

Semantic Parsing via ℓ_0 -norm-based Alignment

Zhihua Liao

Foreign Studies College
Hunan Normal University
Changsha, China

`liao.zhihua61@gmail.com`

Qixian Zeng

Foreign Studies College
Hunan Normal University
Changsha, China

`wanyouto@qq.com`

Qiyun Wang

Foreign Studies College
Hunan Normal University
Changsha, China

`qywang@qq.com`

Abstract

In this paper, we explore the IBM Model with a ℓ_0 -norm prior to the semantic parsing which parses a sentence to its corresponding meaning representation, and compare two supervised probabilistic Combinatory Categorical Grammar (PCCG) online learning approaches that are Unification-Based Learning (UBL) method and Factored Unification-Based Learning (FUBL) one. Specially, we extend manually GeoQuery and ATIS datasets from English to Chinese *pinyin*-format string. The experiment on such benchmark datasets in both English and Chinese with two different meaning representations (i.e., lambda-calculus and variable-free expressions) demonstrates that both methods adopted this IBM Model with ℓ_0 -norm outperform trivially those that used the IBM Model without ℓ_0 -norm, and also shows small improvements of around 0.1% ~ 0.7% of *F1* for the two algorithms on nearly all conditions.

1 Introduction

Learning the mapping from natural language sentences to formal meaning representations has become one of the main targets in natural language processing. Recent research has focused on learning the semantic parsers directly from corpora that consist of sentences paired with their meaning representations (Artzi and Zettlemoyer, 2011; Artzi and Zettlemoyer, 2013; Kwiatkowski et al., 2010; Kwiatkowski et al., 2011; Lu et al., 2008; Zettlemoyer and Collins, 2005; Zettlemoyer and Collins, 2007; Zettlemoyer and Collins, 2009; Zettlemoyer and Collins, 2012). They usually employ corpus-based probabilistic methods. Furthermore, some research work has been explored to

learn to map any natural language to a wide variety of logical expressions of linguistic meaning (Kwiatkowski et al., 2011; Liao and Zhang, 2013). For example, the training data can consist of Turkish, Spanish, Japanese and English sentences paired with lambda-calculus expressions or variable-free logical ones.

Our approach is inspired by the principle of minimum description length (Barron et al., 1998; Ashish et al., 2012). The main motivation is that through adding a ℓ_0 -norm prior this extension of the IBM model can enable it to encourage the sparsity in word-to-word alignment model. It uses an efficient training algorithm based on projected gradient descent. In this paper, we will apply this method to the semantic parsing. Our work focus on the *Initialization* procedure that the weights for lexeme features are initialized according to cooccurrence statistics between words and logical constants. They are implemented with the modification of GIZA++ toolkit which is viewed as the drop-in replacement for GIZA++ (Ashish et al., 2012).

We evaluate our approach on two benchmark corpora (i.e., GeoQuery and ATIS) annotated with Chinese *pinyin*-format string. The GeoQuery corpus has complex sentence and meaning representation pairs whereas the ATIS corpus contains spontaneous and unedited text so that it is difficult to analyze within formal grammar expression. We compare the performances of both PCCG online learning methods using the IBM Alignment Model with and without ℓ_0 -norm. The experimental results demonstrate the effect of this extended IBM Model with ℓ_0 -norm.

2 Background

We start with a brief review of the IBM word alignment model, then present a detailed description about how to add the ℓ_0 -norm into the baseline IBM Model. Besides, we also review the CCG

grammar (CCG) formalism, the probabilistic CCG (PCCG), and the factored CCG lexicon, as well as the lambda-calculus and higher-order unification.

2.1 IBM Model

Assume that a natural language sentence x is parsed using the CCG lexicon to form a logical expression z . Let a natural language sentence x consist of word-based string $x_1 \dots x_j \dots x_k$, and let the output meaning representation z consist of logical forms $z_1 \dots z_j \dots z_k$. Then this model describes the process by which the meaning representation is generated by the sentence via the alignment $\hat{a} = a_1, \dots, a_j, \dots, a_k$. Each a_j is a hidden variable that indicates which x_{a_j} word the logical form z_j is aligned to.

In IBM model, the joint probability of the sentence and alignment can be defined as follows:

$$P(z, \hat{a}|x) = \prod_{j=1}^m d(a_j|a_{j-1}, j) t(z_j|x_{a_j})$$

Here, the two parameters of this equation are the distortion probability $d(a_j|a_{j-1}, j)$ and the translation probability $t(z_j|x_{a_j})$, respectively.

Let θ stand for all the parameters of this model. The standard training process is to find the parameter values to maximize the likelihood. That is, it is to minimize the negative log-likelihood of the observed data as defined by

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} (-\log P(z|x, \theta)) \\ &= \arg \min_{\theta} (-\log \sum_{\hat{a}} P(z, \hat{a}|x, \theta)) \end{aligned}$$

This can be completed by using the expectation-maximization (EM) algorithm.

2.2 MAP-EM Algorithm with ℓ_0 -norm

In the statistical machine translation field the dominant approach has been the IBM model together with the HMM model. Because it is unsupervised, this can enable it apply to any language pair on an available parallel text. Barron et al. (1998) proposed the principle of minimum description length in the word-to-word translation model, which can reduce the overfitting and result in the garbage collection effect. Then the IBM/HMM model by addition with the ℓ_0 -norm prior to encourage the sparsity has been extended (Ashish et al., 2012). This extension makes use of an efficient training method based on projected gradient descent and

line search to constrained optimization problem. It can scale up to the large dataset in word-to-word alignment. Therefore, this provides significant improvement in the alignment quality.

In word alignment by incorporating a smoothed ℓ_0 prior, the maximum of a posteriori (MAP) objective function is defined as

$$\hat{\theta} = \arg \min_{\theta} (-\log P(z|x, \theta) P(\theta))$$

where

$$P(\theta) \propto \exp(-\alpha \|\theta\|_0^\beta)$$

and

$$\|\theta\|_0^\beta = \sum_{x,z} (1 - \exp \frac{-t(z|x)}{\beta})$$

Here, $P(\theta)$ is a smoothed approximation of the ℓ_0 -norm and the hyperparameter β controls the tightness of approximation.

Next, for an EM procedure the M-step is defined as:

$$\hat{\theta} = \arg \min_{\theta} (-\sum_{x,z} E[C(x,z)] \log t(z|x))$$

Here, the count $C(x,z)$ is the number of times that z occurs aligned to x .

Eventually, MAP-EM is given by:

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} (-\sum_{x,z} E[C(x,z)] \log t(z|x) \\ &\quad - \alpha \sum_{x,z} \exp \frac{-t(z|x)}{\beta}) \end{aligned}$$

This optimization problem is non-convex and can be intractable in a closed-form solution. In order to solve this optimization problem, a projected gradient descent has been employed. Therefore, this extension to IBM model can be implemented as a modification to the open-source toolkit GIZA++¹. Due to its simplicity and generality, this modified model can be utilized to compute cooccurrence statistics in **IBM Model 1** between words and logical constants during the *Initialization* procedure.

¹<http://www.isi.edu/avaswani/giza-pp-10.html>

2.3 Combinatory Categorical Grammars (CCGs)

CCGs are a linguistically-motivated formalism for modeling a wide range of language phenomena (Steedman, 1996; Steedman, 2000). A CCG is defined by a lexicon and a set of combinators. The lexicon contains entries that pair words or phrases with categories like the following (Liao and Zhang, 2013):

alasijia:-NP : alaska:s
 alasijiazhou:-NP : alaska:s
 alasijia:-NP : alaska:n
 alasijiazhou:-NP : alaska:n
 zhijiage:-NP : chicago:c
 zhijiageshi:-NP : chicago:c
 zhijiage:-NP : chicago:n
 zhijiageshi:-NP : chicago:n

Lexical entries share much information while their decompositions can lead to more compact lexicons. When beginning from lexical entries, each intermediate parse node is constructed with one of a small set of CCG combinators. These nodes can capture jointly syntax and semantic information. The combinators contain the functional application, coordination, composition, type-raising and type-shifting.

2.4 Probabilistic CCGs (PCCGs)

It is much obvious for extending CCGs to PCCGs. The primary motivation is to deal with the ambiguity by ranking alternative parses for a sentence in order of probability (Kwiatkowski et al., 2010). Given a CCG lexicon Λ , each sentence may contains many possible parses. The parse with the most likelihood can be selected by using a log-linear model. This model usually consists of a feature vector ϕ and a parameter vector θ . Therefore the joint probability of a logical form z constructed with a parse y , given a sentence x is defined as:

$$P(y, z|x; \theta, \Lambda) = \frac{e^{\theta \cdot \phi(x, y, z)}}{\sum_{(y', z')} e^{\theta \cdot \phi(x, y', z')}}$$

2.5 Factored CCG Lexicon

In general, traditional CCG lexicon lists lexical items that pair words and phrases with syntactic and semantic content. This lexicon might be inefficient when some words appear repeatedly

with closely related lexical content. Recently, Kwiatkowski et al. introduced a factored CCG lexicon representation (Kwiatkowski et al., 2011). Each lexical item is composed of a lexeme and a template such as:

hangban:-N: $\lambda x. flight(x)$
 hangban:-N/(S|NP): $\lambda f \lambda x. flight(x) \wedge f(x)$
 boshidun:-NP:*bos*
 boshidun:-N \ N: $\lambda f \lambda x. from(x, bos) \wedge f(x)$
 piaojia:-N: $\lambda x. cost(x)$
 piaojia:-N/(S|NP): $\lambda f \lambda x. cost(x) \wedge f(x)$
 piaojia:-N \ N: $\lambda f \lambda x. cost(x) \wedge f(x)$
 jiage:-N: $\lambda x. cost(x)$
 jiaqian:-N: $\lambda x. cost(x)$

This factored lexicon includes both of lexeme to model word meaning and template to model systematic variation in word usage. It also allows the reuse of common syntactic structures through a small set of templates. In order to induce a factored lexicon, two procedures are adopted for those factor lexical items into lexemes and templates. Next, these factoring operations are integrated into the complete learning algorithm.

2.6 Lambda Calculus and Higher-Order Unification

Suppose that sentence meaning is represented by use of logical expression. This logical form is defined as the typed lambda-calculus expression (Kwiatkowski et al., 2010). The basic type e stands for an entity, t stands for a truth value, and i for a number. Function types of the form $\langle e, t \rangle$ are assigned to lambda expressions. For example, $\lambda x. state(x)$ take an entity x and return a truth value. The meaning of words and phrases are represented by lambda-calculus forms. They contain constants, quantifiers, logical connectors, and lambda abstractions. Due to its generality, the meaning of each words and phrases can be arbitrary lambda-calculus expressions.

The higher-order unification problem involves finding a substitution for the free variables in a pair of lambda-calculus form which makes the expression equal each other when applied. This problem is remarkable complex and intractable. In the unrestricted case, there can be infinitely many solution pairs (f, g) for a given logical expression h . Instead, the restricted higher-order unification is tractable. For example, given an expression h ,

let find an expression for f and g such that either $h = f(g)$ or $h = \lambda x.f(g(x))$. The limited form of the unification problem can define the ways to split h into subparts so that these subparts can be recombined with CCG parsing operations to reconstruct h .

3 Methodology

This section describes two different PCCG online learning methods, namely, Unification-Based Learning (UBL) method and Factored Unification-Based Learning (FUBL) one.

3.1 UBL Algorithm

This subsection describes the UBL algorithm (Kwiatkowski et al., 2010). This algorithm steps through the data incrementally and performs two-step procedure for each training example. First, new lexical items are induced for the training instance by splitting and merging nodes in the best correct parse given the current parameters. Next, the parameters of the PCCG are updated by computing a stochastic gradient update on the marginal likelihood given the updated lexicon.

3.2 FUBL Algorithm

Although the UBL algorithm can effectively use a higher-order-unification-based lexical induction method to define the space of possible grammars in a language-string and a meaning-representation-independent manner, it can not scale well to some challenging spontaneous and unedited natural language input. At the same time, the FUBL algorithm for inducing factored lexicons is also language independent, but can scale well to these challenging sentences (Kwiatkowski et al., 2011). Assuming training data where each example is a sentence paired with a logical form, the algorithm induces a factored PCCG which includes the lexemes, templates and parameters. This online algorithm repeatedly performs both lexical expansion and a parameter update for each training example. First, the learning algorithm adds lexemes and templates to the factored model by performing manipulations on the highest score pairs of the current training example. Next, a stochastic gradient descent update on the parameter of the parsing model is used to update parameter.

4 Experiments

This section describes our experimental setup and comparisons of the result. We follow the setup of Zettlemoyer and Collins (2005; 2007; 2009; 2012) and Kwiatkowski et al. (2010; 2011) except with manually extending two datasets from English to Chinese *pinyin*-format string, including datasets and initialization as well as system, as reviewed below. Finally, we report the experimental results.

Datasets We evaluate on two benchmark datasets. GeoQuery² is made up of natural language queries to a database of geographical information, while ATIS contains natural language queries to a flight booking system (Deborah et al., 1994). Specially, we have made both of original English corpora (i.e., GeoQuery and ATIS) manually translate into the corresponding Chinese *pinyin*-format string ones by five native quite fluent Chinese speaker, who major in English-Chinese translation during their graduate studying stages. Therefore, Chinese GeoQuery and ATIS corpora are new. Furthermore, GeoQuery contains both lambda-calculus and variable-free meaning representations whereas ATIS only includes lambda-calculus expression. The Geo880 dataset has 880(English sentence or Chinese one, logical form) pairs split into a training set of 600 pairs and a test set of 280 ones. The Geo250 is a subset of the Geo880 and is used 10-fold cross validation experiments with the same splits of the data. Figures 1 and 2 show the examples with both lambda-calculus and variable-free meaning representations in Chinese Geo880 dataset, respectively. The ATIS dataset contains 5410 (English sentence or Chinese one, logical form) pairs split into a 5000 example development set and a 450 example test set. Here, Figure 3 shows some examples with lambda-calculus expression in the Chinese ATIS dataset. Next, we report exact match *Recall*, *Precision* and *F1*. For ATIS we also report partial match *Recall*, *Precision* and *F1*.

neige zhou yv mixiegen jierang

(lambda \$0 e (and (state:t \$0) (next.to:t \$0 michigan:s)))

ehaiezhou jingnei de zhuyao chengshi you neixie

(lambda \$0 e (and (major:t \$0) (city:t \$0) (loc:t \$0 ohio:s)))

akensezhou zuidian shi nali

(argmin \$0 (and (place:t \$0) (loc:t \$0 arkansas:s)) (elevation:i \$0))

²<http://www.cs.utexas.edu/users/ml/geo.html>

neixie zhou yv qiaozhiyaya rjierang
(lambda \$0 e (and (state:t \$0) (next.to:t \$0 georgia:s)))

niuyue you duoshao tiao heliu
(count \$0 (and (river:t \$0) (loc:t \$0 new.york:s)))

Figure 1: Examples with lambda-calculus expression in Chinese Geo880.

neige zhou yv mixiegen jierang
(answer (state (next.to.2 (stateid michigan:e))))

ehaiezhou jingnei de zhuyao chengshi you neixie
(answer (major (city (loc.2 (stateid ohio:e))))))

akensezhou zuididian shi nali
(answer (lowest (place (loc.2 (stateid arkansas:e))))))

neixie zhou yv qiaozhiyaya rjierang
(answer (state (next.to.2 (stateid georgia:e))))

niuyue you duoshao tiao heliu
(answer (count (river (loc.2 (stateid new.york:e))))))

Figure 2: Examples with variable-free expression in Chinese Geo880.

neixie hangban cong dalasi feiwang feinikesi
(lambda \$0 e (and (flight \$0) (from \$0 dallas:ci) (to \$0 phoenix:ci))

neixie hangban cong feinikesi feiwang yanhucheng
(lambda \$0 e (and (flight \$0) (from \$0 phoenix:ci) (to \$0 salt_lake.city:ci))

wo xvyao yitang zaodian de hangban cong mierwoji feiwang danfo
(lambda \$0 e (and (flight \$0) (during.day \$0 early:pd) (from \$0 milwaukee:ci)
(to \$0 denver:ci))

zai danfo you neixie dimian jiaotong leixing kede
(lambda \$v0 e (and (ground.transport \$v0) (to.city \$v0 denver:ci))

Figure 3: Examples with lambda-calculus expression in Chinese ATIS.

Initialization For the fair comparison, we first use the baseline IBM Model without ℓ_0 -norm to the *Initialization* procedure. The weights for lexeme features are initialized according to cooccurrence statistics between words and logical constants. These are estimated with the GIZA++ implementation of IBM Model 1 (Och and Ney, 2003; Och and Ney, 2004). For UBL algorithm, we set the initial weight for each ϕ_L to ten times the average score the (word, constant) pairs in L except for the weights of seed lexical entries in

Λ_{NP} which are set to 10. The learning rate α_0 is set to 1.0 and cooling rate C in all training scenarios set to 10^{-5} and the algorithm is ran for $T = 20$ iterations. For FUBL algorithm, the initial weights for templates are set by adding -0.1 for each slash in the syntactic category and -2 if the template contains logical constants. Features on (lexeme, template) pairs and all parse features are initialized to zero.

Next, we use the modification of IBM model with ℓ_0 -norm to initialize the weights of lexeme features according to cooccurrence statistics between word and logical constants. We have implemented this model as an open-source extension to GIZA++. Usage of the extension is identical to the standard GIZA++. The only differences are that the user needs to switch the ℓ_0 prior on or off, and to adjust both hyperparameters α and β . We first set $\alpha = 10$ and $\beta = 0.05$, then ran five iterations of this Model with the smoothed ℓ_0 -norm. Besides them, the other parameters remain the same as the those of IBM Model without ℓ_0 -norm.

System We employ both supervised PCCG online learning approaches. They include UBL system (Kwiatkowski et al., 2010) and FUBL one (Kwiatkowski et al., 2011). They are implemented after the *Initialization* procedure in which GIZA++ with and without the ℓ_0 -norm is used.

Results Tables 1-7 present the results for all of the experiments. In aggregate, they demonstrate that both UBL and FUBL systems achieve some small improvements for adding the ℓ_0 -norm across languages with lambda-calculus and variable-free expressions. The results that both algorithms are used to test Chinese GeoQuery and ATIS corpora are new. In all cases, FUBL with ℓ_0 -norm performs at or near state-of-the-art recall and precision when compared to those comparable systems.

For the Geo250 domain, Tables 1 and 2 show exact match performances of UBL and FUBL systems with and without ℓ_0 -norm between English and Chinese for both different meaning representations. And the systems with ℓ_0 -norm achieve the best scores. For the Geo880 domain, the results from Tables 3 and 4 indicate that the performances of both systems with ℓ_0 -norm exceed slightly those ones without ℓ_0 -norm.

For the ATIS development set, Table 5 shows the exact match performances of both systems with and without ℓ_0 -norm between English and Chinese with lambda-calculus expression. It can

IBM Model without ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	81.8	83.5	82.6	81.9	86.6	84.1
FUBL	83.7	83.7	83.7	83.9	86.8	85.2

IBM Model with ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	82.0	83.6	82.8	82.4	86.8	84.6
FUBL	83.9	83.8	83.9	84.2	87.0	85.6

Table 1: Exact match performance across languages on Geo250 dataset with lambda-calculus expressions.

IBM Model without ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	80.4	80.8	80.6	78.6	79.3	78.9
FUBL	82.7	83.2	82.9	80.0	81.6	80.8

IBM Model with ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	80.8	81.0	80.9	79.0	79.6	79.3
FUBL	82.8	83.5	83.1	80.4	81.8	81.2

Table 2: Exact match performance across languages on Geo250 dataset with variable-free expressions.

be seen that both algorithms with ℓ_0 -norm also outperforms trivially those without ℓ_0 -norm over 0.2% \sim 0.5%. For the ATIS test set, Tables 6 and 7 present the exact and partial match performances of both systems with and without ℓ_0 -norm. The results demonstrate that the systems with ℓ_0 -norm are superior to the ones without ℓ_0 -norm once again.

5 Conclusion

In this paper, we develop a novel method to the semantic parsing which applies a modified IBM Alignment Model to initialize the weights of all lexical features. During *Initialization* procedure for two PCCG online learning algorithms, because of the addition to ℓ_0 -norm this can enable it to better alignment performances between words and logical expressions. On benchmark datasets in both English and Chinese with two different meaning representations, the experimental results demonstrate that the small improvements have been achieved by the addition of ℓ_0 -norm.

Acknowledgments

We are grateful to the anonymous reviewers for their valuable feedback on an earlier version of this paper. This research was supported in part by Undergraduate Innovative Research Training project of Hunan Normal University (grant no.201401021) and the Social Science Foundation (grant no.14YBA260) in Hunan Province, China.

IBM Model without ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	87.9	88.5	88.2	88.1	90.8	89.4
FUBL	88.6	88.6	88.6	88.8	92.0	91.2

IBM Model with ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	88.2	88.6	88.4	88.5	91.0	89.8
FUBL	88.9	88.9	88.9	89.0	92.1	91.3

Table 3: Exact match performance across languages on Geo880 test set with lambda-calculus expressions.

IBM Model without ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	84.3	85.2	84.7	82.0	84.0	83.0
FUBL	85.7	86.4	86.2	83.8	84.6	84.2

IBM Model with ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	84.5	85.6	85.1	82.3	84.1	83.2
FUBL	86.0	86.5	86.3	84.2	84.6	84.4

Table 4: Exact match performance across languages on Geo880 test set with variable-free expressions.

IBM Model without ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	65.6	67.1	66.3	66.8	69.0	68.4
FUBL	81.9	82.1	82.0	83.3	83.8	83.5

IBM Model with ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	65.8	67.5	66.6	67.0	69.4	68.7
FUBL	82.2	82.6	82.4	83.8	84.0	83.9

Table 5: Exact match performance across languages on ATIS development set with lambda-calculus expressions.

IBM Model without ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	71.4	72.1	71.7	72.6	73.0	72.8
FUBL	82.8	82.8	82.8	83.6	83.5	83.6

IBM Model with ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	72.0	72.6	72.3	72.8	73.4	73.1
FUBL	83.2	83.4	83.3	84.0	84.0	84.0

Table 6: Exact match performance across languages on ATIS test set with lambda-calculus expressions.

IBM Model without ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	78.2	98.2	87.1	79.2	98.6	88.0
FUBL	95.2	93.6	94.6	95.3	93.8	94.6

IBM Model with ℓ_0 -norm	English			Chinese		
	Rec.	Pre.	F1	Rec.	Pre.	F1
UBL	78.6	98.5	87.8	80.0	98.6	88.5
FUBL	95.6	94.0	94.8	95.9	94.4	95.2

Table 7: Partial match performance across languages on ATIS test set with lambda-calculus expressions.

References

- Andrew Barron, Jorma Rissanen, and Bin Yu. 1998. *The Minimum Description Length Principle in Coding and Modeling*. *IEEE Transaction on Information Theory*, 44(6):2743–2760.
- Ashish Vaswani, Liang Huang, and David Chiang. 2012. *Smaller Alignment Models for Better Translations: Unsupervised Word Alignment with the L0-norm*. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–319.
- Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. *Expanding the scope of the ATIS task: the ATIS-3 corpus*. *ARPA Human Language Technology Workshop*, pages 43–48.
- Franz Joseph Och and Hermann Ney. 2003. *A Systematic comparison of Various Statistical Alignment Models*. *Computational Linguistics*, 29(1):19–51.
- Franz Joseph Och and Hermann Ney. 2004. *The Alignment Template Approach to Statistical Machine Translation*. *Computational Linguistics*, 30:417–449.
- Luke S. Zettlemoyer and Michael Collins. 2005. *Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorical Grammars*. In *Proceedings of UAI*, pages 658–666.
- Luke S. Zettlemoyer and Michael Collins. 2007. *Online Learning of Relaxed CCG Grammars for Parsing to Logical Form*. In *Proceedings of EMNLP-CoNLL*, pages 678–687.
- Luke S. Zettlemoyer and Michael Collins. 2009. *Learning Context-Dependent Mappings from Sentences to Logical Form*. In *Proceedings of ACL-IJCNLP*, pages 976–984.
- Luke S. Zettlemoyer and Michael Collins. 2012. *Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorical Grammars*. *CoRR abs*.
- Philipp Koehn, Franz Joseph Och, and Daniel Marcu. 2003. *Statistical Phrase-based Translation*. In *Proceedings of NAACL*.
- Steedman Mark. 1996. *Surface Structure and Interpretation*. The MIT Press.
- Steedman Mark. 2000. *The Syntactic Process*. The MIT Press.
- Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. *Inducing Probabilistic CCG Grammars from Logical Form with Higher-order Unification*. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Cambridge, MA.
- Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. *Lexical Generalization in CCG Grammar Induction for Semantic Parsing*. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Edinburgh, UK.
- Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2012. *A Probabilistic Model of Syntactic and Semantic Acquisition from Child-Directed Utterances and their Meanings*. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, Avignon, France.
- Wei Lu, Hwee Tou Ng, Wee Sun Lee, and Luke S. Zettlemoyer. 2008. *A Generative Model for Parsing Natural Language to Meaning Representations*. In *Proceedings of The Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 783–792.
- Yoav Artzi and Luke Zettlemoyer. 2011. *Bootstrapping Semantic Parsers from Conversations*. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Yoav Artzi and Luke Zettlemoyer. 2013. *Weakly Supervised Learning of Semantic Parsers for Mapping Instructions to Actions*. *Transactions of the Association for Computational Linguistics (TACL)*.
- Zhihua Liao and Zili Zhang. 2013. *Learning to Map Chinese Sentences to Logical Forms*. In *Proceedings of the 7th International Conference on Knowledge Science, Engineering and Management (KSEM)*, pages 463–472.