

# ILLINOIS-PROFILER: Knowledge Schemas at Scale

Zhiye Fei, Daniel Khashabi, Haoruo Peng, Hao Wu, Dan Roth

University of Illinois, Urbana-Champaign, Urbana, IL, 61801

{zfei2, khashab2, hpeng7, haowu4, danr}@illinois.edu

## Abstract

In many natural language processing tasks, contextual information from given documents alone is not sufficient to support the desired textual inference. In such cases, background knowledge about certain entities and concepts could be quite helpful. While many knowledge bases (KBs) focus on combining data from existing databases, including dictionaries and other human generated knowledge, we observe that in many cases the information needed to support textual inference involves detailed information about entities, entity types and relations among them; e.g., is the verb “fire” more likely to occur with an organization or a location as its Subject? In order to facilitate reliable answers to these types of questions, we propose to collect large scale graph-based statistics from huge corpora annotated using state-of-the-art NLP tools. In order to systematically design, acquire and access such a knowledge base, we formalize a class of tree based knowledge schemas based on Feature Description Logic (FDL). We define a range of knowledge schemas, then extract and organize the resulting statistical KB using a new tool, the PROFILER. Our experiments demonstrate that the PROFILER helps in classification and textual inference. Specifically, we demonstrate its application to co-reference resolution, showing considerable improvements by careful use of the PROFILER’s statistical knowledge.

## 1 Introduction

Knowledge *representation* and *acquisition* are two central problems in the design of Natural Language Processing (NLP) systems. While these are separate processes, they are absolutely dependent. A *representation* needs to be expressive enough, yet it should not compromise efficiency at acquisition and inference stages [Levesque and Brachman, 1987]. Most importantly, usage of patterns of a KB should dictate how knowledge is represented and how it is acquired.

Beyond relatively “static” and manually curated knowledge bases such as Wordnet and Freebase [Bollacker *et al.*, 2008] the last few years have seen a significant body of

work attempting to acquire KBs. However, even these efforts typically aim at acquiring sets of rules or graphs; examples include significant efforts on open domain information extraction [Banko *et al.*, 2007], using automatic acquisition methods to acquire structured knowledge bases such as the Extended WordNet and the YAGO ontology [Hoffart *et al.*, 2013], and projects such as NELL [Carlson *et al.*, 2010], acquiring a large body of “rules” and “facts”. However, almost all these efforts, while using huge amounts of data, are quite simplistic from the representational perspective – often using simple relational patterns (e.g. *X such as Y, Z*) to extract information from texts; they rarely facilitate *typing* of information, do not support disambiguation (which *Ford* is it?) and do not use the structure of the text in any significant way. These acquisition efforts have seen minimal success in supporting textual inference, mostly, we believe, due to the knowledge representation used – sets of rules or graphs that do not correspond well to inference patterns they need to support.

Consider the problem of co-reference resolution. For the co-reference resolution of the pronoun “he” in the sentence “*Jimbo arrested Robert because he stole an elephant from the zoo*”, there is a need to use some global statistical knowledge indicating that the *Obj* of *arrest* is more likely than its *Subj* to be the *Subj* of *stole*. Providing an NLP system with such information requires that we think carefully about the knowledge representation and about the acquisition process that could support it.

This paper proposes a general framework for relational representation of (statistical) knowledge that is acquired from text, and is designed to support textual inferences. Our representation is driven by the need to facilitate aggregation of knowledge while taking into account *typing* of concepts and entities, their *disambiguation*, and the relational *structure* of the text. In particular, the need to support inferences of the kind shown above necessitates that the knowledge acquisition is designed appropriately to facilitate it.

Our knowledge representation is designed based on Feature Description Logic (FDL), a relational (frame-based) language that is expressive yet supports efficient inference [Cumby and Roth, 2003a; 2003b]. Variations of this language were shown to be useful in multiple applications, from providing principled formulations for feature extraction in machine learning, to formalisms for the semantic web [Baader *et al.*, 2008]. In this work, we use FDL to create a formal

Verbs in dependency path of length 2		Nouns in dependency path of length 2	
Seattle, WA	Seattle Seahawks	grow.03: agriculture	grow.04: go from child to adult
base 0.182	play 0.23	area 0.136	school 0.13
include 0.1	sign 0.1	plant 0.126	family 0.11
serve 0.059	49er 0.072	species 0.075	town 0.067
arrive 0.056	join 0.07	variety 0.067	city 0.065
work 0.055	win 0.067	region 0.059	son 0.061
depart 0.052	select 0.055	soil 0.054	father 0.058
know 0.052	lead 0.053	tree 0.054	child 0.056
form 0.049	make 0.035	garden 0.051	farm 0.055
win 0.049	release 0.033	wine 0.048	area 0.051
become 0.04	defeat 0.031	crop 0.042	village 0.04
bear 0.037	run 0.031	vegetable 0.042	suburb 0.039
found 0.036	become 0.029	fruit 0.036	brother 0.036
play 0.035	place 0.027	flower 0.032	mother 0.035
leave 0.034	announce 0.027	year 0.03	year 0.031
join 0.031	begin 0.025	forest 0.029	neighborhood 0.029
make 0.029	go 0.025	seed 0.027	parent 0.029
locate 0.027	beat 0.022	elevation 0.026	member 0.028
attend 0.027	draft 0.022	time 0.023	daughter 0.027
run 0.025	hold 0.022	field 0.023	county 0.026
hold 0.024	hire 0.022	group 0.021	university 0.026

Table 1: Visualization of sample profiles. The tokens in the vertical axis are the most frequent co-occurring tokens in the corresponding schema, and the associated numbers are normalized by the total number of co-occurrences. The detailed definition of schema is presented in §2.

definition for different types of knowledge schemas defined based on graphs, acquire information from data using these schemas, and retrieve information from our KB using these schemas to support textual inference decisions.

With this flexible schema definition, we are able to extract many useful occurrence statistics on big corpora. Table 1 shows how we structure the extracted information in the PROFILER. The statistics that share an important common constituent are gathered into the same *profiles*. For example, all of the schema instances which contain the entity “Seattle” (the city) as one of their constituents are gathered in the profile of “Seattle” (the city). Similarly we have profiles for “Seattle” (Seahawks), “grow” (sense 3, meaning “produce by cultivation”), “grow” (sense 4, meaning “go from child to adult”), and so on. As can be seen, there can be multiple profile types, e.g. (Wikipedia based) entities, (Propbank based) Verbsense [Kingsbury and Palmer, 2002], etc.<sup>1</sup> Each *profile* has a set of keys that uniquely identifies it. For example, profiles of Wikipedia entities are uniquely identified by both their surface form and the Wikipedia url. In doing this, we are able to disambiguate different entities that have the same surface form, as we show in the visualization.

Given an entity, we can look at various statistics gathered for it. Different schemas might be useful for different tasks. Here we give examples for the problems of named entity recognition and co-reference resolution, although the usage of our schemas is not limited to these applications.

Consider the examples in Table 2. For each sentence and task, a graph is provided to represent the useful schemas. In the first sentence we want to solve a co-reference problem, where the goal is to connect “it” to either “the tree” or “axe”. If we have statistics indicating whether it is more likely to have the adjective “tall” applied to “the tree” or “axe”, we can solve this problem. The schema graph of this knowledge is represented in the first example of Table 2, where  $w$  is a word

<sup>1</sup>Currently we only support the two mentioned profile types, although adding more profile types is just a matter of adding new input annotations.

and can be instantiated as “the tree” and “axe”. The statistics from the two graph instances provides information that is essential to addressing this inference task. In the second example sentence, we want to tag the constituents with NER labels. Having statistics on the pattern PER “bought” ORG, where PER and ORG are possible NER labels, can be used to model the correlation between labeling of two local constituents based on their mutual context. In the graph, the arc labels  $R_1$  and  $R_2$  could be substituted with any proper relations which approximate our goal, such as  $(R_1, R_2) = (\text{before}, \text{after})$ . Consider the co-reference problem in example 3. The knowledge that the *subject* of “steal” is more likely to be the *object* of “arrest” rather than the *subject* of “arrest” would provide enough information for this instance. This corresponds to the graph provided for this example in which *Subj* of “steal” is co-referred to *Obj* of “arrest”.

As can be observed from our examples, there is a wide range of patterns that can be defined for different problems and applications. We create a unified language for expressing these schemas, and we implement a scalable system that allows concise schema definitions and fast statistics extraction from gigantic corpora. To summarize the main contributions of this paper:

1. We propose a formalization for graph-based representation of knowledge schemas, based on FDL.
2. We create PROFILER, a publicly available tool which contains statistics of various knowledge schemas.
3. Finally, we show the application of our tool to dataless classification of people-occupations. We also address hard co-reference resolution problems and show considerable improvements.

The rest of the paper is organized as follows. We explain the graph-based knowledge schemas formulation in §2. The details of the acquisition system are given in §3. We report our preliminary experimental results in §4.

#	Sentence	Schema Graph
1	“I chopped down [the tree] with my [axe] because [it] was tall.”	
2	“[Larry Robbins], founder of Glenview Capital Management, bought shares of [Endo International Plc] ...”	
3	“[Jimbo] arrested [Robert] because [he] stole an elephant from the zoo.”	

Table 2: Example sentences and schema graphs.

Attributes ( $\mathcal{A}$ )	Values ( $\mathcal{V}$ )
Word	Raw text
Lemma	Raw text
POS	labels form Penn Treebank
NER	{ PER, ORG, LOC, MISC }
Wikifier	Wikipedia urls
Verbsense	Verb sense from Verbnets
Role	{ subj, obj }

Table 3: Set of *attribute-values* in our current system.

## 2 Generic Knowledge Description

In this section, we introduce a formal framework for characterizing word-tuple occurrences defined based on concept graphs  $G(V, E)$ , with certain labeling properties. In our formalization, we use FDL as a means to represent relational data as quantified propositions [Khardon *et al.*, 1999; Cumby and Roth, 2003a], with some additional notation. We first give a brief introduction of FDL<sup>2</sup> and explain how we can scale our knowledge description with any schema by taking advantage of FDL’s inductive nature.

**Definition 1** Given a set of attributes  $\mathcal{A} = \{a_1, a_2, \dots\}$ , a set of values  $\mathcal{V} = \{v_1, v_2, \dots\}$  and a set of role alphabets  $\mathcal{R} = \{r_1, r_2, \dots\}$ , an FDL description is defined inductively as following:

1. For an attribute  $a \in \mathcal{A}$  and a value  $v \in \mathcal{V}$ ,  $a(v)$  is a description, and it represents the set  $x \in \mathcal{X}$  for which  $a(x, v)$  is True.
2. For a description  $D$  and a role  $r \in \mathcal{R}$ ,  $(r D)$  is a role description. Such description represents the set  $x \in \mathcal{X}$  such that  $r(x, y)$  is True, where  $y \in \mathcal{Y}$  is described by  $D$ .
3. For given descriptions  $D_1, \dots, D_k$ , then  $(\text{AND } D_1, \dots, D_k)$  is a description, which represents a conjunction of all values described by each description.

Based on the rules of the FDL, the output of each *description* is a set of elements. We will denote the description of each node  $i$  with  $D_i$ . Our goal is to describe the whole concept graph or, in other words, to find tuples of words  $(c_1, c_2, \dots, c_n)$  (where  $n = |V|$ ) which conform to the *roles*

<sup>2</sup>More details can be found in [Cumby and Roth, 2003a].

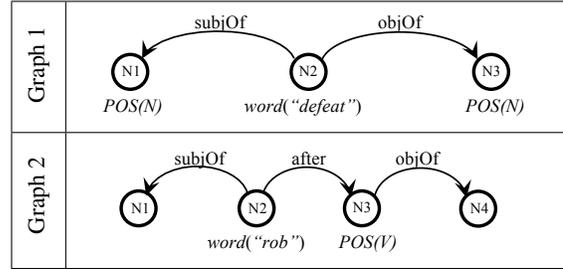


Figure 1: Concept graphs for examples used here. Each concept graph gives a relational description of words. Each node represents a word. The label on each node is an *attribute-value* pair for that word. The labels on edges are the relations (or *roles* based on FDL notation) between words.

Roles ( $\mathcal{R}$ )
Before
After
NearestBefore
NearestAfter
AdjacentToBefore
AdjacentToAfter
ExclusiveContaining
HasOverlap
DependencyPath ( $l$ )
Co-referred
SubjectOf
IsSubjectOf
ObjectOf
IsObjectOf

Table 4: Set of *roles* in our current system.

and *attribute-values* defined based on our desired graph. To this end, we will cross-product the descriptions of individual nodes to get the description of the whole graph. For future use, we define  $D_{i_1, \dots, i_k}$ , a set of  $k$ -element tuples, as the description of nodes  $i_1, \dots, i_k \in [|V|]$ <sup>3</sup>.

Before stating the complete definition of the schemas, we first give a couple of examples. For each example, a concept graph represents the set of *roles* on edges, and *attribute-values* on nodes. The concept graphs corresponding to our examples are depicted in Figure 1.

<sup>3</sup> $[k] = \{1, \dots, k\}$

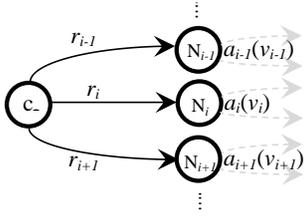


Figure 2: One level of a hypothetical concept graph.

**Example 1:** Suppose we want to describe a pair of nouns serving as subject and object of a verb “defeat”, respectively.

$$D_1 = (\mathbf{AND}(\text{POS}(\mathbb{N}))(\text{subjectOf word(“defeat”)}))$$

$$D_2 = \{\text{word(“defeat”)}\}$$

$$D_3 = (\mathbf{AND}(\text{POS}(\mathbb{N}))(\text{objectOf word(“defeat”)}))$$

In Graph 1,  $D_1$  describes  $N1$  and  $D_3$  describes  $N3$ . Their cross-product,  $D_{1,2,3} = D_1 \otimes D_2 \otimes D_3$  represents the set of tuples.

**Example 2:** Consider Graph 2 in Figure 1. Given the verb “rob”, we want to describe nodes  $N1, N3, N4$ .

$$D_1 = (\text{subjectOf word(“rob”)})$$

$$D_2 = \{\text{word(“rob”)}\}$$

$$D_3 = (\mathbf{AND}(\text{POS}(\mathbb{V}))(\text{after word(“rob”)}))$$

$$D_4(w) = (\text{objectOf word}(w)), \forall w \in D_3$$

$$D_{3,4} = \bigcup_{w \in D_3} (\{w\} \otimes D_4(w))$$

The cross product of the definitions gives the set of all possible quadruples:  $D_{1,2,3,4} = D_1 \otimes D_2 \otimes D_{3,4}$ .

**A general schema definition:** We showed how to formally characterize the set of all elements satisfying a concept graph in our examples. Here we provide general characterization protocols. As mentioned before, in the concept graph the edges are roles  $r \in \mathcal{R}$ , and nodes contain attributes  $a \in \mathcal{A}$  from the set of values  $v \in \mathcal{V}$ . Figure 2 shows one layer of a hypothetical concept graph. Assuming that the concept graph is a rooted tree (hence no cycles) the following rule inductively defines the description of each node in a concept graph:

**Description of each node:** If the parent is fixed (or is a single-element set), the description of each child node is inductively defined based on the description of its parent, given the role/attribute labels. The description of the node  $N_i$  is:

$$D_i = (\mathbf{AND}(a_i(v_i))(r_i \text{ word}(c)))$$

If the parent is described by a set,  $D_{\text{parent}}$ , the description of each child node is inductively defined based on each element of the parent description, given the role/attribute labels. The description of the node  $N_i$ :

$$D_i(c) = (\mathbf{AND}(a_i(v_i))(r_i \text{ word}(c))), \forall c \in D_{\text{parent}}$$

This definition could be further changed depending on the concept graph. For example, if there is no attribute or role associated with the child, they could be omitted. Similarly, if there are multiple attributes or roles, they can be combined with **AND** or any proper operators.

Given the description of each node, we want the joint description of all nodes in the concept graph. In the following we formalize how to combine atomic descriptions and get a global description.

**Combining atomic descriptions:** For a fixed parent, the description of the parent-children nodes is the cross product of the child descriptions with the parent. If  $\mathcal{I}$  represents the set of indices for children:

$$D_{\text{parent, child}} = D_{\text{parent}} \otimes \left( \bigotimes_{i \in \mathcal{I}} D_i \right)$$

Suppose a parent node is described by a set of elements,  $D_{\text{parent}}$ . If  $\mathcal{I}$  represents the set of indices for children, the description of the parent-child nodes is:

$$D_{\text{parent, child}} = \bigcup_{c \in D_{\text{parent}}} \left[ \{c\} \otimes \left( \bigotimes_{i \in \mathcal{I}} D_i(c) \right) \right]$$

**A database of schema scores:** So far, given a graph  $G(V, E)$  we have created a description  $D_{1, \dots, |V|}$ . We define a scoring function  $\mathcal{S} : G \rightarrow \mathbb{R}$  which maps a concept graph to a score value. Now given a set of tuples of the form  $(c_1, \dots, c_{|V|})$ , the basic score can be the number of distinct tuples. The distinctness could be defined either based on the raw text, a normalized form or any of the attributes (Table 3), which defined the level of abstraction.

In many applications, the raw occurrence score might not be very useful; instead, the conditional probabilities are handy. The conditional probability in a graph could be defined in different ways. Consider a subgraph of  $G$ , defined as  $G'(E', V')$ , such that  $E' \subseteq E$  and  $V' \subseteq V$ . A normalized score of  $G$  with respect to its subset is defined as

$$\mathcal{S}(G \setminus G' | G') \triangleq \frac{\mathcal{S}(G)}{\mathcal{S}(G')}$$

Depending on the application, a different definition of  $G'$  might be appropriate.

**Querying the database:** Our generic knowledge description gives us a uniform interface for both knowledge acquisition and queries. In order to get the scores, the user only needs to supply a specific instance of any schema definition  $G$ .

### 3 Knowledge Acquisition Procedure

In this section we explain the knowledge acquisition procedure, the flowchart of which is shown in Figure 3. To support fast data retrieval, we decide to pre-compute the statistics for our schemas, instead of querying document annotations in real time.<sup>4</sup> First, we use ILLINOIS CLOUD NLP [Wu et al., 2014] to process raw documents with the annotations we need in our schemas. ILLINOIS CLOUD NLP is built around

<sup>4</sup>Pre-computing has the benefit of fast queries, although it limits the users in query time. It is possible to create a combination of pre-computing and on the fly computation of statistics for arbitrary queries, which is the subject of our future work.

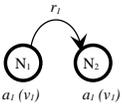
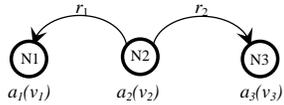
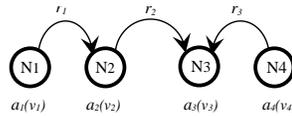
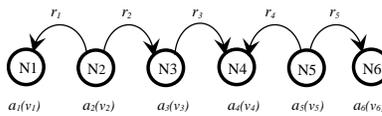
Concept Graph	Attributes = { Values }	Relations	# of Schemas
	$\text{word} = \{ \text{set of words} \}$ $\text{POS} = \{ \text{Noun, Noun-Phrase, Verb, Verb-Phrase, Modifier} \}$ $\text{Wikifier} = \{ \text{URLS} \}$ $\text{Verbsense} = \{ \text{All verb senses} \}$	Possible roles from Table 4 except Co-referred	24
	$\text{word} = \{ \text{set of words} \}$ $\text{POS} = \{ \text{Noun, Noun-Phrase, Verb, Verb-Phrase, Modifier} \}$ $\text{Verbsense} = \{ \text{All verb senses} \}$	Subj, ObjOf	2
	$\text{word} = \{ \text{All words} \}$ $\text{Verbsense} = \{ \text{All verb senses} \}$	Subj, ObjOf, Co-referred	8
	$\text{word} = \{ \text{set of words} \}$ $\text{Verbsense} = \{ \text{All verb sense} \}$	Subj, ObjOf, Co-referred	4

Table 5: Categorization of knowledge schemas used in experiments based the structure of the concept graph.

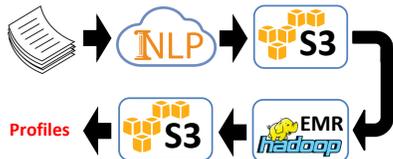


Figure 3: Flowchart of Profiler data processing.

NLPCURATOR [Clarke *et al.*, 2012] and Amazon Web Services’ Elastic Compute Cloud (EC2). It provides a straightforward user interface to annotate large corpora with a variety of NLP tools supported by NLPCURATOR, on EC2. We use Amazon Elastic MapReduce to aggregate the statistics. It takes input directly from Amazon’s distributed key-value storage S3, where ILLINOISCLOUDNLP stores its annotation output. Finally, we store our profiles in MongoDB as it supports a flexible data model and scalable indexing.

In our experiments we annotate a large part of a Wikipedia dump that contains 4,019,936 documents. The serialized annotation output is 1,455 GB in size when uncompressed, and we use it as the input of our Elastic MapReduce program. 200 mid-end EC2 nodes are used for our MapReduce job, and the job completes in 3 hours, at a cost of \$420. The MapReduce result is 198 GB in size, and it contains 3,636,263 profiles for Wikipedia entities and 313,156 profiles for Verbsense entities.

The categorization shown in Table 5 is based on the structure of the graphs. For each graph the set of possible attribute-values and roles are provided, although in practice we do not compute all of the possible combinations. Based on the structure of the problem, we pre-compute the ones which are more likely to make improvements in the task.

## 4 Applications

In this section we evaluate the data gathered based on knowledge schemas on two applications. The knowledge schemas

are expected to be applicable to any problem and are not limited to the two applications we introduce below.

### 4.1 Pattern discovery inside profiles

With the knowledge schemas extracted from big corpora, we expect to discover meaningful regularities in the data. For example, names of athletes tend to appear in similar context, although they might be playing on different teams or at different positions. In other words, the profiles of athletes (or at least some of the schemas) are expected to be similar. With this in mind, we expect to be able to distinguish the *occupation* of people based on some aspects of their schemas. For example, sample statistics of Tom Brady (football player) are shown in Table 6. Many of the context words such as “pass”, “throw”, etc represent the profession component of its profile. Similarly, Nikola Tesla’s profile partly reflects his major activities. Given such information, we can distinguish between people with different occupations from each other without the need of any test data specific training data. This has close connections with the *Dataless Classification* paradigm which has recently gained some popularity [Chang *et al.*, 2008; Song and Roth, 2014]. In addition, the aggregated profile of all football players should be closer to individual football players than that of entities with other occupations.

In order to test our hypothesis we create a dataset of people-professions. First we prepare a list of people, each with his/her name, Wiki url, and profession from Wikipedia<sup>5</sup>. We use this list as starting point to extract profiles of entities. We have gathered a list of entities  $\mathcal{E} = \{e_1, \dots, e_n\}$  such that for each  $e$  we know its profession. For a given entity  $e \in \mathcal{E}$ ,  $v(e)$  represents the vector of statistics given in the profile of

<sup>5</sup>Extracted from [http://en.wikipedia.org/wiki/Lists\\_of\\_people\\_by\\_occupation](http://en.wikipedia.org/wiki/Lists_of_people_by_occupation) for 2 levels of depth. In all cases, we applied an NER filter to eliminate irrelevant profiles.

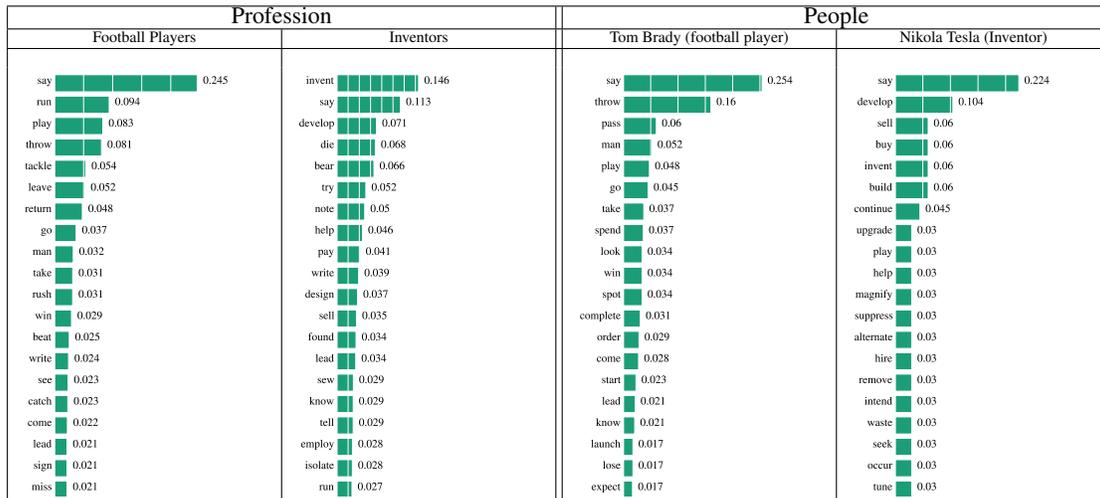


Table 6: Sample statistics in the profiles of two people and two occupations, retrieved from “Verb After” schema. The profiles of the occupations are created by averaging profile of many people with that occupation.

Dataset	Winograd	WinoCoref
Metric	Precision	AntePre
Rahman <i>et al</i> [2012]	73.05	—
Peng <i>et al</i> [2015]	76.41	89.32
Our paper	77.16	89.77

Table 7: Performance results on *Winograd* and *WinoCoref* datasets. With knowledge from our proposed schema, we get performance improvement compared to [Peng *et al*, 2015].

$e^6$ . Consider the set  $\mathcal{P} = \{p_1, \dots, p_m\}$  which contains the set of professions. The vector of statistics for each profession  $v(p_i)$  is the average of  $v(e_j) \in \mathcal{E}$  with the same profession. To make the experiment realistic we do 5 fold cross validation, i.e. when making prediction for some target people, the profiles of the occupations are created using the rest of them. To predict the occupation of an entity  $e_i$  we assign it to the profession with highest similarity measure. In our experiment, we use Okapi BM25 as our similarity measure, and in 72.1% of the test cases, the correct answer is among the top-5 predictions.

## 4.2 Improving Co-reference Resolution

To examine the power of our approach in a real NLP application, we choose to apply our schemas in co-reference resolution problems. Many hard co-reference resolution problems rely heavily on external knowledge [Rahman and Ng, 2011; Ratinov and Roth, 2012; Rahman and Ng, 2012; Peng *et al.*, 2015]. In particular, Winograd [1972] showed that small changes in context could completely change co-reference decisions. In the following examples, minor differences in otherwise identical sentences result in different references of the same pronoun.

Ex.1 The [ball] $_{e1}$  hit the [window] $_{e2}$  and Bill repaired [it] $_{pro}$ .

Ex.2 The [ball] $_{e1}$  hit the [window] $_{e2}$  and Bill caught [it] $_{pro}$ .

In Ex.1, if we know “repair window” is more likely than “repair ball”, we can decide that “it” refers to “window”. Likewise, in Ex.2, one needs to know “catch” is more likely

<sup>6</sup>Here we use only 4 schemas from Table 5: Nearest Noun After, Nearest Noun Before, Modifier Before, Nearest Verb After.

to be associated with “ball” than “window”. These references can be easily solved by humans, but are hard for today’s computer programs. However, the schemas we proposed are designed to capture this kind of knowledge<sup>7</sup>.

The *Winograd* dataset in [Rahman and Ng, 2012] contains 943 pairs of such sentences. It has a training set of 606 pairs and a testing set of 337 pairs. In each sentence, there are two entities and a pronoun, and we model it as a binary classification problem. Peng *et al* [2015] add more pronoun annotations to the dataset, model it as a general co-reference resolution problem and provide the *WinoCoref* dataset<sup>8</sup>. In this dataset, each co-referent cluster only contains 2-4 mentions and are all within the same sentence. We cannot use traditional co-reference metrics in this problem. Instead, we can view predicted co-reference clusters as binary decisions on each antecedent-pronoun pair (linked or not). Following Peng *et al* [2015], we compute the ratio of correct decisions over the total number of decisions made, and we call this metric *AntePre*.

We apply the knowledge extracted with our proposed schemas to the system described in Peng *et al* [2015] and test on both *Winograd* and *WinoCoref* datasets. The system uses Integer Linear Programming (ILP) formulation to solve co-reference problems, and it has a decision variable for each co-reference link. The knowledge from the schemas we proposed are automatically turned into constraints with tuned thresholds at the decision time. We implement the system the same way as described by Peng *et al* [2015]. By using our additional schemas as constraints, we improve the performance reported in [Peng *et al.*, 2015]. Results in Table 7 show that our schemas can be utilized as a reliable knowledge source in solving hard co-reference resolution problems. We expect

<sup>7</sup>In the PROFILER, we can tell that the number of co-occurrences for “catch” to be the nearest verb before “ball” is far larger than that for “catch” to be the nearest verb before “window”. Moreover, we also capture that the probability for “ball” to be object of “catch” is far larger than that for “window” to be object of “catch”.

<sup>8</sup>Available at: [http://cogcomp.cs.illinois.edu/page/resource\\_view/96](http://cogcomp.cs.illinois.edu/page/resource_view/96)

the knowledge we gathered can also be applied to other NLP applications that rely on external knowledge.

## 5 Concluding Remarks

We presented the PROFILER, a knowledge base created by large scale graph-based statistics extracted from huge corpora annotated using state-of-the-art NLP tools. We proposed a tree based formalization based on Feature Description Logic (FDL). The representation takes into account *typing* of concepts and entities, along with their *disambiguation*. We showed the application of our tool on dataless classification of people-occupations and hard co-reference resolution. More experiments need to be done to gain better insight into this knowledge representation, the effect of disambiguation, and other applications which can benefit from it.

Our current implementation includes a wide range of schemas which are efficiently processed on Amazon cloud service. However, this implementation can be generalized for on-demand extraction of desired schemas which is the subject of our future work.

## 6 Acknowledgment

The authors would like to thank Eric Horn, Alice Lai and Christos Christodoulopoulos for comments that helped to improve this work. This work is partly supported by DARPA under agreement number FA8750-13-2-0008 and by a grant from Allen Institute for Artificial Intelligence (AI2). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

## References

- [Baader *et al.*, 2008] F. Baader, I. Horrocks, and U. Sattler. Description logics. *Found. of Artificial Intelligence*, 3:135–179, 2008.
- [Banko *et al.*, 2007] M. Banko, M. Cafarella, M. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.
- [Bollacker *et al.*, 2008] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of ACM SIGMOD international conference on Management of data*, 2008.
- [Carlson *et al.*, 2010] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2010.
- [Chang *et al.*, 2008] M. Chang, L. Ratinov, D. Roth, and V. Srikumar. Importance of semantic representation: Dataless classification. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2008.
- [Clarke *et al.*, 2012] J. Clarke, V. Srikumar, M. Sammons, and D. Roth. An nlp curator (or: How i learned to stop worrying and love nlp pipelines). In *International Conference on Language Resources and Evaluation (LREC)*, 2012.
- [Cumby and Roth, 2003a] C. Cumby and D. Roth. On kernel methods for relational learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.
- [Cumby and Roth, 2003b] C M Cumby and D. Roth. Learning with feature description logics. In *Inductive Logic Programming*, pages 32–47. Springer, 2003.
- [Hoffart *et al.*, 2013] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia (extended abstract). In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.
- [Khardon *et al.*, 1999] R. Khardon, D. Roth, and L. G. Valiant. Relational learning for nlp using linear threshold elements. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 911–917, 1999.
- [Kingsbury and Palmer, 2002] P Kingsbury and M. Palmer. From treebank to propbank. In *International Conference on Language Resources and Evaluation (LREC)*, 2002.
- [Levesque and Brachman, 1987] H. J. Levesque and R. J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational intelligence*, 3(1):78–93, 1987.
- [Peng *et al.*, 2015] H. Peng, D. Khashabi, and D. Roth. Solving hard coreference problems. In *Proceedings of the Conference of the North American Chapter of ACL (NAACL)*, 2015.
- [Rahman and Ng, 2011] A. Rahman and V. Ng. Coreference resolution with world knowledge. In *Proceedings of the Annual Meeting of ACL (NAACL)*. Association for Computational Linguistics, 2011.
- [Rahman and Ng, 2012] A. Rahman and V. Ng. Resolving complex cases of definite pronouns: the winograd schema challenge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 777–789, 2012.
- [Ratinov and Roth, 2012] L. Ratinov and D. Roth. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2012.
- [Song and Roth, 2014] Y. Song and D. Roth. On dataless hierarchical text classification. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2014.
- [Winograd, 1972] T. Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.
- [Wu *et al.*, 2014] H. Wu, Z. Fei, A. Dai, S. Mayhew, M. Sammons, and Roth D. Illinoiscloudnlp: Text analytics services in the cloud. In *International Conference on Language Resources and Evaluation(LREC)*, 5 2014.