How Well can We Learn Interpretable Entity Types from Text?

Dirk Hovy
Center for Language Technology
University of Copenhagen
Njalsgade 140, 2300 Copenhagen
dirk@cst.dk

Abstract

Many NLP applications rely on type systems to represent higher-level classes. Domain-specific ones are more informative, but have to be manually tailored to each task and domain, making them inflexible and expensive. We investigate a largely unsupervised approach to learning interpretable, domain-specific entity types from unlabeled text. It assumes that any common noun in a domain can function as potential entity type, and uses those nouns as hidden variables in a HMM. To constrain training, it extracts co-occurrence dictionaries of entities and common nouns from the data. We evaluate the learned types by measuring their prediction accuracy for verb arguments in several domains. The results suggest that it is possible to learn domain-specific entity types from unlabeled data. We show significant improvements over an informed baseline, reducing the error rate by 56%.

1 Introduction

Many NLP applications, such as question answering (QA) or information extraction (IE), use type systems to represent relevant semantic classes. Types allow us to find similarities at a higher level to group lexically different entities together. This helps to filter out candidates that violate certain constraints (e.g., in QA, if the intended answer type is PERSON, we can ignore all candidate answers with a different type), but is also used for feature generation and fact-checking.

A central question is: *where do the types come from?* Typically, they come from a hand-constructed set. This has some disadvantages. Domain-general types, such as named entities or WordNet supersenses (Fellbaum, 1998), often fail to capture critical domain-specific information (in the medical domain, we might want ANTIBIOTIC, SEDATIVE, etc., rather than just ARTIFACT). Domain-specific types perform much better (Ferrucci et al., 2010), but must be manually adapted to each new domain, which is expensive. Alternatively, unsupervised approaches (Ritter et al., 2010) can be used to learn clusters of similar words, but the resulting types (=cluster numbers) are not human-interpretable, which makes analysis difficult. Furthermore, it requires us to define the number of clusters beforehand.

Ideally, we would like to learn domain-specific types directly from data. To this end, pattern-based approaches have long been used to induce type systems (Hearst, 1992; Kozareva et al., 2008). Recently, Hovy et al. (2011) proposed an approach that uses co-occurrence patterns to find entity type candidates, and then learns their applicability to relation arguments by using them as latent variables in a first-order HMM. However, they only evaluate their method using human sensibility judgements for one domain. While this shows that the types are coherent, it does not tell us much about their applicability.

We extend their approach with three important changes:

1. we evaluate the types by measuring accuracy when using them in an extrinsic task,
2. we evaluate on more than one domain, and
3. we explore a variety of different models.

We measure prediction accuracy when using the learned types in a selectional restriction task for frequent verbs. E.g., given the relation *throw(X, pass)* in the football domain, we compare the model prediction to the gold data *X=QUARTERBACK*. The results indicate that the learned types can be used to in relation extraction tasks.

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 482–487,
Baltimore, Maryland, USA, June 23-25 2014.

Our contributions in this paper are:

- we empirically evaluate an approach to learning types from unlabeled data
- we investigate several domains and models
- the learned entity types can be used to predict selectional restrictions with high accuracy

2 Related Work

In relation extraction, we have to identify the relation elements, and then map the arguments to types. We follow an open IE approach (Banko and Etzioni, 2008) and use dependencies to identify the elements. In contrast to most previous work (Pardo et al., 2006; Yao et al., 2011; Yao et al., 2012), we have **no** pre-defined set of types, but try to learn it along with the relations. Some approaches use types from general data bases such as Wikipedia, Freebase, etc. (Yan et al., 2009; Eichler et al., 2008; Syed and Viegas, 2010), sidestepping the question how to construct those DBs in the first place. We are less concerned with extraction performance, but focus on the accuracy of the learned type system by measuring how well it performs in a prediction task.

Talukdar et al. (2008) and Talukdar and Pereira (2010) present graph-based approaches to the similar problem of class-instance learning. While this provides a way to discover types, it requires a large graph that does not easily generalize to new instances (transductive), since it produces no predictive model. The models we use are transductive and can be applied to unseen data. Our approach follows Hovy et al. (2011). However, they only evaluate one model on football by collecting sensibility ratings from Mechanical Turk. Our method provides extrinsic measures of performance on several domains.

3 Model

Our goal is to find semantic type candidates in the data, and apply them in relation extraction to see which ones are best suited. We restrict ourselves to verbal relations. We build on the approach by Hovy et al. (2011), which we describe briefly below. It consists of two parts: extracting the type candidates and fitting the model.

The basic idea is that semantic types are usually common nouns, often frequent ones from the domain at hand. Thus all common nouns are possible types, and can be used as latent variables in an HMM. By estimating emission and transition parameters with EM, we can learn the subset of nouns to apply.

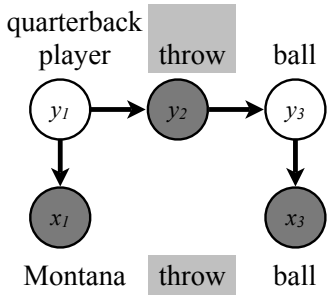

Figure 1: Example of input sentence $\mathbf{x}$ and output types for the HMM. Note that the verb type is treated as observed variable.

However, assuming the set of all common nouns as types is intractable, and would not allow for efficient learning. To restrict the search space and improve learning, we first have to learn which types modify entities and record their co-occurrence, and use this as dictionary.

Kleiman: professor:25, expert:13, *(specialist:1)*
Tilton: executive:37, economist:17, *(chairman:4, president:2)*

Figure 2: Examples of dictionary entries with counts. Types in brackets are not considered.

Dictionary Construction The number of common nouns in a domain is generally too high to consider all of them for every entity. A common way to restrict the number of types is to provide a dictionary that lists all legal types for each entity (Merialdo, 1994; Ravi and Knight, 2009; Täckström et al., 2013). To construct this dictionary, we collect for each entity (i.e., a sequence of words labeled with NNP or NNPS tags) in our data all common nouns (NN, NNS) that modify it. These are

1. nominal modifiers ("*judge* Scalosi ..."),
2. appositions ("Tilton, a *professor* at ..."), and
3. copula constructions ("Finton, who is the *investor* ...").

These modifications can be collected from the dependency parse trees. For each entity, we store the

type candidates and their associated counts. See Figure 2 for examples. We only consider types observed more than 10 times. Any entity without type information, as well as dictionary entities with only singleton types are treated as unknown tokens ("UNK"). We map UNK to the 50 most common types in the dictionary. Verbs are considered to each have their own type, i.e., token and label for verbs are the same.
We do not modify this step.

Original Model Hovy et al. (2011) construct a HMM using subject-verb-object (SVO) parse triples as observations, and the type candidates as hidden variables. Similar models have been used in (Abney and Light, 1999; Pardo et al., 2006). We estimate the free model parameters with EM (Dempster et al., 1977), run for a fixed number of iterations (30) or until convergence.

Note that Forward-backward EM has time complexity of $\mathcal{O}(N^2T)$, where N is the number of states, and T the number of time steps. $T = 3$ in the model formulations used here, but N is much larger than typically found in NLP tasks (see also Table 3). The only way to make this tractable is to restrict the free parameters the model needs to estimate to the transitions.

The model is initialized by jointly normalizing [1] the dictionary counts to obtain the emission parameters, which are then fixed (except for the unknown entities ($P(word = \texttt{UNK}|type = \cdot)$). Transition parameters are initialized uniformly (restricted to potentially observable type sequences), and kept as free parameters for the model to optimize.

Common nouns can be both hidden variables and observations in the model, so they act like annotated items: their legal types are restricted to the identity. All entities are thus constrained by the dictionary, as in (Merialdo, 1994). To further constrain the model, only the top three types of each entity are considered. Since the type distribution typically follows a Zipf curve, this still captures most of the information.

[1] This preserves the observed entity-specific distributions. Under conditional normalization, the type candidates from frequent entities tend to dominate those of infrequent entities. I.e., the model favors an unlikely candidate for entity a if it is frequent for entity b.

The model can be fully specified as

$$P(\mathbf{x}, \mathbf{y}) = P(y_1) \cdot P(x_1|y_1) \prod_{i=2}^{3} P(y_i|y_{i-1}) \cdot P(x_i|y_i) \quad (1)$$

where $\mathbf{x}$ is an input triple of a verb and its arguments, and $\mathbf{y}$ a sequence of types.

4 Extending the Model

The model used by Hovy et al. (2011) was a simple first order HMM, with the elements in SVO order (see Figure 3a). We observe two points: we always deal with the same number of elements, and we have observed variables. We can thus move from a sequential model to a general graphical model by adding transitions and re-arranging the structure.

Since we do not model verbs (they each have their identity as type), they act like observed variables. We can thus move them in first position and condition the subject on it (3b).

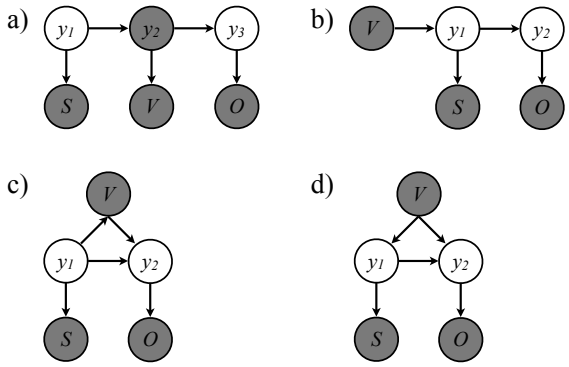

Figure 3: Original SVO. model (a), modified VSO order (b), extension to general models (c and d)

By adding additional transitions, we can constrain the latent variables further. This is similar to moving from a first to a second order HMM. In contrast to the original model, we also distinguish between unknown entities in the first and second argument position.

The goal of these modifications is to restrict the number of potential values for the argument positions. This allows us to use the models to type individual instances. In contrast, the objective in Hovy et al. (2011) was to collect frequent relation templates from a domain to populate a knowledge base.

The modifications presented here extend to

	Football				Finances				Law			
system	**arg1**	**arg2**	**avg**	ΔBL	**arg1**	**arg2**	**avg**	ΔBL	**arg1**	**arg2**	**avg**	ΔBL
baseline	0.28	0.26	0.27	—	0.39	0.42	0.41	—	0.37	0.32	0.35	—
orig.	0.05	0.23	0.14	–0.13	0.08	0.39	0.23	–0.18	0.06	0.31	0.18	–0.17
VSO, seq.	*0.37*	0.28	0.32	+0.05	0.38	0.45	0.41	0.0	*0.45*	0.37	*0.41*	+0.06
SVO, net	*0.63*	***0.60***	***0.62***	+0.35	*0.55*	***0.63***	***0.59***	+0.18	*0.69*	***0.68***	***0.68***	+0.33
VSO, net	***0.66***	*0.58*	***0.62***	+0.35	***0.61***	*0.54*	*0.57*	+0.16	***0.71***	*0.62*	*0.66*	+0.31

Table 1: Accuracy for most frequent sense baseline and different models on three domains. Italic numbers denote significant improvement over baseline (two-tailed t-test at $p < 0.01$). ΔBL = difference to baseline.

	Football			Finances			Law		
system	**arg1**	**arg2**	**avg**	**arg1**	**arg2**	**avg**	**arg1**	**arg2**	**avg**
orig.	0.17	0.38	0.27	0.18	0.52	0.35	0.17	0.48	0.32
VSO, seq.	0.56	0.42	0.49	0.55	0.58	0.57	0.61	0.51	0.56
SVO, net	0.75	**0.69**	**0.72**	0.68	**0.73**	**0.71**	0.78	**0.77**	**0.78**
VSO, net	**0.78**	0.67	**0.72**	**0.74**	0.66	0.70	**0.81**	0.72	0.76

Table 2: Mean reciprocal rank for models on three domains.

verbs with more than two arguments, but in the present paper, we focus on binary relations.

5 Experiments

Since the labels are induced dynamically from the data, traditional precision/recall measures, which require a known ground truth, are difficult to obtain. Hovy et al. (2011) measured sensibility by obtaining human ratings and measuring weighted accuracies over all relations. While this gives an intuition of the general methodology, it is harder to put in context. Here, we want to evaluate the model's performance in a downstream task. We measure its ability to predict the correct types for verbal arguments. We evaluate on three different domains.

As test case, we use a cloze test, or fill-in-the-blank. We select instances that contain a type-candidate word in subject or object position and replace that word with the unknown token. We can then compare the model's prediction to the original word to measure accuracy.

5.1 Data

Like Yao et al. (2012) and Hovy et al. (2011), we derive our data from the New York Times (NYT) corpus (Sandhaus, 2008). It contains several years worth of articles, manually annotated with meta-data such as author, content, etc. Similar to Yao et al. (2012), we use articles whose *content* meta-data field contains certain labels to distinguish data from different domains. We use the labels *Football*[2], *Law and Legislation*, and *Finances*.

We remove meta-data and lists, tokenize, parse, and lemmatize all articles. We then automatically extract subject-verb-object (SVO) triples from the parses, provided the verb is a full verb. Similarly to (Pardo et al., 2006), we focus on the top 100 full verbs for efficiency reasons, though nothing in our approach prevents us from extending it to all verbs. For each domain, we select all instances which have a potential type (common noun) in at least one argument position. These serve as corpus.

	Football	**Finances**	**Law**
unique types	7,139	18,186	10,618
unique entities	38,282	27,528	12,782

Table 3: Statistics for the three domains.

As test data, we randomly select a subset of 1000 instances for each argument, provided they contain one of the 50 most frequent types in subject or object position, such as *player* in "*player* throw pass". This serves as gold data. We then replace those types by UNK (i.e., we get "UNK throw pass") and use this as test set for our model.[3]

Table 3 shows that the domains vary with re-

[2]The data likely differs from Hovy et al. (2011).

[3]We omit cases with two unknown arguments, since this

spect to the ratio of unique types to unique entities. Football uses many different entities (e.g., team and player names), but has few types (e.g., player positions), while the other domains use more types, but fewer entities (e.g., company names, law firms, etc.).

5.2 Evaluation

We run Viterbi decoding on each test set with our trained model to predict the most likely type for the unknown entities. We then compare these predictions to the type in the respective gold data and compute the accuracy for each argument position. As baseline, we predict the argument types most frequently observed for the particular verb in training, e.g., predict *PLAYER* as subject of *tackle* in football. We evaluate the influence of the different model structures on performance.

6 Results

Table 1 shows the accuracy of the different models in the prediction task for the three different domains. The low results of the informed baseline indicate the task complexity.

We note that the original model, a bigram HMM with SVO order (Figure 3a), fails to improve accuracy over the baseline (although its overall results were judged sensible). Changing the input order to VSO (Figure 3b) improves accuracy for both arguments over SVO order and the baseline, albeit not significantly. The first argument gains more, since conditioning the subject type on the (unambiguous) verb is more constrained than starting out with the subject. Conditioning the object directly upon the subject creates sparser bigrams, which capture "who does what to whom".

Moving from the HMMs to a general graphical model structure (Figures 3c and d) creates a sparser distribution and significantly improves accuracy across the board. Again, the position of the verb makes a difference: in SVO order, accuracy for the second argument is better, while in VSO order, accuracy for the subject increases. This indicates that direct conditioning on the verb is the strongest predictor. Intuitively, knowing the verb restricts the possible arguments much more than knowing the arguments restrict the possible verbs (the types of entities who can throw something are limited, but knowing that the subject is a quarterback still allows all kinds of actions).

We also compute the mean reciprocal rank (MRR) for each condition (see Table 2). MRR denotes the inverse rank in the model's k-best output at which the correct answer occurs, i.e., $\frac{1}{k}$. The result gives us an intuition of "how far off" the model predictions are. Across domains, the correct answer is found on average among the top two (rank 1.36). Note that since MRR require k-best outputs, we cannot compute a measure for the baseline.

7 Conclusion

We evaluated an approach to learning domain-specific interpretable entity types from unlabeled data. Type candidates are collected from patterns and modeled as hidden variables in graphical models. Rather than using human sensibility judgements, we evaluate prediction accuracy for selectional restrictions when using the learned types in three domains. The best model improves 35 percentage points over an informed baseline. On average, we reduce the error rate by 56%. We conclude that it is possible to learn interpretable type systems directly from data.

Acknowledgements

The author would like to thank Victoria Fossum, Eduard Hovy, Kevin Knight, and the anonymous reviewers for their invaluable feedback.

References

Steven Abney and Marc Light. 1999. Hiding a semantic hierarchy in a Markov model. In *Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing*, volume 67.

Michele Banko and Oren. Etzioni. 2008. The tradeoffs between open and traditional relation extraction. *Proceedings of ACL-08: HLT*, pages 28–36.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Kathrin Eichler, Holmer Hemsen, and Günter Neumann. 2008. Unsupervised relation extraction from web documents. *LREC. http://www. lrecconf. org/proceedings/lrec2008*.

Christiane Fellbaum. 1998. *WordNet: an electronic lexical database*. MIT Press USA.

becomes almost impossible to predict without further context, even for humans (compare "UNK make UNK").

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3):59–79.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

Dirk Hovy, Chunliang Zhang, Eduard Hovy, and Anselmo Peñas. 2011. Unsupervised discovery of domain-specific knowledge from text. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1466–1475, Portland, Oregon, USA, June. Association for Computational Linguistics.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. *Proceedings of ACL-08: HLT*, pages 1048–1056.

Bernard Merialdo. 1994. Tagging English text with a probabilistic model. *Computational linguistics*, 20(2):155–171.

Thiago Pardo, Daniel Marcu, and Maria Nunes. 2006. Unsupervised Learning of Verb Argument Structures. *Computational Linguistics and Intelligent Text Processing*, pages 59–70.

Sujith Ravi and Kevin Knight. 2009. Minimized Models for Unsupervised Part-of-Speech Tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 504–512. Association for Computational Linguistics.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Uppsala, Sweden, July. Association for Computational Linguistics.

Evan Sandhaus, editor. 2008. *The New York Times Annotated Corpus*. Number LDC2008T19. Linguistic Data Consortium, Philadelphia.

Zareen Syed and Evelyne Viegas. 2010. A hybrid approach to unsupervised relation discovery based on linguistic analysis and semantic typing. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 105–113. Association for Computational Linguistics.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the ACL*.

Partha P. Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1473–1481. Association for Computational Linguistics.

Partha P. Talukdar, Joseph Reisinger, Marcus Paşca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 582–590. Association for Computational Linguistics.

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. 2009. Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1021–1029. Association for Computational Linguistics.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 712–720. Association for Computational Linguistics.