Enhancing Unsupervised Semantic Parsing with Distributed Contextual Representations

Zixuan Ling[1], **Xiaoqing Zheng**[1,*], **Jianhan Xu**[1],
Jinshu Lin[2], **Kai-Wei Chang**[3], **Cho-Jui Hsieh**[3], **Xuanjing Huang**[1]
[1]School of Computer Science, Fudan University, Shanghai, China [2]Hundsun
[3]Department of Computer Science, University of California, Los Angeles, USA
zxling21@m.fudan.edu.cn, {zhengxq,jianhanxu20}@fudan.edu.cn
linjs13607@hundsun.com, {kwchang,chohsieh}@cs.ucla.edu
xjhuang@fudan.edu.cn

Abstract

We extend a non-parametric Bayesian model of (Titov and Klementiev, 2011) to deal with homonymy and polysemy by leveraging distributed contextual word and phrase representations pre-trained on a large collection of unlabelled texts. Then, unsupervised semantic parsing is performed by decomposing sentences into fragments, clustering the fragments to abstract away syntactic variations of the same meaning, and predicting predicate-argument relations between the fragments. To better model the statistical dependencies between predicates and their arguments, we further conduct a hierarchical Pitman-Yor process. An improved Metropolis-Hastings merge-split sampler is proposed to speed up the mixing and convergence of Markov chains by leveraging pre-trained distributed representations. The experimental results show that the models achieve better accuracy on both question-answering and relation extraction tasks.

1 Introduction

The goal of semantic parsing is to map natural language input into a formal meaning representation (MR), which is one of the long-standing challenges in natural language understanding. Unlike shallow semantic analysis tasks such as relation extraction and semantic role labeling, the output of semantic parsing is complete and unambiguous to the point where it is machine interpretable or even can be executed by a computer program in order to enable various tasks including question answering, reading comprehension, parsing utterances in conversational agents, and translating natural language to database queries (Goldwasser et al., 2011).

Early semantic parsing systems were built using hand-crafted rules (Woods, 1973; Johnson, 1984; Androutsopoulos et al., 1995). After the seminal work of (Zelle and Mooney, 1996), much attention has been given to statistical approaches that can learn models on a corpus of pairs of sentences and their desired outputs (Thompson, 2003; Zettlemoyer and Collins, 2005, 2007; Kwiatkowksi et al., 2010). Both rule-based and statistical approaches require a large amount of labor-intensive annotation. Many methods have been proposed to reduce the number of annotated examples including active learning (Thompson et al., 1999), weak supervision (Berant et al., 2013), using auxiliary information (Krishnamurthy and Mitchell, 2012), supervision from conversations (Artzi and Zettlemoyer, 2011), and learning from user feedback (Iyer et al., 2017). However, writing hand-crafted rules or creating training datasets by manual annotation is still a formidable task so they are hard to scale and only work well in certain domains.

Over the last decade, there has been a rise in end-to-end trainable neural network-based approaches using encoder-decoder frameworks for semantic parsing (Jia and Liang, 2016; Cheng et al., 2017; Dong and Lapata, 2018). Arguably, the biggest disadvantage of these approaches is their "black box" nature—it is hard to know how or why a neural network comes up with a certain output. It is still unclear whether the machine truly "understands" natural language or just uses some tricks and shortcuts to fulfill the tasks (Jia and Liang, 2017). Even though neural network-based approaches greatly reduce the burden of defining lexicons, templates and manually selected features, it is hard for them to model meaning and composition at varying levels of granularity by disentangling higher- and lower-level semantic information and capturing meaning from low-level to high-level via compositionality.

Unsupervised approaches are more widely applicable than supervised ones because they do not require humans to manually annotate training data. The work of (Poon and Domingos, 2009) is the first attempt to learn a semantic parser in an unsupervised way. They use Markov logic networks (Richardson and Domingos, 2006) to model the joint probability of dependency trees and their la-

Findings of the Association for Computational Linguistics: ACL 2023, pages 11454–11465

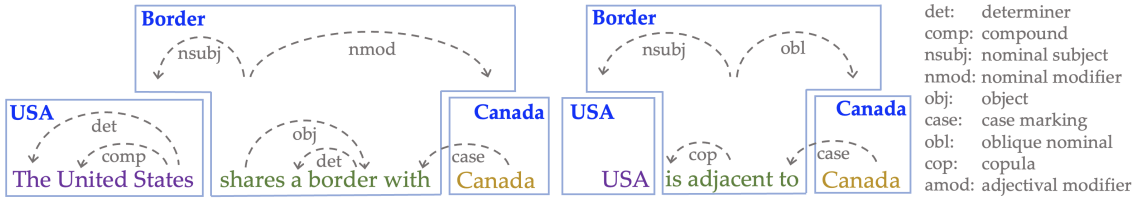

Figure 1: Two example sentences with different syntactic structures but sharing the same meaning representation of `Border(USA,Canada)`. These syntactic structures can be better represented by their contextual embeddings.

tent semantic representations. For each sentence, a Markov network is induced which is an undirected graphical model with nodes representing ground atoms and cliques representing ground clauses. In order for the parameters can be efficiently estimated by a variant of expectation–maximization (EM) algorithm, additional structural constraints were imposed to induce a tree-structured (directed) graph for each sentence. Titov and Klementiev (2011) pointed out that those structural constraints do not fit well with the methodology of Markov logic networks and believe that it is more natural to use a directed model with an underlying generative process specifying how the semantic structure is generated from a dependency parse tree.

Inspired by (Poon and Domingos, 2009), Titov and Klementiev (2011) considered the goal of semantic parsing is to decompose the dependency tree of a sentence into fragments, assign each fragment to a semantically equivalent cluster, and predict predicate-argument relations between the fragments. They use hierarchical Pitman-Yor processes to model dependencies between the meaning representations of predicates and those of their arguments. However, their approach fails to model polysemy while many words in languages are polysemous, carrying multiple related and distinct meanings. As the examples shown in Table 2, their approach cannot discover the words "*windows*" and "*case*" have at least two meanings which seriously degrades the accuracy of semantic parsing, while our proposed algorithm can model such polysemy.

We extend the work of (Titov and Klementiev, 2011) in the following five aspects: (i) the features derived from the contextual word and phrase representations pre-trained on large-scale unlabelled texts are integrated into a non-parametric Bayesian model for unsupervised semantic parsing; (ii) capturing phenomena of homonymy and polysemy by leveraging introduced distributed representations that cannot be modeled before; (iii) phrase-level representations or embeddings are used to better determine whether adjacent words should be composed into a fragment as the smallest semantic unit; (iv) the similarity scores estimated by distributed contextual representations are taken into account in selecting which two semantic classes could be merged into one with priority, which greatly speeds up the mixing and convergence of Markov chains; (v) unlike the situation where only discrete features are used, language semantics can be modeled in a more compact feature space of distributed representations to alleviate the problem of data sparsity. With the above improvements, the enhanced models achieved better performance on both question-answering and relation extraction tasks. The source code of our model can be downloaded from `https://github.com/narcissusLZX/USP`.

2 Method

Similar to (Poon and Domingos, 2009; Titov and Klementiev, 2011), we consider the problem of semantic parsing as a process that seeks to split the words of a sentence into fragments, assign each fragment to a cluster consisting of semantically equivalent expressions, and identify predicate-argument relations between the fragments, given the dependency parse tree of the sentence. As two example sentences shown in Figure 1, we should compose three adjacent words of "*The United States*" to a fragment and assign it to the semantic class "`USA`". The fragments of "*shares a border with*" and "*is adjacent to*" also need to be grouped into a semantic class "`Border`". Therefore, a major challenge in semantic parsing is syntactic variations of the same meaning, which abound in natural languages. By our definition of unsupervised semantic parsing (USP) problem, two main matters need to be addressed. One is to determine whether neighboring words should be composed into fragments, and the other is to cluster fragments into groups based on their similarity in semantic meaning. We demonstrate that pre-trained contextual

word and phrase embeddings are quite useful to better revolve those two matters.

2.1 Semantic Parsing Model

To unsupervisedly induce the semantic representations from the syntactic structures of sentences, we aim to maximize the generation probabilities of the dependency parse trees created for a set of sentences. In order to make the induced meaning representations consistent with each other, the following constraints are imposed on the generation processes of dependency parse trees (an illustrative example is shown in Figure 2).

- Each semantic class c is associated with a distribution ϕ_c that is drawn from a Dirichlet process $\text{DP}(d, H)$ with a base distribution H and a concentration parameter $d > 0$;
- For each semantic class c and each argument type t that is a dependency from the elements in the class (i.e., heads) to modifiers (or dependents), a Pitman-Yor process, denoted as $\theta_{c,t} \sim \text{PY}(\alpha, \beta, G)$, is used to model the distribution of these modifiers where G is a base measure over the syntactic realizations of the modifiers, $0 \leq \alpha < 1$ a discount parameter, and $\beta > -\alpha$ a strength parameter that controls how heavy the tail of the distribution;
- For each semantic class c and each argument type t, a random variable $z_{c,t}$ is used to measure how likely class c has at least one argument of type t, which has a geometric distribution $\text{Geom}(\psi_{c,t})$. The number of additional arguments of the same type t, denoted as $z^+_{c,t}$, is drawn from another geometric distribution of $\text{Geom}(\psi^+_{c,t})$;
- A distribution $\varphi_{c,e} \sim \text{PY}(\alpha, \beta, Q)$ (not shown in Figure 2) is defined over all types of arguments for each pair of classes c and e.

The distribution ϕ_c is used to model the syntactic realizations and their variations for semantic class c. For the predicate of `Border` shown in Figure 1, this distribution should concentrate on syntactic fragments (or lexical items) such as "*shares a border with*", "*is adjacent to*" and "*is bordered by*". The central part of the model is a set of parameters $\theta_{c,t}$, which reflect the preferred selection of certain semantic classes for argument type t of class c. For the arguments of predicate `Border`, these distributions would assign most of their probability mass to semantic classes representing countries or locations. For another example illustrated in Figure 2, the distribution of the arguments for "amod" dependency of the "fruit" class should concentrate on adjectives such as color, quantity, and size. Pitman-Yor processes are considered to be more suitable for modeling the distributions of semantic classes in natural language with power-law tails (Teh, 2006).

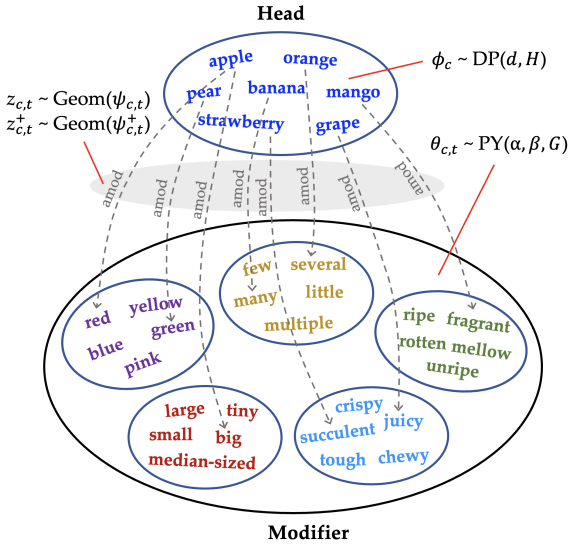

Figure 2: An illustrative example. Each semantic class c (i.e., a set of words referring to some "fruits" here) is associated with a distribution ϕ_c that is drawn from a Dirichlet process $\text{DP}(d, H)$, a Pitman-Yor process $\theta_{c,t} \sim \text{PY}(\alpha, \beta, G)$ is used to model the distribution of the modifiers for the "amod" dependency of the "fruit" class, and parameters $\psi_{c,t}$ and $\psi^+_{c,t}$ are applied to model the number of dependents the "fruit" class could have.

Parameters $\psi_{c,t}$ and $\psi^+_{c,t}$ are used to model how many arguments of type t class c has. For example, a noun could be modified by at least one adjective with a high probability, but the chance of being modified by more than three adjectives is slim. The parameter $\varphi_{c,e}$ defines a distribution over the types of arguments for each pair of classes c and e. For instance, the distribution of the types of arguments between "fruit" class and "color" class should concentrate on "amod". For each pair of semantic classes, a Pitmann-Yor process $\text{PY}(\alpha, \beta, Q)$ is used to model such a distribution. When $\beta = 0$, the Pitman-Yor process reduces to the Dirichlet process. The expected number of components in Pitman-Yor process mixture model scales as αn^β with the number of draws n while it scales logarithmically for Dirichlet processes.

With the distributions described above, we can estimate the generation probability of the dependency parse tree created for a sentence[1]. Starting

[1]In this study, the Stanford (dependency) parser is used to

from the root of the dependency tree, a sentence is generated by recursively drawing a semantic class, the syntactic realization of the class, the number and type of arguments, and the semantic classes for these arguments. Given a set of sentences, we fit the model by maximizing the generation probabilities of all the sentences in the corpus.

2.2 Inference

Pitman-Yor (PY) processes are used to model semantic classes and their arguments in our USP model. A Pitman-Yor process over a set S, denoted by $\text{PY}(\alpha, \beta, G)$, is a stochastic process whose samples are the probability measures on partitions of S. Blackwell and MacQueen (1973) show that the conditional of y_{i+1} given the previous i draws with the probability measures marginalized out follows:

$$y_{i+1}|y_1, \ldots, y_i \sim \sum_{k=1}^{K} \frac{i_k - \beta}{i + \alpha} \delta_{\xi_k} + \frac{K\beta + \alpha}{i + \alpha} G \quad (1)$$

where $\xi_1, \ldots, \xi_K$ are assigned to $y_1, \ldots, y_i$ with K different values (i.e., K different syntactic realizations here). The number of times that ξ_k was assigned is denoted as i_k, and $i = \sum_{k=1}^{K} i_k$.

In the case of conjugate Dirichlet process models (PY processes are the generalization of Dirichlet processes), the Gibbs sampler is the widely-used Markov chain Monte Carlo (MCMC) algorithm. The number of distinct semantic classes is expected to be extremely large for natural languages, and the Gibbs samplers that update the state space one at a time converge very slowly and tend to get stuck in local modes for the problems with large state spaces. Split-merge MCMC algorithms with Metropolis-Hastings (MH) updates (Dahl, 2003; Jain and Neal, 2004) are more efficient than the Gibbs samplers, and can be applied to our model. We consider two moves between states (discuss later) suggested by Titov and Klementiev (2011) to address the above-mentioned two major matters in USP when applying the split-merge MH samplers. The proposed sampling algorithm for unsupervised semantic parsing is given in Algorithm 1.

2.2.1 Metropolis-Hastings Updates

The MH acceptance ratio, denoted as $a(\boldsymbol{\eta}^*|\boldsymbol{\eta})$, is the probability that a proposed state $\boldsymbol{\eta}^*$ is accepted from the current $\boldsymbol{\eta}$. This ratio for the split-merge sampling algorithm is given as follows:

$$a(\boldsymbol{\eta}^*|\boldsymbol{\eta}) = \min\left[1, \frac{p(\boldsymbol{\eta}^*|\boldsymbol{y})}{p(\boldsymbol{\eta}|\boldsymbol{y})} \frac{\pi(\boldsymbol{\eta}|\boldsymbol{\eta}^*)}{\pi(\boldsymbol{\eta}^*|\boldsymbol{\eta})}\right] \quad (2)$$

parse sentences (Manning et al., 2014).

where $\pi(\boldsymbol{\eta}^*|\boldsymbol{\eta})$ is the probability of transiting from state $\boldsymbol{\eta}$ to proposed state $\boldsymbol{\eta}^*$, $p(\boldsymbol{\eta}^*|\boldsymbol{y})$ is the partition posterior distribution evaluated at $\boldsymbol{\eta}^*$, and $\boldsymbol{y}$ is a set of observed data $(y_1, \ldots, y_N)$.

Having proposed a move to $\boldsymbol{\eta}^*$, we determine whether to accept this proposal or not according to the value of $a(\boldsymbol{\eta}^*|\boldsymbol{\eta})$. If the proposal is accepted, the new state is $\boldsymbol{\eta}^*$, otherwise the new state is the same as the current state $\boldsymbol{\eta}$. In this way, we move to the state with a higher probability and repeat the sample until the convergence criterion is met.

2.2.2 Split-Merge Move

In split-merge moves, we decide whether to merge two semantic classes into one or split a class into two. Pre-trained contextual distributed representations are used to choose which two semantic classes should be merged and estimate allocation probabilities of splits. To compute the MH ratio for these moves, only the semantic classes involved in the split and merge operations need to be considered while keeping the rest unchanged. Therefore, such moves can be calculated efficiently.

When the proposal $\boldsymbol{\eta}^*$ is a split move, $\pi(\boldsymbol{\eta}|\boldsymbol{\eta}^*)$ is 1 since these two split classes could only be merged in one way. Similarly, when the proposal $\boldsymbol{\eta}^*$ is a merge update, $\pi(\boldsymbol{\eta}^*|\boldsymbol{\eta}) = 1$. Therefore, we only need to compute $\pi(\boldsymbol{\eta}^*|\boldsymbol{\eta})$ when $\boldsymbol{\eta}^*$ is a split move or $\pi(\boldsymbol{\eta}|\boldsymbol{\eta}^*)$ when it is a merge update.

If a pair of syntactic realizations x_i and x_j randomly selected belong to the same class in $\boldsymbol{\eta}$ (we will discuss how to select them later), we propose $\boldsymbol{\eta}^*$ by attempting a split move. The common class containing x_i and x_j is denoted as S. To compute $\pi(\boldsymbol{\eta}^*|\boldsymbol{\eta})$, we first remove x_i and x_j from S and create two singleton sets $S_i = \{x_i\}$ and $S_j = \{x_j\}$. Letting k be successive values in a uniformly-selected permutation of the indices in S, add x_k to S_i with probability:

$$p(x_k \in S_i|S_i, S_j) = \frac{\sigma(s_k, S_i)}{\sigma(s_k, S_i) + \sigma(s_k, S_j)} \quad (3)$$

where σ is a similarity function whose values are the cosine similarity calculated between the embedding of x_k and the centroid of S_i and then normalized into $[0, 1]$. Note that either S_i or S_j gains a new element at each iteration. After randomly allocating all the elements of S to either S_i or S_j, the split proposal probability $\pi(\boldsymbol{\eta}^*|\boldsymbol{\eta})$ is the product of the allocation probabilities calculated by Equation (3) for each element in S. The merge proposal probability $\pi(\boldsymbol{\eta}|\boldsymbol{\eta}^*)$ can be computed in a similar

way, but which class an element should be allocated is specified in $\boldsymbol{\eta}^*$.

Since the number of semantic classes usually is very large, selecting a pair of x_i and x_j randomly would result in a small proportion of merge moves getting accepted, and lead to a slow-mixing Markov chain. Instead of selecting both of them independently from a uniform distribution, we first choose x_i uniformly, and then randomly select x_j from the distribution based on the cosine similarity of their pre-trained embeddings of x_i and x_j.

2.2.3 Compose-Decompose Move

In compose-decompose moves, we decide whether to compose a pair of head and modifier occurred in some dependency tree into a fragment or decompose it into two. For example, if two randomly-selected fragments have syntactic realizations of "*a*" and "*border*", they would be composed to the fragment "*a* $\xleftarrow{det}$ *border*" that could be further merged with other syntactic structures such as "*share*" and "*with*". Conversely, if two randomly-selected fragments have already been composed, we attempt to split them. After a successful composing or decomposing move, each newly-created fragment will be associated with its distributed representation and assigned to a new semantic class.

The transition probabilities $\pi(\boldsymbol{\eta}|\boldsymbol{\eta}^*)$ of compose-decompose moves are simply estimated based on the number of occurrences of different fragments. For each move, a head-modifier pair will be randomly selected from the distribution based on the number of their occurrences in all the dependency parse trees generated from a text corpus.

2.2.4 Partition Posterior Distribution

In our USP model, the probability of $p(\boldsymbol{\eta}|\boldsymbol{y})$ can be factorized into three parts involving parameters ϕ_c, $\theta_{c,t}$, and $\varphi_{c,e}$ for all the semantic classes affected by proposal $\boldsymbol{\eta}$. Note that for any semantic classes involved, these probabilities need to be computed for two cases: one for them being the role of head, and another for taking the role of modifier (see Figure 2). The probability of $p(\boldsymbol{\eta}|\boldsymbol{y})$ is the product of the probabilities of all parts.

For the first part $\phi_c \sim \mathrm{DP}(d, H)$, the partition prior for a set of syntactic realizations of a semantic class c can be calculated as follows:

$$p(\boldsymbol{\eta}) = d^K \prod_{j=1}^{K} \Gamma(|S_j|) / \prod_{i=1}^{N} (d + i - 1) \tag{4}$$

where $\boldsymbol{\eta} = \{S_1, \ldots, S_K\}$ is a set partition with K different kinds of syntactic realizations, $|S_j|$ is the number of elements in j-th set.

Algorithm 1 A sampling algorithm for USP.

Input: D: A set of unlabelled sentences;
R: A set of pre-trained contextual embeddings;
T: The maximum number of sampling attempts;
E: A desired rejection rate of proposals (e.g., 95%);
L: A similarity threshold for initialization (e.g., 0.8);

Initialization:
Parse the sentence in D and obtain their dependency trees;
Create a set of initial semantic classes and their realizations by assigning the tokens with similarity higher than L to a set.

while the desired rejection rate of proposals E is not achieved **or** the maximum number of sampling T is not reached **do**
  Randomly select which move to be attempted.
  if a merge move is selected **then**
    Randomly choose a pair of semantic classes to merge and propose a merge update $\boldsymbol{\eta}^*$.
  else if a split move is selected **then**
    Randomly select a class to split and propose a split update $\boldsymbol{\eta}^*$
  else randomly select a pair of head and modifier.
    if the selected pair is already composed **then**
      Propose a decomposing update $\boldsymbol{\eta}^*$.
    else Propose a composing update $\boldsymbol{\eta}^*$.
  Compute the MH acceptance ratio a for proposal $\boldsymbol{\eta}^*$ by using Equation (2).
  Generate a random number r between 0 and 1.
  if $r \leq a$ **then** accept $\boldsymbol{\eta}^*$ and move to the new state.
  else reject $\boldsymbol{\eta}^*$ and let the new state be the same as $\boldsymbol{\eta}$.
end

Return: A set of semantic classes and their syntactic realizations as well as a result of semantic parsing for each sentence (i.e., the composed fragments and the predicate-argument relations between them).

For each semantic class c and each argument type t, the partition prior $\theta_{c,t} \sim \mathrm{PY}(\alpha, \beta, G)$ (the second part) is computed as follows:

$$p(\boldsymbol{\eta}) = \beta^K \frac{\Gamma(\frac{\alpha}{\beta} + K)}{\Gamma(\frac{\alpha}{\beta})} \frac{\prod_{j=1}^{K} \frac{\Gamma(|S_j| - \beta)}{\Gamma(1-\beta)}}{\prod_{i=1}^{N} (\alpha + i - 1)} \tag{5}$$

where the definitions of $\boldsymbol{\eta}$, $|S_j|$, and K are similar as Equation (4). For the third part involving parameters $\varphi_{c,e}$, their partition priors also can be calculated using Equation (5), where the elements in sets are argument types rather than the syntactic realizations of semantic classes.

Combining the partition likelihood and the partition prior, Bayes rules give the partition posterior as $p(\boldsymbol{\eta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\eta})p(\boldsymbol{\eta})$, where $p(\boldsymbol{\eta})$ can be computed by Equations (4) and (5). The partition likelihood $p(\boldsymbol{y}|\boldsymbol{\eta})$ is given as a product over components in $\boldsymbol{\eta} = \{S_1, \ldots, S_K\}$ as $\prod_{j=1}^{K} p(\boldsymbol{y}_{S_j})$. Since the observations in each component are fragments with the same syntactic structure, $p(\boldsymbol{y}_{S_j}) = 1$ for all S_j.

To estimate the generation probabilities of dependency parse trees, the probability of the number of arguments that may be provided to a semantic

class also needs to be calculated, which can be viewed as a part of $p(\boldsymbol{y}_{S_j})$. The geometric distribution $\text{Geom}(\psi_{c,t})$ defines the probability of having at least one argument of type t for a given semantic class c, and $\text{Geom}(\psi^+_{c,t})$ models the number of additional arguments of the same type. We denote the number of elements in class c as n, the number of occurrences of argument type t for class c as u, and the number of distinct occurrences as m. The probability of having at least one argument can be calculated by $\text{B}_{m,n-m}(\lambda_0, \lambda_1)$, and that of having an additional argument by $\text{B}_{u-m,m}(\lambda^+_0, \lambda^+_1)$. The function $\text{B}_{x,y}(z_0, z_1)$ can be evaluated as follows:

$$\text{B}_{x,y}(z_0, z_1) = \frac{\Gamma(z_0 + z_1)}{\Gamma(z_0)\Gamma(z_1)} \frac{\Gamma(x + z_0)\Gamma(y + z_1)}{\Gamma(x + z_0 + y + z_1)} \quad (6)$$

3 Experiment

We evaluated the semantic parsing model enhanced by pre-trained contextual embeddings on two tasks of question answering (QA) and relation extraction (RE), comparing to some strong baselines. We also conducted an ablation study to investigate whether contextual embeddings contribute to the problem of homonymy and polysemy and can improve the performance of USP models.

3.1 Evaluation Tasks and Settings

The tasks of question-answering and relation extraction are often used to evaluate semantic parsing models learned in an unsupervised fashion.

Question Answering Following the evaluation setting suggested by Titov and Klementiev (2011), USP models were evaluated on a set of questions and their answers collected by Poon and Domingos (2009) from GENIA corpus (Kim et al., 2003), which consists of $2,000$ biomedical abstracts. All the collected questions are special questions and use "what" at the beginning of the sentence to ask specific questions. For each question, we can obtain the predicate-argument structure of its first word "what" from the semantic parsing results produced by a USP model unsupervisedly trained on $2,000$ abstracts and questions. We then match such a predicate-argument structure against those created for the sentences in the abstracts and extract the matched fragment as the answer.

Relation Extraction Recent research in relation extraction has focused on unsupervised or minimally supervised methods. For the evaluation of this task, we chose to use CASIE dataset (Satyapanich et al., 2020) consisting of $1,000$ English news on cybersecurity. A set of trigger-argument pairs were manually annotated for each news in CASIE dataset, and those triggers can be viewed as predicates in semantic parsing. We collect all the predicate-argument pairs produced by a USP model as the extraction results from the news, and match them against the annotated trigger-argument pairs to calculate the recall and precision.

3.2 Implementation Details

In the implementation of (Titov and Klementiev, 2011), they start with assigning each distinct word (specifically, a word's stem and its part-of-speech tag) into an individual semantic class. Unlike theirs, we first use the distributed contextual representations (also known as embeddings) produced by BERT (Devlin et al., 2018) to generate the feature vector for each word in a sentence and then merge the words with similarity higher than 0.8 into one class for initialization. The cosine similarity is used to measure how similar the words based on their features, which consist of two parts: discrete features and distributed ones. The distributed features are those generated by BERT. For words being split into multiple sub-words or fragments consisting of more than one word, we take the average of their components' embeddings as their distributed feature representations. The discrete feature vector of a word is produced by collecting the number of different dependencies that the word appears as a headword and a modifier (like bag-of-words, but words being replaced by the types of dependencies). The similarity between two words is a weighted sum of the scores calculated based on their discrete and distributed feature vectors. Since it would be better not to choose the values of hyper-parameters for any specific dataset, we simply set the weight to 0.5 when combining the similarity scores estimated using discrete and distributed features.

There would be a large number of distinct feature vectors when distributed contextual representations are used to deal with homonymy and polysemy. To make the computation tractable and speed up the retrieval of similar words or fragments, we used Faiss (Johnson et al., 2019) which is a toolkit for efficient similarity search and clustering of dense vectors. We also applied a well-known algorithm, called Alias (Walker, 1974), for constant-time sampling from a discrete probability distribution.

As shown in Figure 1, for each sampling attempt, we first need to randomly decide which move will be attempted among three options: merge, split, and compose-decompose moves. A merge move will be chosen with 45% probability, a split with 45%, and a compose-decompose with 10% for all the considered tasks. The sampling will continue repeatedly until more than 95% of the proposals were rejected or the maximum number of sampling is reached. The maximum number of sampling was set to $1,500,000$ in all the experiments.

3.3 Results

In Table 1, we report the experimental results of question answering on GENIA corpus and those of relation extraction on CASIE dataset, compared to USP-Bayes (Titov and Klementiev, 2011) from which our model, named USP-DCR, was enhanced in the ability to deal with homonymy and polysemy. For the QA task, we report the number of questions that can be answered by the models, indicated by "Total", the number of questions correctly answered by "Corr", and accuracy by "Accu". For the RE task, precision (indicated by "Prec"), recall, and F1 are reported where F1-score is the harmonic mean of precision and recall.

Model	GENIA			CASIE		
	Total	**Corr**	**Accu**	**Prec**	**Recall**	**F1**
USP-Bayes	**325***	259*	79.7*	37.4	16.9	23.3
USP-DCR	317	**273**	**86.1**	**43.4**	**19.8**	**27.2**
w/o Polysemy	313	256	81.8	40.0	18.0	24.9

Table 1: Results of question answering and relation extraction on GENIA and CASIE datasets respectively. The numbers indicated by the symbol "$*$" were excerpted from the paper of (Titov and Klementiev, 2011).

USP-DCR significantly outperforms USP-Bayes on the question-answering task. Our model can correctly answer more questions than USP-Bayes even though the number of answers returned by theirs is slightly greater than that by ours. USP-Bayes tends to deliver more spurious matches when attempting to answer the questions. USP-DCR performs better than USP-Bayes baseline both in precision and recall on the relations extraction task. The results on GENIA and CASIE datasets demonstrate that both QA and RE tasks can benefit from the introduced contextual distribution representation (CDR) which makes it possible to cluster the fragments that are the same in their appearances but carry distinct meanings into different semantic classes.

3.4 Ablation Study

We conducted an ablation study over GENIA and CASIE datasets to investigate how the performance is impacted if we do not model polysemy. This variant of USP-DCR, indicated by "w/o Polysemy" in Table 1, was trained by assuming that the same syntactic fragments are assigned to the same semantic class (i.e., without polysemous expressions) although the distributed representations are still used to estimate the similarity between two fragments. Note that if the features derived from distributed contextual representations are also not used, our USP-DCR is reduced to USP-Bayes model. The numbers reported in the last row of Table 1 show that the "full-fledged" USP-DCR is superior to its variants, and both contextual embeddings and polysemy modeling are crucial to USP-DCR.

The GENIA corpus is the primary collection of biomedical abstracts, whose texts exhibit a lower degree of polysemy than those from other domains. We extracted a subset of questions from GENIA dataset, which is expected to have a higher degree of polysemy. This subset was constructed by selecting 175 questions that most likely contain polysemous words (the occurrences of these words are far apart in their contextual embedding space). On this subset, USP-DCR achieved 77.4% accuracy and performed better than USP-Bayes by a significant margin of 16.7% improvement in accuracy.

3.5 Case Study

To investigate whether our USP-DCR can truly deal with homonymy and polysemy in the language, we randomly selected two polysemous words and excerpted four related sentences for each word from the datasets used for the evaluation. As shown in Table 2, the first four example sentences were excerpted from CASIE dataset, which all contain the word "windows". In these sentences, that word has two meanings: one is a type of operating system for personal computers, and another is a separate viewing area on a computer display screen. While USP-Bayes is unable to discriminate one meaning from another, the two semantic classes induced by USP-DCR have a clear semantic connection. For example, the first cluster contains nouns used to describe actions or occurrences that can be identified by a program, and all the words in the second cluster are the names of operating systems. The polysemous word "case" and the corresponding sentences were excerpted from GENIA corpus. Again, USP-

Lexicon: windows	
1	Pop-ups are small **windows** that tend to show system warnings which are difficult to close.
2	A user may have multiple **windows** open at a time.
3	And from what I have been finding over the last 6 months, is that the moment you open a brand new laptop with **windows** 10 and start to try to update it, the vulnerability is wide open for attack.
4	In **windows** 7 is almost impossible because those memory address are different in every **windows** installation.
USP-Bayes: {windows, linux, mario, hole}	
USP-DCR:	{**windows**$_{1,2}$, hole, tale, event}, {**windows**$_{3,4}$, Linux, Android, system, macOS}
Lexicon: case	
1	We report an unusual **case** of a 55 year old Japanese woman with a seminoma but relatively normal menses.
2	In each **case**, cytogenetic analysis had either failed or had shown no abnormalities of chromosome 20.
3	In the **case** of thymic selection the mechanism is more subtle depending on the mutual repression of Nur77 and GR.
4	In one **case**, the PTT shift was explained by in-frame splicing out of exon 10, in the presence of a normal exon 10 genomic sequence.
USP-Bayes: {case, study, member, appearance}	
USP-DCR:	{**case**$_{1,2}$, patient, example}, {in the **case**$_3$, **case**$_4$, situation, in the context, in the presence}

Table 2: Example sentences and the corresponding semantic classes (shown below) induced by USP-Bayes and USP-DCR, where the words expressing the same meaning are highlighted in the same color (other than black). The first four sentences were excerpted from the CASIE dataset and the last four from the GENIA corpus. These examples demonstrate that USP-DCR is able to model the polysemy of the words "windows" and "case".

DCR can successfully disambiguate the sense of the word "case" according to its context.

4 Related Work

As one of the major challenges in natural language processing, many methods have been proposed for semantic parsing, which generally can be divided into three categories: rule-based (Woods, 1973; Johnson, 1984; Androutsopoulos et al., 1995), statistical (Zelle and Mooney, 1996; Thompson, 2003; Zettlemoyer and Collins, 2005, 2007; Kwiatkowksi et al., 2010), and neural network-based approaches (Jia and Liang, 2016; Cheng et al., 2017; Dong and Lapata, 2018). Existing approaches differ in the form of meaning representations and the amount of annotation required. In the following, we mainly review prior work on unsupervised statistical methods by which manually labeled training examples are no longer required to build parsing models and refer to two recent surveys (Kamath and Das, 2018; Kumar and Bedathur, 2020) for the other methods.

Poon and Domingos (2009) proposed the first unsupervised approach to semantic parsing which defines a probabilistic model over the dependency tree and semantic parse using Markov logic. Their model recursively clusters and composes the fragments of dependency trees using a hard EM-style procedure. Since they use non-local features and operate over partitions, exact inference is infeasible. They thus resort to a greedy algorithm to find the maximum-a-posteriori parse by searching over partitions. Although it is a powerful model, it is too computationally expensive to run on large corpora. Besides, the methodology of Markov logic networks (innately undirected models) might not be suitable for modeling the semantic structure of a sentence derived from its directed parse tree.

Goldwasser et al. (2011) introduced an unsupervised learning algorithm for semantic parsing, which takes a self-training method driven by confidence estimation. The algorithm iteratively identifies high-confidence self-labeled examples with several simple scoring models and uses the identified samples to re-train the model. To compensate for the absence of direct supervision, Poon (2013) proposed a grounded-learning approach to leverage database schema for indirect supervision. Schmitt et al. (2019) showed that converting a knowledge graph (KG) to its description in natural language (i.e., text generation) and mapping a text back to the KG (i.e, semantic parsing) can be done jointly in an unsupervised manner. Cao et al. (2020) first used an unsupervised paraphrase model to convert natural language utterances into their canonical utterances that were automatically generated by grammar rules and associated with the logic forms, and then trained a semantic parser on a collection of pairs of natural language utterances and the corresponding logic forms in a supervised way. Those approaches are different from our Bayesian model as they rely on either pseudo examples generally without human annotation or external resources such as database schemata or knowledge graphs.

5 Conclusion

We improved the unsupervised learning algorithm proposed by (Titov and Klementiev, 2011) for semantic parsing based on a non-parametric Bayesian

model. Pre-trained contextual word and phrase embeddings were introduced to capture the linguistic phenomena of homonymy and polysemy. Those embeddings and the similarity scores derived from them are also used to determine whether adjacent words can be composed and which semantic classes should be merged during the sequential importance sampling, which can greatly improve computational efficiency. We demonstrate empirically that the semantic parser learned by our approach achieved better performance over the baselines on both question-answering and relation extraction tasks, and show that contextual distributed representations play a vital role in capturing the polysemous variants of words and phrases.

Limitations

This work follows in line with those studies (Poon and Domingos, 2009; Goldwasser et al., 2011; Titov and Klementiev, 2011) where unsupervised semantic parsing relies on the dependency parse trees of texts. Although it enables us to leverage advanced syntactic parsers and to disentangle the complexity in syntactic analysis from that in semantic parsing, the errors made in the dependency parse trees created for input texts could propagate to semantic parsing. In the future, we would like to explore the feasibility of jointly performing syntactic and semantic parsing in a completely unsupervised fashion. Even though an improved MH merge-split sampler was proposed in this study to speed up the mixing and convergence of Markov chains by leveraging pre-trained distributed representations, the computational effort required to fit the model can still be substantial, especially for a large body of texts. We plan to improve computational efficiency beyond that offered by this study by starting with good initialization and updating the state space in a distributed and parallel manner.

Ethics Statement

This work fully complies with the ACL Ethics Policy. All the authors declare that there is no ethical issue in this paper submitted to ACL 2023 for review.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by National Natural Science Foundation of China (No. 62076068), Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0103), and Shanghai Municipal Science and Technology Project (No. 21511102800). Chang is supported in part by Cisco and Sloan fellowship. Hsieh is supported in part by NSF IIS-2008173 and IIS-2048280.

References

Ion Androutsopoulos, Graeme D Ritchie, and Peter Thanisch. 1995. Natural language interfaces to databases–an introduction. *Natural language engineering*, 1(1):29–81.

Yoav Artzi and Luke Zettlemoyer. 2011. Bootstrapping semantic parsers from conversations. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 421–432.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

David Blackwell and James B MacQueen. 1973. Ferguson distributions via pólya urn schemes. *The annals of statistics*, 1(2):353–355.

Ruisheng Cao, Su Zhu, Chenyu Yang, Chen Liu, Rao Ma, Yanbin Zhao, Lu Chen, and Kai Yu. 2020. Unsupervised dual paraphrasing for two-stage semantic parsing. *arXiv preprint arXiv:2005.13485*.

Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. 2017. Learning structured natural language representations for semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

David B Dahl. 2003. An improved merge-split sampler for conjugate dirichlet process mixture models. *Technical R eport*, 1:086.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence driven unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1486–1495.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, Jayant Krishnamurthy, and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Sonia Jain and Radford M Neal. 2004. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of computational and Graphical Statistics*, 13(1):158–182.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Tim Johnson. 1984. Natural language computing: the commercial applications. *The Knowledge Engineering Review*, 1(3):11–23.

Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics (Oxford, England)*, 19 Suppl 1:i180–2.

Jayant Krishnamurthy and Tom Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 754–765.

Pawan Kumar and Srikanta Bedathur. 2020. A survey on semantic parsing from the perspective of compositionality. *arXiv preprint arXiv:2009.14116*.

Tom Kwiatkowksi, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1223–1233.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Hoifung Poon. 2013. Grounded unsupervised semantic parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 933–943.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 1–10.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine learning*, 62(1):107–136.

T. Satyapanich, F. Ferraro, and T. Finin. 2020. Casie: Extracting cybersecurity event information from text. In *AAAI*.

Martin Schmitt, Sahand Sharifzadeh, Volker Tresp, and Hinrich Schütze. 2019. An unsupervised joint system for text generation from knowledge graphs and semantic parsing. *arXiv preprint arXiv:1904.09447*.

Yee Whye Teh. 2006. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 985–992.

Cynthia Thompson. 2003. Acquiring word-meaning mappings for natural language interfaces. *Journal of Artificial Intelligence Research*, 18:1–44.

Cynthia A Thompson, Mary Elaine Califf, and Raymond J Mooney. 1999. Active learning for natural language parsing and information extraction. In *Proceedings of the International Conference on Machine Learning*, pages 406–414. Citeseer.

Ivan Titov and Alexandre Klementiev. 2011. A Bayesian model for unsupervised semantic parsing. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 1445–1455.

A.J. Walker. 1974. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10:127 – 128.

William A Woods. 1973. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the national computer conference and exposition*, pages 441–450.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055.

Luke Zettlemoyer and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687.

Luke S Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
6

☒ A2. Did you discuss any potential risks of your work?
We do not think there are any potential risks of our work.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The license is not required to use the artifacts.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
3

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.