# Fine-grained Semantic Typing of Emerging Entities

**Ndapandula Nakashole, Tomasz Tylenda, Gerhard Weikum**
Max Planck Institute for Informatics
Saarbrücken, Germany
{nnakasho,ttylenda,weikum}@mpi-inf.mpg.de

## Abstract

Methods for information extraction (IE) and knowledge base (KB) construction have been intensively studied. However, a largely underexplored case is tapping into highly dynamic sources like news streams and social media, where new entities are continuously emerging. In this paper, we present a method for discovering and semantically typing newly emerging out-of-KB entities, thus improving the freshness and recall of ontology-based IE and improving the precision and semantic rigor of open IE. Our method is based on a probabilistic model that feeds weights into integer linear programs that leverage type signatures of relational phrases and type correlation or disjointness constraints. Our experimental evaluation, based on crowdsourced user studies, show our method performing significantly better than prior work.

## 1 Introduction

A large number of knowledge base (KB) construction projects have recently emerged. Prominent examples include Freebase (Bollacker 2008) which powers the Google Knowledge Graph, ConceptNet (Havasi 2007), YAGO (Suchanek 2007), and others. These KBs contain many millions of entities, organized in hundreds to hundred thousands of semantic classes, and hundred millions of relational facts between entities. However, despite these impressive advances, there are still major limitations regarding coverage and freshness. Most KB projects focus on entities that appear in Wikipedia (or other reference collections such as IMDB), and very few have tried to gather entities "in the long tail" beyond prominent sources. Virtually all projects miss out on newly emerging entities that appear only in the latest news or social media. For example, the Greenlandic singer Nive Nielsen has gained attention only recently and is not included in any KB (a former Wikipedia article was removed because it "does not indicate the importance or significance of the subject"), and the resignation of BBC director Entwistle is a recently new entity (of type event).

**Goal.** Our goal in this paper is to discover emerging entities of this kind on the fly as they become noteworthy in news and social-media streams. A similar theme is pursued in research on *open information extraction (open IE)* (Banko 2007; Fader 2011; Talukdar 2010; Venetis 2011; Wu 2012), which yields higher recall compared to ontology-style KB construction with canonicalized and semantically typed entities organized in prespecified classes. However, state-of-the-art open IE methods extract all noun phrases that are likely to denote entities. These phrases are not canonicalized, so the same entity may appear under many different names, e.g., "Mr. Entwistle", "George Entwistle", "the BBC director", "BBC head Entwistle", and so on. This is a problem because names and titles are ambiguous, and this hampers precise search and concise results.

Our aim is for all recognized and newly discovered entities to be semantically interpretable by having *fine-grained types* that connect them to KB classes. The expectation is that this will boost the disambiguation of known entity names and the grouping of new entities, and will also strengthen the extraction of relational facts about entities. For informative knowledge, new entities must be typed in a fine-grained manner (e.g., guitar player, blues band, concert, as opposed to crude types like person, organization, event).

Strictly speaking, the new entities that we capture are *typed noun phrases*. We do not attempt any cross-document co-reference resolution, as this would hardly work with the long-tail nature and sparse observations of emerging entities. Therefore, our setting resembles the established task of fine-grained typing for noun phrases (Fleischmann 2002), with the difference being that we disregard common nouns and phrases for prominent in-KB entities and instead exclusively focus on the difficult case of phrases that likely denote new entities. The baselines to which we compare our method are state-of-the-art methods for noun-phrase typing (Lin 2012; Yosef 2012).

**Contribution.** The solution presented in this paper, called PEARL, leverages a repository of relational patterns that are organized in a type-signature taxonomy. More specifically, we harness the PATTY collection consisting of more than 300,000 typed paraphrases (Nakashole 2012). An example of PATTY's expressive phrases is: $\langle$*musician*$\rangle$ * *cover* * $\langle$*song*$\rangle$ for a musician performing someone else's song. When extracting noun phrases, PEARL also collects the co-occurring PATTY phrases. The type signatures of the relational phrases are cues for the type of the entity denoted by the noun phrase. For example, an entity named Snoop Dogg that frequently co-occurs with the $\langle$*singer*$\rangle$ * *distinctive voice in* * $\langle$*song*$\rangle$ pattern is likely to be a singer. Moreover, if one entity in a relational triple is in the KB and can be properly disambiguated (e.g., a singer), we can use a partially bound pattern to infer the type of the other entity (e.g., a song) with higher confidence.

In this line of reasoning, we also leverage the common situation that many input sentences contain one entity registered in the KB and one novel or unknown entity. Known entities are recognized and mapped to the KB using a recent tool for named entity disambiguation (Hoffart 2011). For cleaning out false hypotheses among the type candidates for a new entity, we devised probabilistic models and an integer linear program that considers incompatibilities and correlations among entity types.

In summary, our contribution in this paper is a model for discovering and ontologically typing out-of-KB entities, using a fine-grained type system and harnessing relational paraphrases with type signatures for probabilistic weight computation. Crowdsourced quality assessments demonstrate the accuracy of our model.

## 2 Detection of New Entities

To detect noun phrases that potentially refer to entities, we apply a part-of-speech tagger to the input text. For a given noun phrase, there are four possibilities: *a)* The noun phrase refers to a general concept (a class or abstract concept), not an individual entity. *b)* The noun phrase is a known entity that can be directly mapped to the knowledge base. *c)* The noun phrase is a new name for a known entity. *d)* The noun phrase is a new entity not known to the knowledge base at all. In this paper, our focus is on case d); all other cases are out of the scope of this paper.

We use an extensive dictionary of surface forms for in-KB entities (Hoffart 2012), to determine if a name or phrase refers to a known entity. If a phrase does not have any match in the dictionary, we assume that it refers to a new entity. To decide if a noun phrase is a true entity (i.e., an individual entity that is a member of one or more lexical classes) or a non-entity (i.e., a common noun phrase that denotes a class or a general concept), we base the decision on the following hypothesis (inspired by and generalizing (Bunescu 2006): A given noun phrase, not known to the knowledge base, is a true entity if its headword is singular and is consistently capitalized (i.e., always spelled with the first letter in upper case).

## 3 Typing Emerging Entities

To deduce types for new entities we propose to align new entities along the type signatures of patterns they occur with. In this manner we use the patterns to suggest types for the entities they occur with. In particular, we infer entity types from pattern type signatures. Our approach builds on the following hypothesis:

**Hypothesis 3.1 (Type Alignment Hypothesis)** *For a given pattern such as $\langle$actor$\rangle$'s character in $\langle$movie$\rangle$, we assume that an entity pair $(x, y)$ frequently occurring with the pattern in text implies that $x$ and $y$ are of the types $\langle$actor$\rangle$ and $\langle$movie$\rangle$, respectively.*

**Challenges and Objective.** While the type alignment hypothesis works as a starting point, it introduces false positives. Such false positives stem from the challenges of polysemy, fuzzy pattern matches, and incorrect paths between entities. With polysemy, the same lexico-syntactic pattern can have different type signatures. For example, the following are three different patterns: $\langle$*singer*$\rangle$ *released* $\langle$*album*$\rangle$, $\langle$*music_band*$\rangle$ *released* $\langle$*album*$\rangle$, $\langle$*company*$\rangle$ *released* $\langle$*product*$\rangle$. For an entity pair $(x, y)$ occurring with the pattern "released", $x$ can be one of three different types.

We cannot expect that the phrases we extract in text will be exact matches of the typed relational patterns learned by PATTY. Therefore, for better recall, we must accept fuzzy matches. Quite often however, the extracted phrase matches multiple relational patterns to various degrees. Each of the matched relational patterns has its own type signature. The type signatures of the various matched patterns can be incompatible with one another.

The problem of incorrect paths between entities emerges when a pair of entities occurring in the same sentence do not stand in a true subject-object relation. Dependency parsing does not adequately solve the issue. Web sources contain a plethora of sentences that are not well-formed. Such sentences mislead the dependency parser to extract wrong dependencies.

Our solution takes into account polysemy, fuzzy matches, as well as issues stemming from potential incorrect-path limitations. We define and solve the following optimization problem:

**Definition 1 (Type Inference Optimization)** *Given all the candidate types for $x$, find the best types or "strongly supported" types for $x$. The final solution must satisfy type disjointness constraints. Type disjointness constraints are constraints that indicate that, semantically, a pair of types cannot apply to the same entity at the same time. For example, a $\langle$university$\rangle$ cannot be a $\langle$person$\rangle$.*

We also study a relaxation of type disjointness constraints through the use of type correlation constraints. Our task is therefore twofold: first, generate candidate types for new entities; second, find the best types for each new entity among its candidate types.

## 4 Candidate Types for Entities

For a given entity, candidate types are types that can potentially be assigned to that entity, based on the entity's co-occurrences with typed relational patterns.

**Definition 2 (Candidate Type)** *Given a new entity $x$ which occurs with a number of patterns $p_1, p_2, ..., p_n$, where each pattern $p_i$ has a type signature with a domain and a range: if $x$ occurs on the left of $p_i$, we pick the domain of $p_i$ as a candidate type for $x$; if $x$ occurs on the right of $p_i$, we pick the range of $p_i$ as a candidate type for $x$.*

For each candidate type, we compute confidence weights. Ideally, if an entity occurs with a pattern which is highly specific to a given type then the candidate type should have high confidence. For example "is married to" is more specific to people then "expelled from". A *person* can be expelled from an *organization* but a *country* can also be expelled from an *organization* such as NATO. There are various ways to compute weights for *candidate types*. We first introduce a uniform weight approach and then present a method for computing more informative weights.

### 4.1 Uniform Weights

We are given a new entity $x$ which occurs with phrases $(x\ phrase_1\ y_1)$, $(x\ phrase_2\ y_2)$, ..., $(x\ phrase_n\ y_n)$. Suppose these occurrences lead to the facts $(x, p_1, y_1)$, $(x, p_2, y_2)$,..., $(x, p_n, y_n)$. The $p_is$ are the *typed relational patterns* extracted by PATTY. The facts are generated by matching $phrases$ to relational patterns with type signatures. The type signature of a pattern is denoted by:

$$sig(p_i) = (domain(p_i), range(p_i))$$

We allow fuzzy matches, hence each fact comes with a match score. This is the similarity degree between the phrase observed in text and the typed relational pattern.

**Definition 3 (Fuzzy Match Score)** *Suppose we observe the surface string:* $(x\ phrase\ y)$ *which leads to the fact:* $x, p_i, y$. *The fuzzy match similarity score is:* $sim(phrase, p_i)$, *where similarity is the n-gram Jaccard similarity between the phrase and the typed pattern.*

The confidence that $x$ is of type $domain$ is defined as follows:

**Definition 4 (Candidate Type Confidence)** *For a given observation* $(x\ phrase\ y)$, *where phrase matches patterns* $p_1, ..., p_n$, *with domains* $d_1, ..., d_b$ *which are possibly the same:*

$$typeConf(x, phrase, d) = \sum_{\{p_i:domain(p_i)=d\}} \Big(sim(phrase, p_i)\Big)$$

*Observe that this sums up over all patterns that match the phrase.*

To compute the final confidence for $typeConf(x, domain)$, we aggregate the confidences over all $phrases$ occurring with $x$.

**Definition 5 (Aggregate Confidence)** *For a set of observations* $(x, phrase_1, y_1)$, $(x, phrase_2, y_2)$, *...,* $(x, phrase_n, y_n)$, *the aggregate candidate type confidence is given by:*

$$\begin{aligned} &aggTypeConf(x, d) = \\ &\quad \sum_{phrase_i} typeConf(x, phrase_i, d) \\ &= \sum_{phrase_i} \sum_{\{p_j:domain(p_j)=d\}} (sim(phrase_i, p_j)) \end{aligned}$$

The confidence for the range $typeConf(x, range)$ is computed analogously. All confidence weights are normalized to values in $[0, 1]$.

The limitation of the uniform weight approach is that each pattern is considered equally good for suggesting *candidate types*. Thus this approach does not take into account the intuition that an entity occurring with a pattern which is highly specific to a given type is a stronger signal that the entity is of the type suggested. Our next approach addresses this limitation.

### 4.2 Co-occurrence Likelihood Weight Computation

We devise a likelihood model for computing weights for entity *candidate types*. Central to this model is the estimation of the likelihood of a given type occurring with a given pattern.

Suppose using PATTY methods we mined a typed relational pattern $\langle t_1 \rangle\ p\ \langle t_2 \rangle$. Suppose that we now encounter a new entity pair $(x, y)$ occurring with a *phrase* that matches $p$. We can compute the likelihood of $x$ and $y$ being of types $t_1$ and $t_2$, respectively, from the likelihood of $p$ co-occurring with entities of types $t_1, t_2$. Therefore we are interested in the type-pattern likelihood, defined as follows:

**Definition 6 (Type-Pattern Likelihood)** *The likelihood of p co-occurring with an entity pair* $(x, y)$ *of the types* $(t_1, t_2)$ *is given by:*

$$P[t_1, t_2|p] \qquad (1)$$

*where* $t_1$ *and* $t_2$ *are the types of the arguments observed with p from a corpus such as Wikipedia.* $P[t_1, t_2|p]$ *is expanded as follows:*

$$P[t_1, t_2|p] = \frac{P[t_1, t_2, p]}{P[p]}. \qquad (2)$$

The expressions on the right-hand side of Equation 2 can be directly estimated from a corpus. We use Wikipedia (English), for corpus-based estimations. $P[t_1, t_2, p]$ is the relative occurrence frequency of the typed pattern among all entity-pattern-entity triples in a corpus (e.g., the fraction of ⟨*musican*⟩ *plays* ⟨*song*⟩ among all triples). P[p] is the relative occurrence frequency of the untyped pattern (e.g., plays) regardless of the argument types. For example, this sums up over both ⟨*musican*⟩ *plays* ⟨*song*⟩ occurrences and ⟨*actor*⟩ *plays* ⟨*fictional character*⟩. If we observe a fact where one argument name can be easily disambiguated to a knowledge-base entity so that its type is known, and the other argument is considered to be an out-of-knowledge-base entity, we condition the joint probability of $t_1$, $p$, and $t_2$ in a different way:

**Definition 7 (Conditional Type-PatternLikelihood)** *The likelihood of an entity of type* $t_1$ *occurring with* *a pattern p and an entity of type $t_2$ is given by:*

$$P[t_1|t_2,p] = \frac{P[t_1,t_2,p]}{P[p,t_2]} \qquad (3)$$

*where the $P[p,t_2]$ is the relative occurrence frequency of a partial triple, for example, ⟨*⟩ plays ⟨song⟩.*

Observe that all numbers refer to occurrence frequencies. For example, $P[t_1,\ p,\ t_2]$ is a fraction of the total number of triples in a corpus.

Multiple patterns can suggest the same type for an entity. Therefore, the weight of the assertion that $y$ is of type $t$, is the total support strength from all phrases that suggest type $t$ for $y$.

**Definition 8 (Aggregate Likelihood)** *The aggregate likelihood candidate type confidence is given by:*

$$typeConf(x, domain)) = \sum_{phrase_i} \sum_{p_j} \Big( sim(phrase_i, p_j) * \Upsilon \Big)$$

*Where $\Upsilon = P[t_1,t_2|p]$ or $P[t_1|t_2,p]$ or $P[t_2|t_1,p]$*

The confidence weights are normalized to values in $[0,1]$. So far we have presented a way of generating a number of *weighted candidate types* for $x$. In the next step we pick the best types for an entity among all its candidate types.

### 4.3 Integer Linear Program Formulation

Given a set of *weighted candidate types*, our goal is to pick a compatible subset of types for $x$. The additional asset that we leverage here is the compatibility of types: how likely is it that an entity belongs to both type $t_i$ and type $t_j$. Some types are mutually exclusive, for example, the type *location* rules out *person* and, at finer levels, *city* rules out *river* and *building*, and so on. Our approach harnesses these kinds of constraints. Our solution is formalized as an Integer Linear Program (ILP). We have candidate types for $x$: $t_1, .., t_n$. First, we define a decision variable $T_i$ for each candidate type $i = 1, \ldots, n$. These are binary variables: $T_i = 1$ means type $t_i$ is selected to be included in the set of types for $x$, $T_i = 0$ means we discard type $t_i$ for $x$.

In the following we develop two variants of this approach: a "hard" ILP with rigorous disjointness constraints, and a "soft" ILP which considers type correlations.

**"Hard" ILP with Type Disjointness Constraints.** We infer type disjointness constraints from the YAGO2 knowledge base using occurrence statistics. Types with no overlap in entities or insignificant overlap below a specified threshold are considered disjoint. Notice that this introduces *hard constraints* whereby selecting one type of a disjoint pair rules out the second type. We define type disjointness constraints $T_i + T_j \le 1$ for all disjoint pairs $t_i, t_j$ (e.g. person-artifact, movie-book, city-country, etc.). The ILP is defined as follows:

**objective**

max $\sum_i T_i \times w_i$

**type disjointness constraint**

$\forall (t_i, t_j)_{disjoint}\ \ T_i + T_j \le 1$

The weights $w_i$ are the aggregrated likelihoods as specified in Definition 8.

**"Soft" ILP with Type Correlations.** In many cases, two types are not really mutually exclusive in the strict sense, but the likelihood that an entity belongs to both types is very low. For example, few drummers are also singers. Conversely, certain type combinations are boosted if they are strongly correlated. An example is guitar players and electric guitar players. Our second ILP considers such soft constraints. To this end, we pre-compute *Pearson correlation* coefficients for all type pairs $(t_i, t_j)$ based on co-occurrences of types for the same entities. These values $v_{ij} \in [-1, 1]$ are used as weights in the objective function of the ILP. We additionally introduce pair-wise decision variables $Y_{ij}$, set to 1 if the entity at hand belongs to both types $t_i$ and $t_j$, and 0 otherwise. This coupling between the $Y_{ij}$ variables and the $T_i, T_j$ variables is enforced by specific constraints. For the objective function, we choose a linear combination of per-type evidence, using weights $w_i$ as before, and the type-compatibility measure, using weights $v_{ij}$. The ILP with correlations is defined as follows:

**objective**

max $\alpha \sum_i T_i \times w_i + (1-\alpha) \sum_{ij} Y_{ij} \times v_{ij}$

**type correlation constraints**

$\forall_{i,j}\ \ Y_{ij} + 1 \geq T_i + T_j$

$\forall_{i,j}\ \ Y_{ij} \leq T_i$

$\forall_{i,j}\ \ Y_{ij} \leq T_j$

Note that both ILP variants need to be solved per entity, not over all entities together. The "soft" ILP has a size quadratic in the number of candidate types, but this is still a tractable input for modern solvers. We use the Gurobi software package to compute the solutions for the ILP's. With this design, PEARL can efficiently handle a typical news article in less than a second, and is well geared for keeping up with high-rate content streams in real time. For both the "hard" and "soft" variants of the ILP, the solution is the best types for entity $x$ satisfying the constraints.

## 5 Evaluation

To define a suitable corpus of test data, we obtained a stream of news documents by subscribing to *Google News* RSS feeds for a few topics over a six-month period (April 2012 – September 2012). This produced $318,434$ documents. The topics we subscribed to are: *Angela Merkel, Barack Obama, Business, Entertainment, Hillary Clinton, Joe Biden, Mitt Romney, Newt Gingrich, Rick Santorum, SciTech and Top News*. All our experiments were carried out on this data. The type system used is that of YAGO2, which is derived from WordNet. Human evaluations were carried out on Amazon Mechanical Turk (MTurk), which is a platform for crowd-sourcing tasks that require human input. Tasks on MTurk are small questionnaires consisting of a description and a set of questions.

**Baselines.** We compared PEARL against two state-of-the-art baselines: **i). NNPLB** (No Noun Phrase Left Behind), is the method presented in (Lin 2012), based on the propagation of types for known entities through salient patterns occurring with both known and unknown entities. We implemented the algorithm in (Lin 2012) in our framework, using the relational patterns of PATTY (Nakashole 2012) for comparability. For assessment we sampled from the top-5 highest ranked types for each entity. In our experiments, our implementation of NNPLB achieved precision values comparable to those reported in (Lin 2012). **ii). HYENA** (Hierarchical tYpe classification for Entity NAmes), the method of (Yosef 2012), based on a feature-rich classifier for fine-grained, hierarchical type tagging. This is a state-of-the-art representative of similar methods such as (Rahman 2010; Ling 2012).

**Evaluation Task.** To evaluate the quality of types assigned to emerging entities, we presented turkers with sentences from the news tagged with out-of-KB entities and the types inferred by the methods under test. The turkers task was to assess the correctness of types assigned to an entity mention. To make it easy to understand the task for the turkers, we combined the extracted entity and type into a sentence. For example if PEARL inferred that Brussels Summit is an political event, we generate and present the sentence: *Brussels Summit is an event*. We allowed four possible assessment values: *a) Very good* output corresponds to a perfect result. *b) Good* output exhibits minor errors. For instance, the description *G20 Summit is an organization* is wrong, because the summit is an event, but G20 is indeed an organization. The problem in this example is incorrect segmentation of a named entity. *c) Wrong* for incorrect types (e.g., *Brussels Summit is a politician*). *d) Not sure / do not know* for other cases.

**Comparing PEARL to Baselines.** Per method, turkers evaluated 105 entity-type pair test samples. We first sampled among out-of-KB entities that were mentioned frequently in the news corpus: in at least 20 different news articles. Each test sample was given to 3 different turkers for assessment. Since the turkers did not always agree if the type for a sample is good or not, we aggregate their answers. We use voting to decide whether the type was assigned correctly to an entity. We consider the following voting variants: i) majority "very good" or "good", a conservative notion of precision: precision$_{lower}$. ii) at least one "very good" or "good", a liberal notion of precision: precision$_{upper}$. Table 1 shows precision for PEARL-hard, PEARL-soft, NNPLB, and HYENA, with a 0.9-confidence Wilson score interval (Brown 2001). PEARL-hard outperformed

| | Precision$_{lower}$ | Precision$_{upper}$ |
|---|---|---|
| PEARL-hard | **0.77**±0.08 | **0.88**±0.06 |
| PEARL-soft | 0.53±0.09 | 0.77±0.09 |
| HYENA | 0.26±0.08 | 0.56±0.09 |
| NNPLB | 0.46±0.09 | 0.68±0.09 |

Table 1: Comparison of PEARL to baselines.

| $\kappa$ | *Fleiss* | *Cohen* |
|---|---|---|
| | 0.34 | 0.45 |

Table 2: Lower bound estimations for inter-judge agreement kappa: Fleiss' $\kappa$ & adapted Cohen's $\kappa$.

PEARL-soft and also both baselines. HYENA's relatively poor performance can be attributed to the fact that its features are mainly syntactic such as bigrams and part-of-speech tags. Web data is challenging, it has a lot of variations in syntactic formulations. This introduces a fair amount of ambiguity which can easily mislead syntactic features. Leveraging semantic features as done by PEARL could improve HYENA's performance. While the NNPLB method performs better than HYENA, in comparison to PEARL-hard, there is room for improvement. Like HYENA, NNPLB assigns negatively correlated types to the same entity. This limitation could be addressed by applying PEARL's ILPs and probabilistic weights to the candidate types suggested by NNPLB.

To compute inter-judge agreement we calculated Fleiss' kappa and Cohen's kappa $\kappa$, which are standard measures. The usual assumption for Fleiss'$\kappa$ is that labels are categorical, so that each disagreement counts the same. This is not the case in our settings, where different labels may indicate partial agreement ("good", "very good"). Therefore the $\kappa$ values in Table 2 are lower-bound estimates of agreement in our experiments; the "true agreement" seems higher. Nevertheless, the observed Fleiss $\kappa$ values show that the task was fairly clear to the turkers; values $> 0.2$ are generally considered as acceptable (Landis 1977). Cohen's $\kappa$ is also not directly applicable to our setting. We approximated it by finding pairs of judges who assessed a significant number of the same entity-type pairs.

| | Precision$_{lower}$ | Precision$_{upper}$ |
|---|---|---|
| Freq. mentions | 0.77±0.08 | 0.88±0.06 |
| All mentions | 0.65±0.09 | 0.77±0.08 |

Table 3: PEARL-hard performance on a sample of frequent entities (mention frequency$\geq$ 20) and on a sample of entities of all mention frequencies.

**Mention Frequencies.** We also studied PEARL-hard's performance on entities of different mention frequencies. The results are shown in Table 3. Frequently mentioned entities provide PEARL with more evidence as they potentially occur with more patterns. Therefore, as expected, precision when sampling over all entities drops a bit. For such infrequent entities, PEARL does not have enough evidence for reliable type assignments.

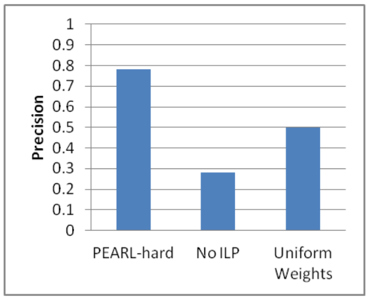

Figure 1: Variations of the PEARL method.

**Variations of PEARL.** To quantify how various aspects of our approach affect performance, we studied a few variations. The first method is the full PEARL-hard. The second method is PEARL with no ILP (denoted No ILP), only using the probabilistic model. The third variation is PEARL without probabilistic weights (denoted Uniform Weights). From Figure 1, it is clear that both the ILP and the weighting model contribute significantly to PEARL's ability to make precise type assignments. Sample results from PEARL-hard are shown in Table 4.

**NDCG.** For a given entity mention $e$, an entity-typing system returns a ranked list of types $\{t_1, t_2, ..., t_n\}$. We evaluated ranking quality using the top-5 ranks for each method. These assessments were aggregated into the normalized discounted cumulative gain (NDCG), a widely used measure for

| Entity | Inferred Type | Sample Source Sentence (s) |
|---|---|---|
| Lochte | medalist | **Lochte** won America's lone gold ... |
| Malick | director | ... the red carpet in Cannes for **Malick**'s 2011 movie ... |
| Bonamassa | musician | **Bonamassa** recorded Driving Towards the Daylight in Las Vegas ...<br>... **Bonamassa** opened for B.B. King in Rochester , N.Y. |
| Analog Man | album | Analog Man is Joe Walsh's first solo album in 20 years. |
| Melinda Liu | journalist | ... in a telephone interview with journalist **Melinda Liu** of the Daily Beast. |
| RealtyTrac | publication | Earlier this month, **RealtyTrac** reported that ... |

Table 4: Sample types inferred by PEARL.

ranking quality. The NDCG values obtained are 0.53, 0.16, and 0.16, for PEARL-hard, HYENA, and NNPLB, respectively. PEARL clearly outperforms the baselines on ranking quality, too.

## 6 Related Work

Tagging mentions of named entities with lexical types has been pursued in previous work. Most well-known is the Stanford named entity recognition (NER) tagger (Finkel 2005) which assigns coarse-grained types like person, organization, location, and other to noun phrases that are likely to denote entities. There is fairly little work on fine-grained typing, notable results being (Fleischmann 2002; Rahman 2010; Ling 2012; Yosef 2012). These methods consider type taxonomies similar to the one used for PEARL, consisting of several hundreds of fine-grained types. All methods use trained classifiers over a variety of linguistic features, most importantly, words and bigrams with part-of-speech tags in a mention and in the textual context preceding and following the mention. In addition, the method of (Yosef 2012) (HYENA) utilizes a big gazetteer of per-type words that occur in Wikipedia anchor texts. This method outperforms earlier techniques on a variety of test cases; hence it served as one of our baselines.

Closely related to our work is the recent approach of (Lin 2012) (NNPLB) for predicting types for out-of-KB entities. Noun phrases in the subject role in a large collection of fact triples are heuristically linked to Freebase entities. This yields type information for the linked mentions. For unlinkable entities the NNPLB method (inspired by (Kozareva 2011)) picks types based on co-occurrence with salient relational patterns by propagating types of linked entities to unlinkable entities that occur with the same patterns. Unlike PEARL, NNPLB does not attempt to resolve inconsistencies among the predicted types. In contrast, PEARL uses an ILP with type disjointness and correlation constraints to solve and penalize such inconsistencies. NNPLB uses untyped patterns, whereas PEARL harnesses patterns with type signatures. Furthermore, PEARL computes weights for candidate types based on patterns and type signatures. Weight computations in NNPLB are only based on patterns. NNPLB only assigns types to entities that appear in the subject role of a pattern. This means that entities in the object role are not typed at all. In contrast, PEARL infers types for entities in both the subject and object role.

Type disjointness constraints have been studied for other tasks in information extraction (Carlson 2010; Suchanek 2009), but using different formulations.

## 7 Conclusion

This paper addressed the problem of detecting and semantically typing newly emerging entities, to support the life-cycle of large knowledge bases. Our solution, PEARL, draws on a collection of semantically typed patterns for binary relations. PEARL feeds probabilistic evidence derived from occurrences of such patterns into two kinds of ILPs, considering type disjointness or type correlations. This leads to highly accurate type predictions, significantly better than previous methods, as our crowdsourcing-based evaluation showed.

## References

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z.G. Ives: DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, pages 722–735, Busan, Korea, 2007.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni: Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676, Hyderabad, India, 2007.

K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor: Freebase: a Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages, 1247-1250, Vancouver, BC, Canada, 2008.

Lawrence D. Brown, T.Tony Cai, Anirban Dasgupta: Interval Estimation for a Binomial Proportion. Statistical Science 16: pages 101–133, 2001.

R. C. Bunescu, M. Pasca: Using Encyclopedic Knowledge for Named entity Disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy, 2006.

A. Carlson, J. Betteridge, R.C. Wang, E.R. Hruschka, T.M. Mitchell: Coupled Semi-supervised Learning for Information Extraction. In *Proceedings of the Third International Conference on Web Search and Web Data Mining (WSDM)*, pages 101–110, New York, NY, USA, 2010.

S. Cucerzan: Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, 2007.

A. Fader, S. Soderland, O. Etzioni: Identifying Relations for Open Information Extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545, Edinburgh, UK, 2011.

J.R. Finkel, T. Grenager, C. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 363–370, Ann Arbor, Michigan, 2005.

Michael Fleischman, Eduard H. Hovy: Fine Grained Classification of Named Entities. In *Proceedings the International Conference on Computational Linguistics*, COLING 2002.

X. Han, J. Zhao: Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge. In *Proceedings of 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 215 – 224,Hong Kong, China, 2009.

C. Havasi, R. Speer, J. Alonso. ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge. In *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, 2007.

Sebastian Hellmann, Claus Stadler, Jens Lehmann, Sren Auer: DBpedia Live Extraction. OTM Conferences (2) 2009: 1209-1223.

J. Hoffart, M. A. Yosef, I.Bordino and H. Fuerstenau, M. Pinkal, M. Spaniol, B.Taneva, S.Thater, Gerhard Weikum: Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 782–792, Edinburgh, UK, 2011.

J. Hoffart, F. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, G. Weikum: YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*, pages 229–232, Hyderabad, India. 2011.

J. Hoffart, F. Suchanek, K. Berberich, G. Weikum: YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. Artificial Intelligence 2012.

Z. Kozareva, L. Voevodski, S.-H.Teng: Class Label Enhancement via Related Instances. EMNLP 2011: 118-128

J. R. Landis, G. G. Koch: The measurement of observer agreement for categorical data in Biometrics. Vol. 33, pp. 159174, 1977.

C. Lee, Y-G. Hwang, M.-G. Jang: Fine-grained Named Entity Recognition and Relation Extraction for Question Answering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 799–800, Amsterdam, The Netherlands, 2007.

T. Lin, Mausam , O. Etzioni: No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 893–903, Jeju, South Korea, 2012.

Xiao Ling, Daniel S. Weld: Fine-Grained Entity Recognition. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2012

D. N. Milne, I. H. Witten: Learning to Link with Wikipedia. In *Proceedings of 17th ACM Conference on Information and Knowledge Management (CIKM)*, pages 509-518, Napa Valley, California, USA, 2008.

N. Nakashole, G. Weikum, F. Suchanek: PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1135 -1145, Jeju, South Korea, 2012.

V. Nastase, M. Strube, B. Boerschinger, Cäcilia Zirn, Anas Elghafari: WikiNet: A Very Large Scale Multi-Lingual Concept Network. In *Proceedings of the 7th International Conference on Language Resources and Evaluation(LREC)*, Malta, 2010.

H. T. Nguyen, T. H. Cao: Named Entity Disambiguation on an Ontology Enriched by Wikipedia. In *Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future in Computing & Communication Technologies (RIVF)*, pages 247–254, Ho Chi Minh City, Vietnam, 2008.

Feng Niu, Ce Zhang, Christopher Re, Jude W. Shavlik: DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. In the VLDS Workshop, pages 25-28, 2012.

A. Rahman, Vincent Ng: Inducing Fine-Grained Semantic Classes via Hierarchical and Collective Classification. In *Proceedings the International Conference on Computational Linguistics (COLING)*, pages 931-939, 2010.

F. M. Suchanek, G. Kasneci, G. Weikum: Yago: a Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web (WWW)* pages, 697-706, Banff, Alberta, Canada, 2007.

F. M. Suchanek, M. Sozio, G. Weikum: SOFIE: A Self-organizing Framework for Information Extraction. In*Proceedings of the 18th International Conference on World Wide Web (WWW)*, pages 631–640, Madrid, Spain, 2009.

P.P. Talukdar, F. Pereira: Experiments in Graph-Based Semi-Supervised Learning Methods for Class-Instance Acquisition. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1473-1481, 2010.

P. Venetis, A. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, C. Wu: Recovering Semantics of Tables on the Web. In *Proceedings of the VLDB Endowment*, PVLDB 4(9), pages, 528–538. 2011.

W. Wu, H. Li, H. Wang, K. Zhu: Probase: A Probabilistic Taxonomy for Text Understanding. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 481–492, Scottsdale, AZ, USA, 2012.

M. A. Yosef, S. Bauer, J. Hoffart, M. Spaniol, G. Weikum: HYENA: Hierarchical Type Classification for Entity Names. In *Proceedings the International Conference on Computational Linguistics(COLING), to appear*, 2012.