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 types is useful in many applications, including knowledge base construction, relation extraction and query intent prediction. Fine-grained entity type ontologies are especially valuable, but typically difficult to design because of unavoidable 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 more than ten thousands of entity types, such as financier, criminologist, and musical trio. We robustly learn mutual implication among this large union by learning latent vector embeddings from probabilistic matrix factorization, thus avoiding the need for hand-labeled data. Experimental results demonstrate more than 30% reduction in error versus a traditional classification approach on predicting fine-grained entities types.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text analysis

Keywords

Entity Prediction, Matrix Factorization

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, 17]. In others, such as relation extraction [32, 24] 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 works [15, 16, 11]. 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 [15, 30]. However labeling such data is painful—especially with fine-grained ontologies. Furthermore, when the ontology evolves or expands (as it often does), the data labeling must be re-visited. 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 Pantel et al [21] show that the entity types in the WordNet ontology [13] 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 [31, 12, 21], 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 [23, 33], 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 many thousands of entity types (particularly from the textual surface forms). It does not force the natural diversity and ambiguity of the original in-

AKBC'13, October 27–28, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2509558.2509572 ...$15.00.

put types into a smaller set of types. These types may also be understood as unary relations.

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, obtained by coreference on the entity mentions; 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 observing 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 *movie-mogul* or a *waterfall*. 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. Table 1 shows our example input and output: the first column lists several entities, the second column shows observed types for an entity and the third column shows newly predicted types. Note that we use simple string match to do coreference.

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 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, 7% against a multi-instance multi-label entity recognition system. 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 comparable results as the baseline, and our approach is better when running experiments in the same setting. In spite of the large number of types, training our system is still efficient, taking approximately 2.5 hours on one machine for 50 iterations, 100 components on about 623K entities and 51K 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 "Mohamed al-Fayed, a tycoon", we fill the corresponding cell (Mohamed al-Fayed, tycoon) with 1.

In our matrix factorization approach we associate each entity and relation with latent vectors $\mathbf{a}_e$ and $\mathbf{v}_r$ in a K-dimensional space, respectively. The dot-product $\theta_{e,r} = \sum_c^K a_{e,c} v_{r,c}$ 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 σ is the sigmoid function. This model corresponds to an instantiation of generalized PCA [8].

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 unobserved relations for an entity based on their frequencies in the whole dataset and treating them as negative. 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 three negative cells for each training cell, and update the corresponding $\mathbf{a}_e$ and $\mathbf{v}_r$ vectors for the positive and negative cells based on their gradients (Equations omitted here for brevity).

2.1 Neighbor Model

In our matrix factorization model, we can take surface patterns and entity types from existing knowledge bases as columns. There are other information sources that can help entity prediction, such as tokens in the entity itself, the neighbor tokens of the entity. To incorporate these information sources into entity prediction, we introduce neighbor model. That is, we take these information sources as features, columns in the matrix as target labels and learn a multi-class classifier. Similarly to our matrix factorization model, we define the natural parameter for a cell as $\theta_{e,r} = \sum_i (w_i f_i(e,r))$, where $f_i(e,r)$ is a conjunctive feature, for example "token=Association,label=organization." We employ a combined model in our experiments for predicting Freebase relations (see section 3.2), defining $\theta_{e,r}$ as the sum of matrix factorization and neighbor model score. We still employ maximum log likelihood and learn the parameters using stochastic gradient descent.

3. EXPERIMENTS

We extract unary relations from New York Times data for the years 1990 to 2007 [25]. We preprocess the documents by performing NER tagging [14] and dependency parsing [20]. Following [29], 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.

3.1 Pattern-based Evaluation

Our universal schema approach typically yields many thousands of entity types, particularly from the textual surface patterns. In this section, we demonstrate our predictions

Entity	Observed	Predicted
Mohamed al-Fayed	tycoon, owner, entrepreneur, financier estate of X, purchase by X	billionaire, millionaire, magnate, person:Freebase
House of Pain	X's member, reminiscent of X, music by X, rap group X	trio, band, X rap singer of X, rapper X
Sprite	commercial, drink, brand, ad drink X, campaign for X	product:Freebase food:Freebase
Pierre Louys	poem of, novel by, 's poem, 's novel	poet, novelist, author
Rick Wamsley	goaltender, goalie	goalkeeper

Table 1: Some entities, observed unary relations and predicted ones by our system. We can describe an entity in any granularity based on the patterns or types from ontologies. The patterns are translated from dependency parsing paths as described in section 3.

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 directly correspond 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 approximately 500K rows, 17K columns. We apply our matrix factorization model on this data, and set the regularizers λs for both entity and unary relation vectors as 0.02. We use 100 components, and run 100 iterations using stochastic gradient optimization. When training the model we hide all query patterns in the test set. We compare our approach against a binary classifier (Marked "Classifier" in table) that considers entities co-occurring with the query pattern as positive examples and all others as negative examples. As the classification model we use maximum entropy. We also compare against a multi-instance multi-label distant supervision classifier for fine-grained entity recognition (UW) [18].

Entities in the test set are selected for annotation by the following rules. Entities are ranked with respect to each pattern by our system and the baseline systems separately. Top 100 entities of each target pattern ranked by each system are shown to the annotators. We also acquire annotations from Freebase. For example, we consider all entities that have labels *politician*, *us_congressperson*, *us_senator*, *us_vicepresident* as instances of our target pattern "X, a politician." Similarly we obtain entities for target patterns "X, a player," and "X, an actor." Mappings from Freebase labels to these three patterns are from UW. In total, we have annotations for 14,991 entities in the test set.

We measure precision, recall and F1 for entities in this set. For each entity, patterns with probabilities above a threshold are considered as true. The threshold is set to 0.5. In scenarios where an entity has no patterns above the threshold, the top ranked one is selected. This may lower the precision of each system, however, it does increase the recall and F1 score for both our approach and the baseline.

Query	Classifier	UW	Universal
politician	0.448	0.549	**0.738**
scientist	0.354	0.404	**0.499**
magnate	0.460	**0.529**	0.433
band	**0.427**	0.404	0.413
reporter	0.330	0.356	**0.437**
actor	0.518	0.598	**0.649**
player	0.711	0.794	**0.840**
magazine	0.675	0.727	**0.845**
Overall	0.557	0.635	**0.701**

Table 2: F1 measure on 8 patterns. Our approach (Universal) achieves significant better performance than the classifier and UW approach on almost of all these patterns.

Table 2 lists the F1 measures for each pattern. We can see that our approach performs significantly better than the baseline on 6 out of 8 patterns. We perform slightly worse on two patterns. On micro average, we gain approximately 15% in F1 score against the classifier, 7% against UW. When analyzing the errors made by the baseline system we see most problems when there are no patterns above the threshold. In these cases the baseline's top ranked patterns (now with score under the threshold) are mostly incorrect. However, for our model the top ranked patterns, when under the threshold, are still often correct.

3.2 Closed Set Evaluation

Our framework can also incorporate 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 use labeled entities occurring in a set of held-out documents with a predefined set of types in Freebase. Exhaustive annotation for all entities with all possible types enables us to measure precision, recall and F1. We compare against UW [18], in which the authors employ a multi-class multi-label classifier for fine-grained entity recognition. They test their model on a much smaller 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 use simple string match to cluster mentions into entities. As the UW approach works on predicting entity mentions, we

Classifier	UW	UW-mention	Universal
0.501	0.523	0.553	0.553

Table 3: F1 scores of different approaches.

obtained two results, one run with applying their code to the entities, i.e. cluster of mentions (UW), the other run by translating their predictions on mentions into predictions on entities with string match coreference (UW-mention). Again we compare against a classifier based approach that is used in distant supervision. We train a series of classifiers on labels of entities, with one binary classifier per entity type, and one entity can have several relations.

In entity prediction, besides dependency path, features, such as tokens, head tokens, the contextual unigrams and bigrams of the entity mentions are useful as well. These features are similar to those used in UW [18]. We use neighbor model to incorporate these features; see 2.1. We employ sequential training, first train a neighbor model with a few iterations, fix the feature weights, and train our matrix factorization model. We tried joint training, updating parameters for these two models. On this data, sequential training leads to better results. Table 3 shows the F1 scores of different systems. Our approach is better than the classifier based approach, achieves better performance than running UW code on entities, comparable with taking the UW mention predictions and collapsing them into entity labels.

Notice that our approach (Universal) and the classifier baseline use the same training data as that is used in UW [18]. The input matrix has approximately 623K entities, 51K columns and 1.8M observed cells. We sample three times as many negative cells as positive cells. It takes approximately 2.5 hours to train the model for 50 iterations using 100 components.

4. RELATED WORK

Universal Schema. The authors previously introduced *universal schema* for relation extraction [23, 33]. 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, 9, 3] and rising interest [28, 19]. Much of this work has been for the embedding of individual words [5, 2, 3, 19]. Some has been for structured natural language processing, such as part-of-speech tags, chunks, named entity tags and semantic roles [9], or for parsing [28]. Our work is the first to predict "open domain" universal schema for entity types by leveraging natural language inputs.

Entailment. Our work is also related to semantic inference over text [10], 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 [29] 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 [26, 27].

Fine-Grained Entity Type Classification. Classifying entities into large ontologies is a common task and widely acknowledged as useful, not only for knowledge base construction, but also query log prediction [22]. Some researchers have explored entity type classification specifically for people [15, 16, 11]. 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 [18]. 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 approximately 17K 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 derived entity types. We achieve comparable performance with a multi-instance multi-class entity prediction system on predefined types from ontologies.

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

[1] K. Balog and R. Neumayer. Hierarchical target type identification for entity-oriented queries. In *Proceedings of CIKM*, 2012.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[3] J. Blitzer, K. Q. Weinberger, L. K. Saul, and F. C. N. Pereira. Hierarchical distributed representations for statistical language modeling. In *Proceedings of NIPS*, 2004.

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA, 2008. ACM.

[5] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[6] A. Carlson, J. Betteridge, R. Wang, E. Hruschka, and T. Mitchell. Coupled semi-supervised learning for information extraction. In *Third ACM International Conference on Web Search and Data Mining (WSDM '10)*, 2010.

[7] R. A. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of ICML*, 1993.

[8] M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal component analysis to the exponential family. In *Proceedings of NIPS*, 2001.

[9] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, 2008.

[10] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.

[11] A. Ekbal, E. Sourjikova, A. Frank, and S. P. Ponzetto. Assessing the challenge of fine-grained named entity recognition and classification. In *Named Entities Workshop, ACL*, 2010.

[12] M. Elsner, E. Charniak, and M. Johnson. Structured generative models for unsupervised named-entity clustering. In *Proceedings of NAACL*, 2009.

[13] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[14] J. R. Finkel, T. Grenager, and C. Manning. 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 '05)*, pages 363–370, June 2005.

[15] M. Fleischman and E. Hovy. Fine grained classification of named entities. In *Proceedings of Coling*, 2002.

[16] C. Giuliano and A. Gliozzo. Instance-based ontology population exploiting named-entity substitution. In *Proceedings of Coling*, 2008.

[17] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 2012.

[18] X. Ling and D. S. Weld. Fine-grained entity recognition. In *Proceedings of AAAI*, 2012.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representation in vector space. In *Proceedings of workshop at ICLR*, 2013.

[20] J. Nivre, J. Hall, and J. Nilsson. Memory-based dependency parsing. In *Proceedings of CoNLL*, pages 49–56, 2004.

[21] P. Pantel, R. Bhagat, B. Coppola, T. Chklovski, and E. Hovy. ISP: Learning Inferential Selectional Preferences. In *Proceedings of NAACL HLT*, 2007.

[22] P. Pantel, T. Lin, and M. Gamon. Mining entity types from query logs via user intent modeling. In *Proceedings of ACL*, 2012.

[23] S. Riedel, L. Yao, A. McCallum, and B. Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL*, 2013.

[24] D. Roth and W. Yih. Global inference for entity and relation identification via a linear programming formulation. 2007.

[25] E. Sandhaus. *The New York Times Annotated Corpus*. Linguistic Data Consortium, Philadelphia, 2008.

[26] S. Schoenmackers, O. Etzioni, and D. S. Weld. Scaling textual inference to the web. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–88, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[27] S. Schoenmackers, O. Etzioni, D. S. Weld, and J. Davis. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1088–1098, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[28] R. Socher, C. D. Manning, and A. Y. Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *NIPS*, 2010.

[29] I. Szpektor and I. Dagan. Learning entailment rules for unary templates. In *Proceedings of Coling*, 2008.

[30] H. Tanev and B. Magnini. Weakly supervised approaches for ontology population. In *Proceedings of 11st Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[31] L. Yao, A. Haghighi, S. Riedel, and A. McCallum. Structured relation discovery using generative models. In *Proceedings of EMNLP*, 2011.

[32] L. Yao, S. Riedel, and A. McCallum. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023, Cambridge, MA, October 2010. Association for Computational Linguistics.

[33] L. Yao, S. Riedel, and A. McCallum. Probabilistic databases of universal schema. In *Proceedings of the AKBC-WEKEX Workshop at NAACL*, 2012.