 semantic dependency labels used in the task. Since the classifier also needs to decide whether a dependency relation exists between each pair of words, we add a “NoRel” label for those pairs of words which, according to the gold annotation, do not hold any dependency relation between them.

We train a one-vs-all multiclass classifier from the Vowpal Wabbit machine learning package.⁴ We use the following features to predict the dependency labels:

Lexical features: the 300-dimensional word embeddings of the head and dependent words trained with *word2vec* (Mikolov et al., 2013) on the full Chinese Wikipedia, the Chinese Gigaword as well as the training and test datasets in this task;

POS features: the 50-dimensional vector representations of the POS tags of the head and dependent trained with the POS tag sequences from training and test datasets in this task;

Linear distance: the linear distance of the head and the dependent in the sentence;

Path distance: the distance of the nodes of the head and the dependent in the syntactic tree;

Syntactic categories: the GCG syntactic categories of the head and the dependent;

Pred-arg dependency labels: the dependency labels predicted by the parser, such as ‘1’ or ‘2’;

Repetition penalty: the reciprocal of the number of heads that the dependent word has, to penalize proposing too many heads for one word;

Joint features: two-way combinational features between GCG syntactic categories of the two words

⁴https://github.com/JohnLangford/vowpal_wabbit/wiki

	LF	NLF	UF	NUF
News	58.99	54.99	76.64	63.51
Text	65.31	56.74	78.19	66.61

Table 1: Results of development set where LF is the F1 score of the labeled dependency, NLF is the F1 score of the non-local dependency, UF is the F1 score of the unlabeled dependency and NUF is the F1 score the non-local unlabeled dependency.

		LF	NLF	UF	NUF
News	GCG	55.69	49.23	73.72	60.71
	TOP	58.78	40.84	77.64	60.20
Text	GCG	65.17	54.70	78.83	65.71
	TOP	68.59	50.57	82.41	64.58

Table 2: Results of the test set compared with the best system where GCG is the current system and TOP is the system with the best labeled F1 score.

and the dependency label, such “V-aN_1” or “N_1”;

4 Results and Discussion

This Chinese semantic dependency parsing task comes in two domains, the newspaper articles (News) and texts selected from Chinese textbooks (Text). In our experiment, we found the combining the two training sets yields better accuracy for the textbook corpus and a slightly worse performance for the newspaper corpus. Therefore the News results reported in Table 1 and 2 are obtained by a classifier only trained on the newspaper corpus, and the Text results are obtained by a classifier trained on the combined training set of the newspaper corpus and the textbook corpus.

The results in Table 1 show that newspaper text is more difficult to parse, even though the GCG parser is trained on a newspaper corpus. However, it also shows that the parser trained on the newspaper corpus can generalize nicely to another domain such as the textbook corpus, where more diverse syntactic constructions are found.

Table 2 shows the results on the test set compared with the system yielding the highest labeled F1 score. We can see that the current system is around 3 percentage lower than the top system in terms of the labeled F1 score. Considering the fact that the parser is not directly trained on the task-

specific dependency annotations and gold POS tags, these results look reasonably good with the rather simplistic machine learning architecture. Table 2 also shows that the current system achieves the best performance on non-local dependencies according to the official evaluation, which supports the corresponding findings in English where parsers trained on GCG English annotations achieve the state-of-the-art performance in long distance dependency recovery (Nguyen et al., 2012).

5 Error Analysis

We randomly inspected around 20 sentences from each domain where the predictions of the current system are different from the gold annotations to examine the reason, and we identified the following sources of errors.

Parsing errors: Parsing errors caused around half of the wrong predictions we inspected. One type of mistake that we notice the parser often makes is wrong predictions about the internal structure of complex noun phrases. For example, for the noun phrase ‘国际货币基金组织 *International Monetary Fund Organization*’, the parser proposes all first three nouns to be modifiers of the head noun ‘组织 *organization*’, while ‘货币 *monetary*’ actually modifies ‘基金 *fund*’ in gold annotations. The parser also often makes mistakes when parsing questions, since there are not many questions in the Treebank.

Uncovered linguistic phenomena: the gold dependency annotations issued by the task contain dependencies involving co-reference.

- (1) 我 觉得 自己 是 世界上 最幸福
I think myself is in the world happiest
的 人 了。
DE person ASP.
‘I think I am the happiest person in the world.’
- (2) 鲁肃 问 他: “你 叫 我 来 做
Lusu asked him: “you asked me come do
什么? ”
what? ”
‘Lusu asked him: “Why did you ask me to come?”’

In (1), ‘我 *I*’ is annotated to have a *eEqu* relation with ‘自己 *myself*’. In (2), ‘鲁肃 *Lusu*’ has a *eEqu* relation with ‘我 *me*’. Dependencies like these, especially the one in (2), cannot be resolved easily by

a syntactic parser, which means we might need an extra layer of post-processing to do co-reference inference based on discourse information.

Ambiguous constructions: In some cases, a sentence can be analyzed in more than one way. All of them are reasonable analyses but each gives different dependencies.

- (3) 我 爱 他 有 志气。
I love he have aspiration.
'I love he has aspiration.' or 'I love him to have aspiration.'

- (4) 这里 一定 有 人 来 过。
here must have people come TENSE.
'Someone must have come here before'

The current parser parses both (3) and (4) as object control sentences. In (3) '他 *he*' is the object of '爱 *love*' and the subject of '有志气 *have aspiration*'. In the gold annotation of (3), '爱 *love*' takes a sentential complement 'he has aspiration'. Therefore, for (3), our system proposes a dependency between '爱 *love*' and '他 *he*', while in the gold annotations, the dependency is between '爱 *love*' and '有 *have*'. In (4), our parser parses '人 *person*' as the object of '有 *have*' and the subject of '来过 *came*'. '有 *have*', therefore, is the root of the sentence. In gold annotations, '来 *come*' is the root of the sentence.

Inconsistent annotations: There are some cases where our predictions are systematically different from the gold annotations. For example, the current system is consistently different from the gold annotations on the identification of root where some adverbial clauses are involved.

- (5) 万一 明天 下雨, 则 推迟 野游
If tomorrow rain, then postpone trip
日期。
data.
'If it rains tomorrow, postpone the date of trip'
- (6) 既然 她 讨厌 伦敦, 为什么 他 还
Given she dislike London, why he still
在 那里 买 了 房子?
at there buy ASP house?
'Given that she does not like London, why did he buy a house there?'

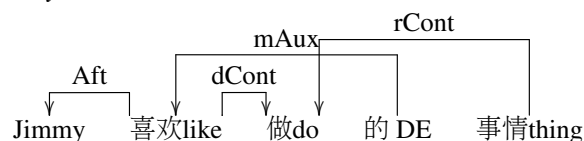
We think both (5) and (6) contain a conditional subordinate clause, and the root of the sentence should

be '推迟 *postpone*' in (5) and '买 *buy*' in (6). In gold annotations, '下雨 *rain*' is annotated to be the root of (5) and '讨厌 *dislike*' the root of (6). Our assumption is that those clauses are handled as conjunctions in gold annotations.

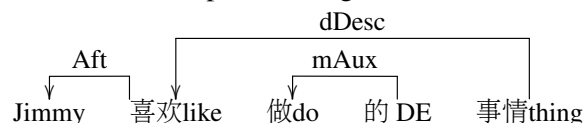
Our predictions also do not agree with the gold annotations in some relative clauses.

- (7) 这 正是 杰里米 喜欢 做 的 事情。
This is Jimmy like do DE thing.
'This is the thing that Jimmy likes to do'

Our system predicts the follow dependencies for the noun phrase 杰里米喜欢做的事情 *things that Jimmy likes to do*.



For the same noun phrase, the gold annotation is



For (7), we think there is a dependency between '做 *do*' and '事情 *thing*', rather than '喜欢 *like*' and '事情 *thing*'.

6 Conclusion

This paper introduces the Chinese semantic dependency parsing system based on the predicate-argument dependencies predicted by a Berkeley parser trained on Chinese GCG trees reannotated from the Penn Chinese Treebank. This system achieves comparable performance for the overall labeled dependency prediction and superior performance for the non-local dependency recovery. Our error analysis shows that many dependency parsing errors can be attributed to the syntactic parsing errors. In the future, we will expand the training set of the parser to cover more diverse syntactic constructions such as questions. We will also consider including corpora from different domains to make the parser more adaptable to data from new domains.

References

- Emmon Bach. 1981. Discontinuous constituents in generalized categorial grammars. *Proceedings of the Annual Meeting of the Northeast Linguistic Society (NELS)*, 11:1–12.
- Manjuan Duan and William Schuler. 2015. Parsing Chinese with a Generative Categorial Grammar. In *Proceedings of Grammar Engineering Across Frameworks*, pages 25–32.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781:1–12.
- Luan Nguyen, Marten van Schijndel, and William Schuler. 2012. Accurate unbounded dependency recovery using generalized categorial grammars. In *Proceedings of COLING ’12*, pages 2125–2140, Mumbai, India.
- Richard T. Oehrle. 1994. Term-labeled categorial type systems. *Linguistics and Philosophy*, 17(6):633–678.
- Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT 2007*, pages 404–411, Rochester, New York, April. Association for Computational Linguistics.
- Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of EMNLP 2009*, volume 2, pages 813–821.
- Mark Steedman. 2000. *The syntactic process*. MIT Press/Bradford Books, Cambridge, MA.
- Mark Steedman. 2012. *Taking Scope - The Natural Semantics of Quantifiers*. MIT Press.
- Daniel Tse and James R. Curran. 2010. Chinese CCG-bank: extracting CCG derivations from the penn chinese treebank. In *Proceedings of COLING ’10*, pages 1083–1091.
- Daniel Tse and James R. Curran. 2012. The Challenges of Parsing Chinese with Combinatory Categorial Grammar. In *Proceedings of NAACL-HLT ’12*, pages 295–304, Montréal, Canada.
- Nianwen Xue, Fei Xian, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese Treebank: Phrase Structure annotation of a large corpus. *Natural Language Engineering*, 11:207–238.