Scalable Semantic Parsing with Partial Ontologies

Eunsol Choi **Tom Kwiatkowski**[†] **Luke Zettlemoyer**
Computer Science & Engineering
University of Washington
eunsol@cs.washington.edu, tomkwiat@google.com, lsz@cs.washington.edu

Abstract

We consider the problem of building scalable semantic parsers for Freebase, and present a new approach for learning to do partial analyses that ground as much of the input text as possible without requiring that all content words be mapped to Freebase concepts. We study this problem on two newly introduced large-scale noun phrase datasets, and present a new semantic parsing model and semi-supervised learning approach for reasoning with partial ontological support. Experiments demonstrate strong performance on two tasks: referring expression resolution and entity attribute extraction. In both cases, the partial analyses allow us to improve precision over strong baselines, while parsing many phrases that would be ignored by existing techniques.

1 Introduction

Recently, significant progress has been made in learning semantic parsers for large knowledge bases (KBs) such as Freebase (FB) (Cai and Yates, 2013; Berant et al., 2013; Kwiatkowski et al., 2013; Reddy et al., 2014). Although these methods can build general purpose meaning representations, they are typically evaluated on question answering tasks and are designed to only parse questions that have complete ontological coverage, in the sense that there exists a logical form that can be executed against Freebase to get the correct answer.[1] In this paper, we instead consider the problem of learning semantic parsers for open domain text containing concepts that may or may not be representable using the Freebase ontology.

Wikipedia	Haitian human rights activists
	Art museums and galleries in New York
	School buildings completed in 1897
	Olympic gymnasts of Norway
Appos.	the capital of quake-hit Sichuan Province
	a major coal producing province
	the relaxed seaside capital of Mozambique

Figure 1: Example noun phrases from Wikipedia category labels and appositives in newswire text.

Even very large knowledge bases have two types of incompleteness that provide challenges for semantic parsing algorithms. They (1) have partial ontologies that cannot represent the meaning of many English phrases and (2) are typically missing many facts. For example, consider the phrases in Figure 1. They include subjective or otherwise unmodeled phrases such as "relaxed" and "quake-hit." Freebase, despite being large-scale, contains a limited set of concepts that cannot represent the meaning of these phrases. They also refer to entities that may be missing key facts. For example, a recent study (West et al., 2014) showed that over 70% of people in FB have no birth place, and 99% have no ethnicity. In our work, we introduce a new semantic parsing approach that explicitly models ontological incompleteness and is robust to missing facts, with the goal of recovering as much of a sentence's meaning as the ontology supports. We argue that this will enable the application of semantic parsers to a range of new tasks, such as information extraction (IE), where phrases rarely have full ontological support and new facts must be added to the KB.

Because existing semantic parsing datasets have been filtered to limit incompleteness, we introduce two new corpora that pair complex noun phrases with one or more entities that they describe. The

[†] Now at Google, NY.

[1] To ensure all questions are answerable, the data is manually filtered. For example, the WebQuestions dataset introduced by Berant et al. (2013) contains only the 7% of the originally gathered questions.

*Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics
and the 7th International Joint Conference on Natural Language Processing*, pages 1311–1320,
Beijing, China, July 26-31, 2015. ©2015 Association for Computational Linguistics

(a) Wikipedia category	x :	Symphonic Poems by Jean Sibelius
	$\mathbf{e}$:	{The Bard, Finlandia,Pohjola's Daughter, En Saga, Spring Song, Tapiola... }
	l_0 :	$\lambda x.Symphonic(x) \wedge Poems(x) \wedge by(JeanSibelius, x)$
	y :	$\lambda\texttt{x}.\texttt{composition.form}(\texttt{x}, \texttt{Symphonicpoems}) \wedge \texttt{composer}(\texttt{JeanSibelius}, \texttt{x})$
	x :	Defunct Korean football clubs
	$\mathbf{e}$:	{ Goyang KB Kookmin Bank FC,Hallelujah FC, Kyungsung FC }
	l_0 :	$\lambda x.defunct(x) \wedge korean(x) \wedge football(x) \wedge clubs(x)$
	y :	$\lambda x.\texttt{OpenType}[\texttt{defunct}](\texttt{x}) \wedge \texttt{OpenRel}(\texttt{x}, \texttt{KOREA}) \wedge \texttt{football\_clubs}(\texttt{x}))$
(b) Appos	x :	a driving force behind the project
	$\mathbf{e}$:	Germany
	l_0 :	$\lambda x.driving(x) \wedge force(x) \wedge behind(x, theproject)$
	y :	$\lambda\texttt{x}.\texttt{OpenType}[\texttt{driving\_force}](\texttt{x}) \wedge \texttt{OpenRel}[\texttt{behind}](\texttt{x}, \texttt{OpenEntity}[\texttt{the\_project}])$
	x :	an EU outpost in the Mediterranean
	$\mathbf{e}$:	Malta
	l_0 :	$\lambda x.outpost(x) \wedge EU(x) \wedge in(x, theMediterranean)$
	y :	$\lambda\texttt{x}.\texttt{OpenRel}(\texttt{x}, \texttt{EU}) \wedge \texttt{OpenType}[\texttt{outpost}](\texttt{x}) \wedge \texttt{contained\_by}(\texttt{x}, \texttt{MediterraneanSea})$

Figure 2: Examples of noun phrases x, from the Wikipedia category and apposition datasets, paired with the set of entities $\mathbf{e}$ they describe, their underspecified logical form l_0, and their final logical form y.

first new dataset contains 365,000 Wikipedia category labels (Figure 1, top), each paired with the list of the associated Wikipedia entity pages. The second has 67,000 noun phrases paired with a single named entity, extracted from the appositive constructions in KBP 2009 newswire text (Figure 1, bottom).[2] This new data is both large scale, and unique in the focus on noun phrases. Noun phrases contain a number of challenging compositional phenomena, including implicit relations and noun-noun modifiers (e.g. see Gerber and Chai (2010)).

To better model text with only partial ontological support, we present a new semantic parser that builds logical forms with concepts from a target ontology and *open* concepts that are introduced when there is no appropriate concept match in the target ontology. Figure 2 shows examples of the meanings that we extract. Only the first of these examples can be fully represented using Freebase, all other examples require explicit modeling of open concepts. To build these logical forms, we follow recent work for Combinatory Categorical Grammar (CCG) semantic parsing with Freebase (Kwiatkowski et al., 2013), extended to model when open concepts should be used. We develop a two-stage learning algorithm: we first compute broad coverage lexical statistics over all of the data, which are then incorporated as features in a full parsing model. The parsing model is tuned on a hand-labeled data set with gold analyses.

Experiments demonstrate the benefits of the new approach. It significantly outperforms strong baselines on both a referring expression resolution task, where much like in the QA setting we directly evaluate if we recover the correct logical form for each input noun phrase, and on entity attribute extraction, where individual facts are extracted from the groundable part of the logical form. We also see that modeling incompleteness significantly boosts precision; we are able to more effectively determine which words should not be mapped to KB concepts. When run on all of the Wikipedia category data, we estimate that the learned model would discover 12 million new facts that could be added to Freebase with 72% precision.

2 Overview

Semantic Parsing with Open Concepts Our goal is to learn to map noun phrase referring expressions x to logical forms y that describe their meaning. In this work, y is built using both concepts from a knowledge base $\mathcal{K}$ and *open concepts* that lie outside of the scope of $\mathcal{K}$. For example, in Figure 2 the phrase "Defunct Korean football clubs" is modeled using a logical form y that contains the $\mathcal{K}$ concept $\texttt{football\_clubs}(\texttt{x})$ as well as the open concepts $\texttt{OpenType}[\texttt{defunct}](\texttt{x})$.

In this paper we describe a new method for learning the mapping from x to y from corpora of referring expression noun phrases, paired with a sets of entities e that these referring expressions describe. Figure 2 shows examples of these data drawn from two sources.

Tasks We introduce two new datasets (Sec. 3) that pair referring noun phrases x with one or more

[2] All new data is available on the authors' websites.

entities **e** that they describe. These data support evaluation for two tasks: referring expression resolution and information extraction.

In referring expression resolution, the parser is given x and is used to predict the referring expression logical form y that describes **e**. Since the majority of our data cannot be fully modeled with Freebase, we evaluate each y against a hand labeled gold standard instead of trying to extract **e** from $\mathcal{K}$.

The entity attribute extraction task also involves mapping phrases x to logical forms y, with the goal of adding new facts to the knowledge base $\mathcal{K}$. To do this, we assume each x is additionally paired with an set of entities **e**. We also define an *entity attribute* to be a literal in y that uses only concepts from $\mathcal{K}$. Finally, we extract, for each entity in **e**, all of the attributes listed in y. For example, the first logical form y in Figure 2 has two entity attributes: $\mathtt{composer}(\mathtt{JeanSibelius}, \mathtt{x})$ and $\mathtt{composition\_form}(\mathtt{x}, \mathtt{Symphonic\_poems})$ which can be added to $\mathcal{K}$ for the entities $\{\mathtt{TheBard}, \mathtt{Finlandia}\}$.

Model and Learning Our approach extends the two-stage semantic parser introduced by Kwiatkowski et al (2013). We use CCG to build domain-independent logical forms l_0 and then introduce a new method for reasoning about how to map this intermediary representation onto both open concepts and $\mathcal{K}$ concepts (Sec. 4).

To learn this model, we assume access to data with two different types of annotations. The first contains noun phrase descriptions x and described entity sets e (as in Figure 2), which can be easily gathered at scale with no manual data labeling effort. However, this data, in general, has significant amount of knowledge base incompleteness; many described concepts and entity attributes will be missing from $\mathcal{K}$ (see Sec. 3 for more details). Therefore, to support effective learning, we will also use a small hand-labeled dataset containing x, **e**, a gold logical form y, an intermediary CCG logical form l_0, and a mapping from words in x to constants in $\mathcal{K}$ and open concepts. Our full learning approach (Sec. 5) estimates a linear model on the small labeled dataset, with broad coverage features derived from the larger dataset.

3 Data

We gathered two new datasets that pair complex noun phrases with one or more Freebase entities.

The Wikipedia category dataset contains 365,504 Wikipedia category names paired with the list of entities in that category. [3] Table 1 shows the details of this dataset and examples are given in Figure 2. For each development and test data, we randomly select 500 categories consisted of 3-10 words and describing fewer than 100 entities.

The apposition dataset is a large set of complex noun phrases paired with named entities, extracted from appositive constructions such as "Gustav Bayer, a former Olympic gymnast for Norway." For this example, we extract the entity "Gustav Bayer" and pair it with the noun phrase "a former Olympic gymnast for Norway." To identify appositive constructions, we ran the Stanford dependency parser on the newswire section of the KBP 2009 source corpus,[4] and selected noun phrases composed of 3 to 10 words, starting with an article, and paired with a named entity that is in Freebase.

This procedure of identifying complex entity descriptions allows for information extraction from a wide range of sources. However, it is also noisy and challenging. The dependency parser makes errors, for example "the next day against the United States, Spain" is falsely detected as an apposition. Furthermore, addressing context and co-reference is often necessary. For example, "Puerto Montt, a city south of the capital" or "the company's parent, Shenhua Group" requires reference resolution. We gathered 67 thousand appositions, which will be released to support future work, and randomly selected 300 for testing.

Measuring Incompleteness To study the amount of incompleteness in this data, we hand labeled logical forms for 500 Wikipedia categories in the development set. Examples of annotations are given in the rows labeled y in Figure 2. We use these to measure the schema and fact coverage of Freebase. Many of the entities in this dataset do not have the Freebase attributes described by the category phrases. When a concept is not in Freebase, we annotate it as OpenType or OpenRel, as shown in Figure 2. On average, each Wikipedia category name describes 2.58 Freebase attributes, and 0.39 concepts that cannot be mapped to FB. Overall, 27.2% of the phrases contain concepts that do not exist in the Freebase schema.

[3] Compiled by the YAGO project, available at: www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/

[4] http://www.nist.gov/tac/2009/

	entire set	dev	test
# categories	365,504	500	500
# words per category	4.1	4.4	4.3
# unique words	84,996	1,100	1,063
# entities per category	19.9	19.1	18.7
# entities	2,813,631	9,511	9,281
# entity-category pairs	7,292,326	9,549	9,331

Table 1: Wikipedia category data statistics.

	entire set	test set
# appositions	66,924	300
# unique words	25,472	817
# words per apposition	5.73	5.93

Table 2: Appositive data statistics.

Each category may have multiple correct logical forms. For example, "Hotels" can be mapped to: $\texttt{hotel}(\texttt{x})$, $\texttt{accomodation.type}(\texttt{x}, \texttt{hotel})$, or $\texttt{building\_function}(\texttt{x}, \texttt{hotel})$. There are also genuine ambiguities in meaning. For example, "People from Bordeaux" can be interpreted as $\texttt{people}(\texttt{x}) \wedge \texttt{place\_lived}(\texttt{x}, \texttt{Bordeaux})$ or $\texttt{people}(\texttt{x}) \wedge \texttt{place\_of\_birth}(\texttt{x}, \texttt{Bordeaux})$. We made a best effort attempt to gather as many correct logical forms as possible, finding on average 1.8 logical forms per noun phrase. There were 97 unique binary relations, and 247 unique unary attributes in the annotation.

Given these logical forms, we also measured factual coverage. For the 72.8% of phrases that can be completely represented using Freebase, we executed the logical forms and compared the result to the labeled entity set. In total, 56% of the queries returned no entities and those that did return results have on average 15% overlap with the Wikipedia entity set. We also measured how often attributes from the labeled logical forms were assigned to the Wikipedia entities in FB, finding that only 33.6% were present. Given this rate, we estimate that it is possible to add 12 million new facts into FB from the 7 million entity-category pairs.

4 Mapping Text to Meaning

We adopt a two-stage semantic parsing approach (Kwiatkowski et al., 2013). We first use a CCG parser to define a set $\textsc{ccg}(x)$ of possible logical forms l_0. Then we will choose the logical form l_0 that closely matches the linguistic structure of the input text x, according to a learned linear model, and use an ontological match step that defines a set of transformations $\textsc{ont}(l_0, \mathcal{K})$ to map this meaning to a Freebase query y. Figure 2 shows examples of x, l_0 and y. In this section we describe our approach with the more detailed example derivation in Figure 3. We also describe the parameterization of a linear model that scores each derivation.

CCG parsing We use a CCG (Steedman, 1996) semantic parser (Kwiatkowski et al., 2013) to generate an underspecified logical form l_0. Figure 3a shows an example parse. The constants $Former$, $Municipalities$, in, $Brandenburgh$ in l_0 are not tied to the target knowledge base, causing the logical form to be underspecified. They can be replaced with Freebase constants in the later ontology matching step.

Ontological Matching The ontological match step has *structural match* and *constant match* components. Structural match operators can collapse or expand sub-expressions in the logical forms to match equivalent typed concepts in the target knowledge base. We adopt existing structural match operators (Kwiatkowski et al., 2013) and refer readers to that work for details.

Constant match operators replace underspecified constants in the underspecified logical form l_0 with concepts from the target knowledge base. There are four constant match operations used in Figure 3. The first two constant matches, shown below, match underspecified constants with constants of the same type from Freebase.

$$\begin{aligned} in &\rightarrow \texttt{location.containedby} \\ Brandenburgh &\rightarrow \texttt{BRANDENBURGH} \end{aligned}$$

However, because we are modeling the semantics of phrases that are not covered by the Freebase schema, we also require the following two constant matches:

$$\begin{aligned} Former(x) &\rightarrow \texttt{OpenType} \\ municipalities(x) &\rightarrow \texttt{OpenRel}(\texttt{x}, \texttt{Municipality}) \end{aligned}$$

Here, the word 'former' has been associated with a placeholder typing predicate since Freebase has no way of expressing end dates of administrative divisions. There is also no Freebase type representing the concept 'municipalities.' However, this word is associated with an entity in Freebase. Since there is no suitable linking predicate for the entity `Municipality`, we introduce a placeholder linking predicate `OpenRel` in the step from $l_2 \rightarrow l_3$. Our constant match operators can also introduce placeholder entities `OpenEntity` when there is no good match in Freebase.

(a) **CCG parse** builds an underspecified semantic representation of the sentence.

$$
\begin{array}{cccc}
\text{Former} & \text{municipalities} & \text{in} & \text{Brandenburgh} \\
\hline
N/N & N & N\backslash N/NP & NP \\
\lambda f \lambda x.f(x) \wedge former(x) & \lambda x.municipalities(x) & \lambda f \lambda x \lambda y.f(y) \wedge in(y,x) & Brandenburg
\end{array}
$$

$$
\begin{array}{cc}
\xrightarrow{\hspace{4cm}>} & \xrightarrow{\hspace{4cm}>} \\
N & N\backslash N \\
\lambda x.former(x) \wedge municipalities(x) & \lambda f \lambda y.f(y) \wedge in(y, Brandenburg)
\end{array}
$$

$$
\begin{array}{c}
\xrightarrow{\hspace{8cm}<} \\
N \\
l_0 = \lambda x.former(x) \wedge municipalities(x) \wedge in(x, Brandenburg)
\end{array}
$$

(b) **Constant matches** replace underspecified constants with Freebase concepts

$\mathbf{l_0} = \lambda x.former(x) \wedge municipalities(x) \wedge in(x, Brandenburg)$

$\mathbf{l_1} = \lambda x.former(x) \wedge municipalities(x) \wedge in(x, \texttt{Brandenburg})$

$\mathbf{l_2} = \lambda x.former(x) \wedge municipalities(x) \wedge \texttt{location.containedby}(x, \texttt{Brandenburg})$

$\mathbf{l_3} = \lambda x.former(x) \wedge \texttt{OpenRel}(x, \texttt{Municipality}) \wedge \texttt{location.containedby}(x, \texttt{Brandenburg})$

$\mathbf{l_4} = \lambda x.\texttt{OpenType}(x) \wedge \texttt{OpenRel}(x, \texttt{Municipality}) \wedge \texttt{location.containedby}(x, \texttt{Brandenburg})$

Figure 3: Derivation of the analysis for "Former municipalities in Brandenburgh". This analysis contains a placeholder type and a placeholder relation as described in Section 4.

We also allow the creation of typing predicates from matched entities through the introduction of linking predicates. For example, there is no native type associated with the word 'actor' in Freebase. Instead we create a typing predicate by matching the word to a Freebase entity `Actor` using Freebase API and allowing the introduction of linked predicates such as `person.profession` :

$$actor(x) \rightarrow \texttt{person.profession(x, Actor)}$$

Scoring Full Parses Our goal in this paper is to learn a function from the phrase x to the correct analysis y. We score each parse using a linear model with features that signal attributes of the underspecified parse ϕ_p and those that signal attributes of the ontological match ϕ_{ont}. Since the model factors over the two stages of parser, we split the prediction problem similarly. First, we select the maximum scoring underspecified logical form:

$$l^* = \arg \max_{l \in \text{CCG}(x)} (\theta_p \cdot \phi_p(l))$$

and then we select the highest scoring Freebase analysis y^* that can be built from l^*:

$$y^* = \arg \max_{r \in \text{ONT}(l^*, \mathcal{K})} (\theta_{ont} \cdot \phi_{ont}(r))$$

We describe an approach to learning the parameter vectors θ_p and θ_{ont} below.

5 Learning

We introduce a learning approach that first collates aggregate statistics from the 7 million Wikipedia entity-category pairs and existing facts in FB, and then uses a small labeled training set to tune the weights for features that incorporate these statistics.

Wikipedia Category	
Wars involving the Grand Duchy of Lituania	
Entity	Attribute
BattleOfGrunwald	`type(x, military.conflict)`
GollubWar	`type(x, military.conflict)`
BattleOfGrunwald	`time.event.loc(x, Grunwald)`
. . .	. . .
Entity	Relation
BattleOfGrunwald	`military_conflict.combatants`
GollubWar	`time.event.start_time`
BattleOfGrunwald	`military_conflict.commanders`
. . .	. . .

Figure 4: Labeled entities are associated with attributes and relations.

Broad Coverage Lexical Statistics Each Wikipedia category is associated with a number of entities, most of which exist in FB. We use these entities to extract relations and attributes in FB associated with that category. For example, in Figure 4 the category 'Wars involving the Grand Duchy of Lithuania' is associated with the relation `military_conflict.combatants` and the attribute `type(x, military.conflict)` multiple times, because they are present in many of the category's entities. For each of the sub-phrases in the category name we count these associations over the entire Wikipedia category set.

We use these counts to calculate Pointwise Mutual Information (PMI) between words and Freebase attributes or relations. We choose PMI to avoid overcompensating common words, attributes, or relations. For example, the word 'Wars' is seen with the incorrect analysis `type(x, time.event)` more frequently than the correct analysis `type(x, military.conflict)`. However, PMI penalizes the attribute `type(x, time.event)` for its popularity and the correct analysis is preferred. As PMI has a tendency to emphasize rare counts, we chose PMI squared, which takes the squared value of the co-occurence count (PMI^2(a, b) = $\log\frac{count(a \wedge b)^2}{count(a)*count(b)}$), as a feature.

Structural KB Statistics Existing semantic parsers typically make use of type constraints to limit the space of possible logical forms. These strong type constraints are not feasible when the knowledge base is incomplete. For example, in Freebase the relation `military_conflict.combatants` expects an entity of type `military_conflict.combatant` as its object. However, many countries that have been involved in wars are not assigned this type.

We instead calculate type overlap statistics for all Freebase entities, to find likely missing types. For example, including the fact that the object of `military_conflict.combatants` is very often of type `location.country`.

Learning from Labeled Data We train each half of the prediction problem separately, as defined in Section 4, using the labeled training data introduced in Section 3. We use structured max-margin perceptrons to learn feature weights for both the underspecified parse and the ontological match step following (Kwiatkowski et al., 2013). The aggregate statistics collected from 7 million category-entity pairs produce very useful lexical features. We integrate these statistics into our linear model by summing their values for each derivation and treating them as a feature. All of the other features described in Section 6 are not word specific and are therefore far less sparse.

6 Features

We include a number of features that enable soft type checking on the output logical form, described first below, along with other features that measure different aspects of the analysis.

Coherency features For example, consider the phrase "The UK home city of the Queen," with Freebase logical form $y = \lambda \mathtt{x}.\mathtt{home}(\mathtt{QEII}, \mathtt{x}) \wedge \mathtt{in}(\mathtt{x}, \mathtt{UK}) \wedge \mathtt{city}(\mathtt{x})$. Each of the relations has expected types for their argument: the relation $\langle\mathtt{home}\rangle$ expects a subject of type $\langle\mathtt{person}\rangle$ and an object of type $\langle\mathtt{location}\rangle$. Each type in Freebase lives in a hierarchy, so the type `city` implies $\{\mathtt{location}, \mathtt{administrative\_division}, \ldots\}$. The next four features test agreement of these types on different parts of the output logical form.

Relation arguments trigger a feature if their type is in the set of types expected by the relation. `QEII` is a person so this feature is triggered for the relation-argument application in $\mathtt{home}(\mathtt{QEII}, \mathtt{x})$.

Relation-relation pairs can share variable arguments. For example, the variable x is the object of $\langle\mathtt{home}\rangle$ and the subject of $\langle\mathtt{in}\rangle$. Each relation expects a set of types of x. We have features to signal if: these sets are disjoint; one set subsumes the other; and the PMI between the highest level expected type (described in Section 5) if the sets are disjoint. In the example given here, the type $\langle\mathtt{location}\rangle$ expected by $\langle\mathtt{in}\rangle$ subsumes the type $\langle\mathtt{city}\rangle$ expected by $\langle\mathtt{home}\rangle$ so the second feature fires. We treat types such as $\mathtt{city}(x)$ as unary relations and include them in this feature set.

Type domain measures compatibility among domains in Freebase. Freebase is split into high-level domains and some of these are relevant, such as 'football' and 'sports'. We identify those by counting their co-occurrences. This becomes an indicator feature that signals their co-occurrence in y.

Named entity type features test if the entity e that we are extracting attributes for have Freebase type "person", "location" or "organization". If it does, we have a feature indicating if y defines a set of the same type. This features is not used in the referring expression task presented in Section 7 since we cannot assume access to the entities that are described.

CCG parse feature signals which lexical items were used in the CCG parse. Another feature fires if capitalized words map to named entities.

String similarity features signal exact string match, stemmed string match, and length weighted string edit distance between a phrase in the sentence and the name of the Freebase element it was matched on. We also use the Freebase search API to generate scores for phrase, entity pairs and include the log of this score as a features.

Lexical PMI feature includes the lexical Pointwise Mutual Information described in Section 5.

Freebase constant features signal the use of linking predicates, as defined in Section 4, and the log frequency count of the Freebase attributes across all entities in the Wikipedia category set.

Other features indicate the use of `OpenRel`, `OpenEntity`, `OpenType` in y and count repetitions of Freebase concepts in y.

7 Experimental Setup

Knowledge base We use the Jan. 26, 2014 Freebase dump. After pruning binary predicates taking numeric values, it contains 9351 binary predicates, 2754 unary predicates, and 1.2 billion assertions.

Pruning and Feature Initialization We perform beam search at each semantic parsing stage, using the Freebase search API to determine candidate named entities (10 per phrase), binary predicates (300 per phrase), and unary predicates (500 per phrase). The ontology matching stage considers the highest scored underspecified parse.

The features are initialized to prefer well-typed logical forms. Type checking features are initially set to -2 for mismatch. Features signalling incompatible topic domains and repetition are initialized as -10. All other initial feature weights are set to 1.

Datasets and Annotation We evaluate on the Wikipedia category and appositive datasets introduced in Sec. 3. On the Wikipedia development data, we annotated 500 logical forms, underspecified logical forms and constant mappings for ontology matching. The Wikipedia test data is composed of 500 unseen categories. We did not train on the appositive dataset, as it contains challenges such as co-reference and parsing errors as described in Sec. 3. Instead, we chose 300 randomly selected examples for evaluation, and ran on the model trained on the Wikipedia development data.

Evaluation Metrics We report five-fold cross validation for development but ran the final model once on the test data, manually scoring the output.

For evaluation on the referring expression resolution performance (as defined in Sec. 2), we include accuracy for the final logical form (*Exact Match*). We also evaluate precision and recall for predicting individual literals in this logical form on the development set. To control for missing facts, we did not evaluate the set of returned entities.

To evaluate entity attribute extraction performance (as defined in Sec. 2), we identified three classes of predictions. Extractions can be correct, benign, or false. Correct attributes are actually described in the phrase, benign extraction may not have been described but are still true, and false extractions are not true. For example, if the phrase "the capital of the communist-ruled nation" is mapped to the pair of attributes `capital_of_administrative_division(x)` , `location(x)`, the first is correct and the second is benign. Other incorrect facts would be false.

System	Exact Match	Partial Match		
		P	R	F1
KCAZ13	1.4	9.6	6.3	7.0
IE Baseline	6.8	37.0	23.3	28.6
NoPMI	11.0	23.7	20.8	21.6
NoOpenSchema	13.7	35.8	30.0	31.1
NoTyping	9.6	37.6	29.3	31.8
Our Approach	15.9	39.3	33.5	35.1
with Gold NE	20.8	46.6	40.5	42.3

Table 3: Referring expression resolution performance on the development set on gold references.

Data	System	Exact Match Accuracy
Wikipedia	IE Baseline	21.8%
	Our Approach	28.4%
Appos	IE Baseline	0.0%
	Our Approach	4.7%

Table 4: Manual evaluation for referring expression resolution on the test sets.

On the development set, we report precision and recall against the union of the FB attributes in our annotations without adjusting for benign extractions or the fact that the annotations are not complete. For the test sets, we computed precision (P) where benign extractions are considered to be wrong, as well as an adjusted precision metric (P*) where benign extractions are counted as correct. As we do not have full test set annotations, we cannot report recall. Finally, we report the average number of facts extracted per noun phrase (fact #).

Comparison Systems We compare performance to a number of ablated versions of the full system, where we have removed the open-constant ontology matching operators (NoOpenSchema), the PMI features (NoPMI), or the type checking features (NoTyping). For the referring expression resolution task, we excluded the named entity type feature, as this assumes typing information about the entity we are extracting attributes for.

We report results without the PMI features and the open schema matching operators (KCAZ13), which is a reimplementation of a recent Freebase QA model (Kwiatkowski et al., 2013). We also learn with gold named entity linking (Gold NE).

For the entity attribute extraction, we built a supervised learning baseline that combines the output of two discrete SVMs, one for predicting unary relations and one for binary relations. Each classifier

System	Top n	P	R	F1	fact #
IE Baseline	-	37.3	26.5	30.6	1.6
Our Approach	1	44.2	32.8	37.7	1.9
	2	36.9	38.0	37.5	2.6
	3	30.7	42.7	35.7	3.6
	4	27.0	44.7	33.6	4.2
	5	23.7	47.2	31.6	5.1
	10	15.9	52.0	24.3	8.5

Table 5: Entity attribute extraction performance on the Wikipedia category development set.

Data	System	P	P*	fact #
Wikipedia	IE Baseline	56.7	58.7	1.6
	Our Approach	61.2	72.6	2.0
Appos	IE Baseline	4.9	13.9	1.3
	Our Approach	33.2	61.4	0.9

Table 6: Manual evaluation for entity attribute extraction on the test sets.

is trained using the annotated Wikipedia categories. This dataset contains hundreds of unary and binary relations, which the IE baseline can predict. Each classifier is further anchored on a specific word, and includes n-gram and POS context features around that word, following features from Mintz et al (2009). To predict binary relations, we used named entities as anchors. For unary attributes we anchored on all possible nouns and adjectives. The final logical form includes the best relation predicted by each classifier. We use the Stanford CoreNLP[5] toolkit for tokenization, named entity recognition, and part-of-speech tagging.

8 Results

Tables 3 and 4 show performance on the referring expression resolution task. Tables 5 and 6 show performance on the extraction task. Reported precision is lower on the labeled development set than on the test set, where predicted logical forms are manually evaluated. This reflects the fact that, despite our best attempts, the development set labels are incomplete, as discussed in Section 3.

Referring expression resolution The systems retrieve the full meaning with 28.4% accuracy on the Wikipedia test set, and 15.9% on the development set. The gold named entity input improves performance by modest amounts. This suggests that the errors stem from ontology mismatches, as we will describe in more detail later in the qualitative analysis. We also see that all of the ablations hurt performance, and that the KCAZ13 model performs extremely poorly. The independent classifier baseline performs well at the sub-clause level, but fails to form a full logical form of the referring expression. Partial grounding and broad-coverage data statistics are essential for this problem.

Entity attribute extraction In the two test sets, the approach achieves high benign precision levels (P*) of 72.6 and 61.4. However, the appositives data is significantly more challenging, and the model misses many of the true facts that could be extracted. Many errors comes in the early stages of the pipeline, which can be attributed at least in part to both (1) the higher levels of noise in the input data (see Section 3), and (2) the fact that the CCG parser was developed on the Wikipedia category labels. While the IE baseline performs reasonably on the Wikipedia test data, its performance degrades significantly on appositions. As it is trained to predict pre-determined relations, it does not generalize to different domains.

For the development set, Table 5 also shows the precision-recall trade off for the set of Freebase attributes that appear in the top-n predicted logical forms. Precision drops quickly but recall can be improved significantly, showing that the model can produce many of the labeled facts.

Qualitative evaluation We sampled 100 errors from the Wikipedia test set for qualitative analysis. 10% came from entity linking. About 30% come from choosing a superset or subset of the desired meaning, for example by mapping “novel” to `book`. About 10% of the errors are from domain ambiguity, such as mapping “stage actor” to `film.film_actor`. 10% of the cases are from spurious string similarity, such as mapping “Hungarian expatriates“ to `nationality(x,Hungary)`. 15% of the failures were due to incorrect underspecified logical forms and, finally, about 10% of the errors were because the typing features encouraged compound nouns to be split into separate attributes. On the apposition dataset, 65% of errors stems from parsing, either in apposition detection or CCG parsing. Better modeling the complex attachment decisions for the noun phrases in the apposition dataset remains an area for future work.

One advantage of our approach, especially in comparison to classifier based models like the IE baseline, is the ability to predict previously unseen relations. Counting only the correctly predicted

[5] http://nlp.stanford.edu/software/corenlp.html

triples, we see that over 40% of the unique relations we predict is not in the development set; our model learns to generalize based on the learned PMI features and other lexical cues.

Finally, our approach extracted 2.0 entity attributes per Wikipedia phrase and 0.9 per apposition on average. This matches our intuition that the apposition dataset contains many more words that cannot be modeled with concepts in Freebase.

9 Related Work

Recent work has begun to study the problem of knowledge base incompleteness and reasoning with open concepts. Joshi et al. (2014) describes an approach for mapping short search queries to a single Freebase relation, that benefits from modeling schema incompleteness. Additionally, Krishnamurthy et al. (2012; 2014) present a semantic parser that builds partial meaning representations with Freebase for information extraction applications. This is similar in spirit to the approach we present here, however they focus on a small, fixed, set of binary relations while we aim to represent as much of the text as possible using the entire Freebase ontology. Krishnamurthy and Mitchell (2015) have also studied semantic parsing with open concepts via matrix factorization. They use Freebase entities but do not include Freebase concepts.

The problem of building complete sentence analyses using all of the Freebase ontology has recently received attention within the context of question answering systems (Cai and Yates, 2013; Kwiatkowski et al., 2013; Berant et al., 2013; Berant and Liang, 2014; Reddy et al., 2014). Since they do not model KB incompleteness, these models will not work well on data that cannot be fully modeled by Freebase. In section 7, we report results using one of these systems to provide a reference point for our approach. There has also been other work on Freebase question answering (Yao and Van Durme, 2014; Bordes et al., 2014; Wang et al., 2014) that directly searches the facts in the KB to find answers without explicitly modeling compositional semantic structure. Therefore, these methods will suffer when facts are missing.

The syntactic and semantic structure of noun phrases has been extensively studied. For example, work on NomBank (Meyers et al., 2004; Gerber and Chai, 2010) focus on the challenge of modeling implicit arguments introduced by nominal predicates. In a manual study, we discovered that the 65% of our noun phrases contain implicit relations. We build on insights from Vadas and Curran (2008), who studied how to model the syntactic structure of noun phrases in CCGBank. While we are, to the best of our knowledge, the first to study compound noun phrases for semantic parsing to knowledge-bases, semantic parsers for noun phrase referring expressions have been built for visual referring expression (FitzGerald et al., 2013).

There has been little work on IE from compound noun phrases. Most existing IE algorithms extract a single relation, usually represented as a verb that holds between a pair of named entities, for example with supervised learning techniques (Freitag, 1998) or via distant supervision (Mintz et al., 2009; Riedel et al., 2013; Hoffmann et al., 2011). We aim to go beyond relations between entity pairs, and to retrieve full semantics of noun phrases, extracting unary and binary relations for a single entity. A notable exception to this trend is the ReNoun system (Yahya et al., 2014) which models noun phrase structure for open information extraction. They report that 97% of the attributes in Freebase are commonly expressed as noun phrases. However, unlike our work, they considered open information extraction and did not ground the extractions in an external KB.

10 Conclusion

In this paper, we present a semantic parsing approach with knowledge base incompleteness, applied to the problem of information extraction from noun phrases. When run on all of the Wikipedia category data, the approach would extract up to 12 million new Freebase facts at 72% precision.

There is significant potential for improving the parsing models, as well as better optimizing the precision recall trade-off for the extracted facts. It would also be interesting to gather data with compositional phenomena, such as negation and disjunction, and study its impact on the performance of the semantic parser.

Acknowledgments

The research was supported in part by DARPA, under the DEFT program through the AFRL (FA8750-13-2-0019) and the CSSG (N11AP20020), as well as the NSF (IIS-1115966, IIS-1252835) and a gift from Google, Inc. The authors thank Xiao Ling for providing the apposition data, and Yoav Artzi, Nicholas FitzGerald, Mike Lewis, Sameer Singh and Mark Yatskar for discussion and comments.

References

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the Empirical Methods in Natural Language Processing*.

Antoine Bordes, Jason Weston, and Sumit Chopra. 2014. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar. Association for Computational Linguistics.

Qingqing Cai and Alexandar Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Nicholas FitzGerald, Yoav Artzi, and Luke Zettlemoyer. 2013. Learning distributions over logical forms for referring expression generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Dayne Freitag. 1998. Toward general-purpose learning for information extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*.

Matthew Gerber and Joyce Y Chai. 2010. Beyond nombank: a study of implicit arguments for nominal predicates. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Dan Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the Conference of the Association of Computational Linguistics*.

Mandar Joshi, Uma Sawat, and Soumen Chakrabarti. 2014. Knowledge graph and corpus driven segmentation and answer inference for telegraphic entity-seeking queries. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jayant Krishnamurthy and Tom M Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics.

Jayant Krishnamurthy and Tom M. Mitchell. 2014. Joint syntactic and semantic parsing with combinatory categorial grammar. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jayant Krishnamurthy and Tom Mitchell. 2015. Learning a compositional semantics for freebase with an opne predicate vocabulary. *Transactions of the Association for Computational Linguistics*.

Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating noun argument structure for nombank. In *LREC*, volume 4.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2.

Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. 2013. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics*.

Mark Steedman. 1996. *Surface Structure and Interpretation*. The MIT Press.

David Vadas and James R. Curran. 2008. Parsing noun phrase structure with ccg. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Zhenghao Wang, Shengquan Yan, Huaming Wang, and Xuedong Huang. 2014. An overview of microsoft deep qa system on stanford webquestions benchmark. Technical Report MSR-TR-2014-121, September.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of World Wide Web Conference*.

Mohamed Yahya, Steven Euijong Whang, Rahul Gupta, and Alon Halevy. 2014. Renoun: Fact extraction for nominal attributes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.