Universal Schema for Entity Type Prediction

Limin Yao  
Dept. of Computer Science  
University of Massachusetts  
Amherst, MA, United States  
lmyao@cs.umass.edu

Sebastian Riedel  
Dept. of Computer Science  
University College London  
London, United Kingdom  
s.riedel@ucl.ac.uk

Andrew McCallum  
Dept. of Computer Science  
University of Massachusetts  
Amherst, MA, United States  
mccallum@cs.umass.edu

ABSTRACT
Categorizing entities by their type is useful in many applications, such as knowledge base construction, relation extraction and query intent prediction. Fine-grained entity type ontologies are especially valuable, but typically difficult to design because of endless quandaries about level of detail and boundary cases. Automatically classifying entities by type is challenging as well, usually involving hand-labeling data and training a supervised predictor. This paper presents a universal schema approach to fine-grained entity type prediction. The set of types is taken as the union of textual surface patterns (e.g. appositives) and pre-defined types from available databases (e.g. Freebase)—yielding not tens or hundreds of types, but tens of thousands of entity types, such as financier, criminologist, and musical trio. We robustly learn mutual implicature among this large union by matrix completion using embeddings learned from probabilistic matrix factorization, thus avoiding the need for hand-labeled data. Experimental results demonstrate more than 15% reduction in error versus a traditional classification approach on predicting fine-grained entities types.

1. INTRODUCTION
Classifying entities into different categories is a common task in many NLP systems. In some cases, such as knowledge base construction, entity types may be a prominent user-visible feature [6, 18]. In others, such as relation extraction [36, 27] or query intent discovery [1] entity types are hidden variables included to improve accuracy on the target task. Occasionally the ontology of entity types is coarse, such as the four types in the CoNLL-2003 shared task (person, organization, location and miscellaneous), but often finer-grained ontologies are more useful. For example, specializations of people, including politician, scientist, and athlete are defined in [16, 17, 12]. Others are even more detailed; for instance the Unified Medical Language System (UMLS) defines an ontology of 987,321 biomedical concepts.

Defining such ontologies is a significant challenge, often giving rise to debates about desired granularity and subtle questions about boundary cases. These difficulties appear both when the assignment of entities to types is exclusive and when it is one-to-many.

Once the ontology is defined, the problem of building the automated classification system remains. The most common approach is supervised training from a set of entity mentions labeled into the ontology [16, 34]. However labeling such data is difficult—especially with fine-grained ontologies. Furthermore, when the ontology evolves or expands (as it often does), the data labeling must be revisited. Even when used as hidden variables, the set of entity types may warrant adjustment because an ontology tuned to the task at hand typically performs better—for example the authors of [24] show that the entity types in the WordNet ontology [14] is not as effective as those derived from automatic clustering for the task of learning selectional preferences. Unsupervised clustering may also be employed to derive entity types [35, 13, 24], but the resulting types often have peculiar, undesirable boundary and granularity choices.

This paper presents an approach to fine-grained entity type classification that avoids the need to hand-design an ontology, avoids the need for labeled data, and avoids the boundary difficulties that arise from forcing our semantics into finite, pre-defined, somewhat arbitrary “boxes”. We accomplish this by adopting the universal schema approach, which is previously applied to relation extraction [26, 37], and extending it to entity types. In “universal schema”, our types are the union of all available types from all input sources, including multiple pre-existing ontologies and naturally-occurring textual surface-form expressions that indicate entity type, such as appositives, isa-expressions, or even adjectival or verb phrases. For example, “James Cameron” may appear as a person/director in Freebase, as a PERSON in TAC/KBP, and as a movie-mogul, Canadian citizen, and jerk in various appositives in available text. Rather than five, fifty or five-hundred entity types, this universal schema approach typically yields tens-of-thousands of entity types (particularly from the textual surface forms). It does not force the natural diversity and ambiguity of the original input types into a smaller set of types.

The key characteristic of universal schema is that it models directed implicature among the many candidate types of an entity by casting the problem as a large matrix completion task. Each row in the matrix corresponds to an entity;
each column an entity type; some cells of the matrix are observed and marked true; many are unobserved; it is the job of matrix completion to “fill in” the matrix, marking the unobserved cells as either true or false. For example, although we may not have directly observed that “Barack Obama” is a leader, our model will infer it by having observed that he is a president and commander-in-chief—doing so by leveraging various patterns of co-occurrences among these types in other entities. Similarly it will infer that he is not a mouse-mogul or masterpiece. As in our previous work, we perform this matrix completion task using probabilistic matrix factorization—efficiently estimating vector embeddings for both entities and types by online stochastic gradient descent optimization. The confidence in an entity’s type assignment is determined by the dot-product of the corresponding embeddings, mapped through a logistic function.

Having so many entity types, including types appearing in natural language, allows users to query our system in natural language. That is, rather than having to learn an idiosyncratic ontology, users can ask about entity types in their own vocabulary, and we will most likely already have a column to match. The large number of entity types does make evaluation a challenge. We cannot evaluate every cell in the matrix. Thus we choose to evaluate a subset of the columns (entity types) on a closed set of entities which we have annotated. Here our approach achieves a 15% absolute increase in F1 versus the traditional classification method. Furthermore, although it does not leverage the diversity of universal schema, we also compare against a baseline method for Freebase type prediction [4]; here we achieve similar results as the baseline. In spite of the large number of types, training our system is still efficient, taking approximately 6 hours on one machine for 100 iterations, 100 components on about 500K entities and 10K types.

2. FACTORIZATION MODELS

We present a matrix factorization model to collectively learn semantic implication among unary relations and predict new relations for entities. We fill a matrix $E \times R$ with unary relation instances, where $E$ corresponds to entities and $R$ to unary relations. Assume we index an entity with $e$ and a relation with $r$. Each matrix cell is a binary variable, denoted as $x_{e,r}$. The variable is 1 when relation $r$ holds for entity $e$, and 0 otherwise. For example, observing “directed by Marzieh Meshkini”, we fill the corresponding cell (Marzieh Meshkini, directed by X) with 1.

In our matrix factorization approach we associate each entity and relation with latent vectors $a_e, v_r$, in a $K$-dimensional space, respectively. The dot-product $\theta_{e,r} = \sum_e a_e v_r$ of these vectors for a given entity $e$ and unary relation $r$ then becomes the natural parameters of a Bernoulli distribution that generates the observed binary data. That is, the probability of $x_{e,r} = 1$ is given by $\sigma(\theta_{e,r})$ where $\sigma$ is the sigmoid function. This model corresponds to an instantiation of generalized PCA [9].

To learn the low dimensional latent vectors we maximize the log likelihood of the observed cells under the probabilistic model above. Notice that in our training data we only observe positive cells and have no accurate data on which relations do not hold for an entity. However, learning requires negative training data. We address this issue by sampling negative relations for an entity based on their frequencies in the whole dataset. In our experiments we use stochastic gradient optimization to effectively deal with the large scale of our matrices. In each iteration, we traverse random permutations of all training cells, randomly sample some negative cells for each training cell, and update the corresponding $a_e$ and $v_r$ vectors for the positive and negative cells based on their gradients (omitted here for brevity).

3. EXPERIMENTS

We extract unary relations from New York Times data for the years 1990 to 2007 [28]. We preprocess the documents by performing NER tagging [15] and dependency parsing [23]. Following [33], we extract dependency paths originating from a (named) entity mention as unary relations. Specifically, we traverse from the head token of the entity mention to the root of the dependency tree. Whenever we come across a content word (nouns, adjectives etc.), the current (lexicalized) path from the entity mention to this content word node is used as one unary relation. We stop when approaching a verb or a clause boundary. Additionally, when a verb is encountered, other modifiers of the verb are also included in the path. For example, we can have “X buy share”, “X roll over”. This yields many relations that could serve as entity types, including appositive structures. For example, the unary relation “X, a magnate” can define “magnate” as the corresponding entity type.

In training our matrix factorization model, we set the regularizers $\lambda$s for both entity and unary relation vectors as 0.02. We use 100 components, and run 100 iterations using stochastic gradient optimization. We experiment with different configurations and the current one results in the best performance.

Our task is to predict missing cells in the matrix. In the following, we design experiments to measure the accuracies of these predictions.

3.1 Pattern-based Evaluation

Our universal schema approach typically yields tens-of-thousands of entity types, particularly from the textual surface patterns. In this section, we demonstrate our predictions on these unary relations. Since there is no ground truth for these patterns (other than a subsample of positive-only cells), we cannot easily evaluate them. Instead, we query some of them and list the top ranked entities according to our model and the baseline. The set of queried unary relations in our experiments consists of patterns based on appositives. For example, “X, a scientist”, “X, an actor”, and “X, a band”. Intuitively these relations are most directly corresponding to entity types, scientist, actor, and band in this case. We ask human annotators to annotate each returned entity for a given pattern-based relation. The annotators are provided with sentences in which the entities are mentioned.

Entities from NYT articles are split into training and test set: 355,942 entities vs 147,359 entities. In total the input matrix has about 500K rows, 16K columns. When training the model we hide all query patterns in the test set. We compare our approach against a binary classifier that con-
Table 1: F1 measure on 8 patterns of different approaches. Our approach (Univ) achieves significant better performance on almost of all these patterns.

<table>
<thead>
<tr>
<th>Query</th>
<th>Univ</th>
<th>ME</th>
</tr>
</thead>
<tbody>
<tr>
<td>politician</td>
<td>0.738</td>
<td>0.448</td>
</tr>
<tr>
<td>scientist</td>
<td>0.499</td>
<td>0.354</td>
</tr>
<tr>
<td>magnate</td>
<td>0.433</td>
<td>0.460</td>
</tr>
<tr>
<td>band</td>
<td>0.413</td>
<td>0.427</td>
</tr>
<tr>
<td>reporter</td>
<td>0.437</td>
<td>0.330</td>
</tr>
<tr>
<td>actor</td>
<td>0.649</td>
<td>0.518</td>
</tr>
<tr>
<td>player</td>
<td>0.840</td>
<td>0.711</td>
</tr>
<tr>
<td>magazine</td>
<td>0.845</td>
<td>0.675</td>
</tr>
<tr>
<td>Micro</td>
<td>0.701</td>
<td>0.557</td>
</tr>
<tr>
<td>Macro</td>
<td>0.607</td>
<td>0.490</td>
</tr>
</tbody>
</table>

Table 3: F1 scores of different approaches.

<table>
<thead>
<tr>
<th>System</th>
<th>Univ</th>
<th>ME</th>
<th>UW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Micro</td>
<td>0.515</td>
<td>0.501</td>
<td>0.553</td>
</tr>
<tr>
<td>Macro</td>
<td>0.172</td>
<td>0.120</td>
<td>0.180</td>
</tr>
</tbody>
</table>

Our framework can also incorporates entity types from ontologies as columns. To make comparison against traditional approaches for entity type classification, we evaluate our predictions on pre-defined entity types as well. Specifically, we label entities occurring in a set of held-out with labels from a predefined set of types in Freebase. In this dataset all entities are annotated with all possible relations exhaustively, and this enables us to measure precision, recall and F1. We choose to compare against UW [20], in which the authors employ a multi-class multi-label classifier for fine-grained entity recognition. They test their model on a dataset of 18 documents and about 430 sentences. In this data set, entity mentions are annotated and each mention is labeled with all possible entity types. Our approach is only concerned about the types of entities (sets of entity mentions), not entity mentions. We therefore collapse entity mentions that have the same surface string into single entities. As the UW works on an entity mention basis, we aggregate their output for entity mentions of the same entity. Again we also compare against maximum entropy (ME) classifiers that are trained on labels of entities instead of mentions. Here we have one binary classifier per unary relation, and hence the same entity can have several relations.

Notice that our approach (Univ) and the ME baseline use the same training data as that is used in UW [20]. The input matrix has about 623K entities and 724K columns, 12M positive cells and 36M sampled negative cells. It takes about 10 hours to train the model for 100 iterations using 100 components.

Table 3 shows the F1 scores of different systems. The macro average numbers are small due to that many types get 0.0 in F1. Our approach is better than ME, slightly worse than UW. Looking closely at different entity types, we find that our approach is better at predicting “sports_league”, “athlete”, “person/coach”, “education/department”; It is worse at predicting “location/city”, “location/province”, “location/country”, and “news_agency”. Our model has higher recall but sometimes suffers from lower precision for fine-grained types that have fewer training instances, such as “news_agency”.

4. RELATED WORK

**Universal Schema.** The authors previously introduced universal schema for relation extraction [26, 37]. There the rows correspond to entity-pairs and columns correspond to relation types. A special three-part parameterization for matrix factorization is employed to complete the matrix. In this paper we extend the universal schema approach to entity types. This represents a first step towards future work in joint entity-type and relation-type prediction, both with universal schema and matrix factorization.

**Low Dimensional Embedding.** Learning low dimensional embeddings of high-dimensional NLP data has been
of both long-standing [5, 2, 10, 3] and rising interest [32, 21]. Much of this work has been for the embedding of individual words [5, 2, 3, 21]. Some has been for structured natural language processing, such as part-of-speech tags, chunks, named entity tags and semantic roles [10], or for parsing [32]. Some recent work uses tensor factorization for embedding relations from triples of semantic role labeling (subj-verb-obj) [19], from a structured knowledge base [22] or from WordNet relations [8]. Our work is the first of which to predict “open domain” universal schema for relations or entity types by leveraging natural language inputs.

**Entailment.** Our work is also related to semantic inference over text [11], in which given a hypothesis and textual evidence, the system must predict whether or not the text entails the hypothesis. In our framework, observed cells of the matrix are the evidence, and newly predicted cells are the hypotheses. Szpektor and Dagan [33] aim to discover implications among unary patterns for predicting new unary facts. We have a similar goal here. They concentrate on verb-triggered patterns, whereas we focus on patterns that define entity types, including noun-triggered patterns (such as appositives) and verb-triggered patterns. They employ distributional similarity where we use matrix factorization. Other work addresses semantic inference over relation instances, for example, by learning rules that conjoin textual patterns extracted by OpenIE [29, 30].

**Fine-Grained Entity Type Classification.** Classifying entities into large ontologies is a commonly tackled task and is widely acknowledged as useful, not only for knowledge base construction, but also query log prediction [25]. Some researchers have explored entity type classification specifically for categories of people [16, 17, 12]. Others have defined ontologies with a wider variety of entity types, but not implemented methods to automatically classify instances into the ontology [31]. Large-scale knowledge bases, such as Freebase and its fine-grained entity types, have significant collections of entities that can be used for training traditional classification methods by distant supervision [20]. Others have also performed entity type classification with a multi-label classifier in a hierarchy of types [38]. The main differences to our approach are (1) that we use matrix factorization rather than classification as the framework for our model, and more importantly (2) we do not restrict ourselves to predefined entity types, instead leveraging the wide diversity of naturally available data. Even when a pre-existing knowledge base can provide supervision for a classifier, the resulting entity type classifier is still limited by the types envisioned by the creators of the knowledge base ontology. Furthermore, note that even when the goal is merely classification into a specific ontology, matrix factorization’s striving to predict many other text-based entity types provides a kind of multi-task learning [7] that can be beneficial.

5. CONCLUSION

This paper has presented universal schema for assigning entities into multiple of over 16,000 entity types on NYT data. We use the term “universal” because the set of types is formed by the union of textual surface patterns and multiple input entity type ontologies. We find that our approach, based on matrix factorization, reduces error by more than 30% in comparison with a traditional classifier on a set of entities from the widely-diverse textual appositive-derived entity types.

There are significant opportunities for future work. We have begun to integrate this paper’s entity type model with our previous relation type model, and we expect further increases in accuracy to arise from learning these two jointly. We also plan to explore new strategies for obtaining negative training signal and for integrating more observed features of the entity mentions.

6. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by DARPA under agreement number FA8750-13-2-0020. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

7. REFERENCES


<table>
<thead>
<tr>
<th>Query</th>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>politician</td>
<td>legislator, official, politician like, vote, campaign of, criticism from, ally, accuse of, defeat by, election, lash out, lawmaker, whip, endorse, oppose by, run against, re-elected, opponent, conservative</td>
</tr>
<tr>
<td>scientist</td>
<td>criminologist, biologist, psychologist, sociologist, professor, researcher, neuroscientist, co-author with, ecologist, expert, physicist</td>
</tr>
<tr>
<td>band</td>
<td>tune, album, song by, act like, ’s singer, hit for, ’s song, country, concert, singer of, wing with, music of, tour with, musician like, trio, duo, blues, sound of, rock, folk, pop, recording by,</td>
</tr>
</tbody>
</table>


