Using Entity Information from a Knowledge Base to Improve Relation Extraction

Lan Du[1]*, Anish Kumar[2], Mark Johnson[2] and Massimiliano Ciaramita[3]
[1]Faculty of Information Technology, Monash University Clayton, VIC 3800, Australia
[2]Department of Computing, Macquarie University Sydney, NSW 2109, Australia
[3]Google Research, Zurich, Switzerland

Abstract

Relation extraction is the task of extracting predicate-argument relationships between entities from natural language text. This paper investigates whether background information about entities available in knowledge bases such as FreeBase can be used to improve the accuracy of a state-of-the-art relation extraction system. We describe a simple and effective way of incorporating FreeBase's *notable types* into a state-of-the-art relation extraction system (Riedel et al., 2013). Experimental results show that our notable type-based system achieves an average 7.5% weighted MAP score improvement. To understand where the notable type information contributes the most, we perform a series of ablation experiments. Results show that the notable type information improves relation extraction more than NER labels alone across a wide range of entity types and relations.

1 Introduction

The goal of relation extraction is to extract relational information about entities from a large text collection. For example, given the text "Michael Bay, the director of Transformers, visited Paris yesterday," a relation extraction system might extract the relationship *film_director(Michael Bay, Transformers)*. These tuples can be then used to extend a knowledge base. With the increase in the amount of textual data available on the web, relation extraction has gained wide applications in information extraction from both general newswire texts and specialised document collections such as biomedical texts (Liu et al., 2007).

A typical relation extraction system functions as a pipeline, first performing named entity recognition (NER) and entity disambiguation to link the entity mentions found in sentences to their database entries (e.g., "Michael Bay" and "Transformers" would both be linked to their respective database ids). Then the context in which these entity mentions co-occur is used to predict the relationship between the entities. For example, the path in a syntactic parse between two mentions in a sentence can be used as a feature to predict the relation holding between the two entities. Continuing our example, the text pattern feature *X-the-director-of-Y* (or a corresponding parse subtree fragment) might be used to predict the database relation *film_director(X,Y)*. In such a pipeline architecture, information about the entities from the database is available and can be used to help determine the most appropriate relationship between the entities. The goal of this paper is to identify whether that information is useful in a relation extraction task, and study such information about the entities with a set of ablation experiments.

We hypothesise that information from database entries can play the role of background knowledge in human sentence comprehension. There is strong evidence that humans use world knowledge and contextual information in both syntactic and semantic interpretation (Spivey-Knowlton and Sedivy, 1995), so it is reasonable to expect a machine might benefit from it as well. Continuing with our example, if our database contained the information that one particular entity with the name Michael Bay had died a decade before the movie Transformers was released, then it might be reasonable to conclude that this particular individual was unlikely to have directed Transformers. Clearly, modelling all the ways in which such background information about entities might be used would be extremely complex. This paper explores a simple way of using some of the background information

*This work was partially done while Lan Du was at Macquarie University.

Lan Du, Anish Kumar, Mark Johnson and Massimiliano Ciaramita. 2015. Using Entity Information from a Knowledge Base to Improve Relation Extraction . In *Proceedings of Australasian Language Technology Association Workshop*, pages 31−38.

about entities available in FreeBase (Bollacker et al., 2008).

Here we focus on one particular kind of background information about entities — the information encoded in FreeBase's *notable types*. FreeBase's notable types are simple atomic labels given to entities that indicate what the entity is notable for, and so serve as a useful information source that should be relatively easy to exploit. For example, the search results for "Jim Jones" given by FreeBase contains several different entities. Although they all have the same name entity (NE) category PERSON, their notable types are different. The notable types for the top 4 "Jim Jones" results are *organization/organization_founder*, *music/composer*, *baseball/baseball_player* and *government/politician*. It is clear that the notable type information provides much finer-grained information about "Jim Jones" than just the NE category. It is reasonable to expect that notable types would be useful for relation extraction; e.g., the politician Jim Jones is likely to stand for election, while the baseball player is likely to be involved in sport activities.

We extend one state-of-the-art relation extraction system of Riedel et al. (2013) to exploit this notable type information. Our notable type extensions significantly improve the mean averaged precision (MAP) by 7.5% and the weighted MAP by 6% over a strong state-of-the-art baseline. With a set of ablation experiments we further evaluate how and where the notable type information contributes to relation extraction.The rest of this paper is structured as follows. The next section describes related work on relation extraction. Section 3 describes how a state-of-the-art relation extraction system can be extended to exploit the notable type information available in FreeBase. Section 4 specifies the inference procedures used to identify the values of the model parameters, while section 5 explains how we evaluate our models and presents a systematic experimental comparison of the models by ablating the notable type in different ways based on entities' NE categories. Section 6 concludes the paper and discusses future work.

2 Related work

Most approaches to relation extraction are either supervised or semi-supervised. Supervised approaches require a large set of manually annotated text as training data (Culotta and Sorensen, 2004), but creating these annotations is both expensive and error-prone. Semi-supervised approaches, by contrast, rely on correlations between relations and other large data sources.

In relation extraction, most semi-supervised approaches use *distant supervision*, which aligns facts from a large database, e.g., Freebase, to unlabelled text by assuming some systematic relationship between the documents and the database (Bunescu and Mooney, 2007; Mintz et al., 2009; Riedel et al., 2010; Yao et al., 2010). Typically, we assume that (a) an entity linker can reliably identify entity mentions in the text and map them to the corresponding database entries, and (b) for all tuples of entities that appear in a relation in the database, if we observe that entity tuple co-occurring in a suitable linguistic construction (e.g., a sentence) then that construction expresses the database relationship about those entities. Previous work (Weston et al., 2013; Riedel et al., 2013; Bordes et al., 2013; Chang et al., 2014) has shown that models leveraging rich information from database often yield improved performance.

In this work we are particularly interested in exploring entity type information in relation extraction, as semantic relations often have selectional preference over entity types. Yao et al. (2010), Singh et al. (2013), Yao et al. (2013), Koch et al. (2014) and Chang et al. (2014) have shown that the use of type information, e.g., NE categories, significantly improves relation extraction. Our work here is similar except that we rely on Freebase's notable types, which provide much finer-grained information about entities. One of the challenges in relation extraction, particularly when attempting to extract a large number of relations, is to generalise appropriately over both entities and relations. Techniques for inducing *distributed vector-space representations* can learn embeddings of both entities and relations in a high-dimensional vector space, providing a natural notion of similarity (Socher et al., 2013) that can be exploited in the relation extraction task (Weston et al., 2013). Instead of treating notable types as features Ling and Weld (2012), here we learn distributed vector-space representations for notable types as well as entities, entity tuples and relations.

3 Relation extraction as matrix completion

Riedel et al. (2013) formulated the relation extraction task as a matrix completion problem. In this section we extend this formulation to exploit notable types in a simple and effective way. Specifically, we follow Riedel et al. (2013) in assuming that our data $\mathcal{O}$ consists of pairs $\langle r, t \rangle$, where $r \in \mathcal{R}$ is a relation and $t \in \mathcal{T}$ is a tuple of entities. The tuples are divided into training and test depending on which documents they are extracted from. In this paper, the tuples in $\mathcal{T}$ are always pairs of entities, but nothing depends on this. There are two kinds of relations in $\mathcal{R}$: syntactic patterns found in the document collection, and those appearing in the database (including target relations for extraction). For our notable type extension we assume we have a function n that maps an entity e to its FreeBase notable type $n(e)$.

For example, given text "Michael Bay, the director of Transformers, visited Paris yesterday" we extract the pair $\langle r, t \rangle$ where $t = \langle$*Michael Bay*, *Transformers*$\rangle$ and $r =$ *X-the-director-of-Y* (actually, the path in a dependency parse between the named entities). From FreeBase we extract the pair $\langle r', t \rangle$ where $r' =$ *film/director*. FreeBase also tells us that $n($*Michael Bay*$) =$ *Person* and $n($*Transformers*$) =$ *Film*. Our goal is to learn a matrix $\mathbf{\Theta}$ whose rows are indexed by entity tuples in $\mathcal{T}$ and whose columns are indexed by relations in $\mathcal{R}$. The entry $\theta_{t,r}$ is the *log odds of relation* $r \in \mathcal{R}$ *holding of tuple* $t \in \mathcal{T}$, or, equivalently, the probability that relation r holds of tuple t is given by $\sigma(\theta_{t,r})$, where σ is the logistic function: $\sigma(x) = (1 + e^{-x})^{-1}$.

Riedel et al. (2013) assume that $\mathbf{\Theta}$ is the sum of three submodels: $\mathbf{\Theta} = \mathbf{\Theta}^{\mathrm{N}} + \mathbf{\Theta}^{\mathrm{F}} + \mathbf{\Theta}^{\mathrm{E}}$, where $\mathbf{\Theta}^{\mathrm{N}}$ is the neighbourhood model, $\mathbf{\Theta}^{\mathrm{F}}$ is the latent feature model and $\mathbf{\Theta}^{\mathrm{E}}$ is the entity model (these will be defined below). Here we extend these submodels using FreeBase's notable types.

3.1 A notable type extension to the neighbourhood model

The neighbourhood model $\mathbf{\Theta}^{\mathrm{N}}$ captures dependencies between the syntactic relations extracted from the text documents and the database relations extracted from FreeBase. This is given by:

$$\theta^{\mathrm{N}}_{r,t} = \sum_{\langle r',t \rangle \in \mathcal{O} \setminus \{\langle r,t \rangle\}} w_{r,r'},$$

where $\mathcal{O}$ is the set of relation/tuple pairs in the data and $\mathcal{O} \setminus \{\langle r, t \rangle\}$ is $\mathcal{O}$ with the tuple $\langle r, t \rangle$ removed. $\boldsymbol{w}$ is a matrix of parameters, where $w_{r,r'}$ is a real-valued weight with which relation r' "primes" relation r that will be learnt from the training data. The neighbourhood model can be regarded as predicting an entry $\theta_{r,t}$ by using entries along the same row. It functions as a logistic regression classifier predicting the log odds of a FreeBase relation r applying to the entity tuple t using as features the syntactic relations r' that hold of t.

Our notable type extension to the neighbourhood model enriches the syntactic patterns in the training data $\mathcal{O}$ with notable type information. For example, if there is a syntactic pattern for *X-director-of-Y* in our training data (say, as part of the tuple $\langle$*X-director-of-Y*, $\langle$*Michael Bay*, *Transformers*$\rangle\rangle$), then we add a new syntactic pattern $\langle$*Person(X)-director-of-Film(Y)*$\rangle$, where *Person* and *Film* are notable types and add the tuple $\langle$*Person(X)-director-of-Film(Y)*, $\langle$*Michael Bay*, *Transformers*$\rangle\rangle$ to our data $\mathcal{O}$. Each new relation corresponds to a new column in our matrix completion formulation. More precisely, the new relations are members of the set $\mathcal{N} = \{\langle r, n(t) \rangle : \langle r, t \rangle \in \mathcal{O}\}$, where $n(t)$ is the tuple of notable types corresponding to the entity tuple t. For example, if $t = \langle$*Michael Bay*, *Transformers*$\rangle$ then $n(t) = \langle$*Person*, *Film*$\rangle$. Then the notable type extension of the neighbourhood model is:

$$\theta^{\mathrm{N}'}_{r,t} = \sum_{\langle r',t \rangle \in \mathcal{O} \setminus \{\langle r,t \rangle\}} w_{r,r'} + w'_{r,\langle r',n(t) \rangle}$$

where $\boldsymbol{w}'$ is a matrix of weights relating the relations $\mathcal{N}$ to the target FreeBase relation r.

3.2 A notable type extension to the latent feature model

The latent feature model generalises over relations and entity tuples by associating each of them with a 100-dimensional real-valued vector. Intuitively, these vectors organise the relations and entity tuples into clusters where conceptually similar relations and entity tuples are "close," while those that are dissimilar are far apart. In more detail, each relation $r \in \mathcal{R}$ is associated with a latent feature vector $\boldsymbol{a}_r$ of size $K = 100$. Similarly, each entity tuple $t \in \mathcal{T}$ is also associated with a latent feature vector $\boldsymbol{v}_t$ of size K as well. Then the latent feature score for an entity tuple t and relation r is just the dot product of the corresponding relation and entity tuple vectors, i.e.: $\theta^{\mathrm{F}}_{r,t} = \boldsymbol{a}_r \cdot \boldsymbol{v}_t$.

We extend the latent feature model by associating a new latent feature vector with each notable type sequence observed in the training data, and use this vector to enrich the vector-space representations of the entity tuples. Specifically, let $\mathcal{T}' = \{n(t) : t \in \mathcal{T}\}$ be the set of notable type tuples for all of the tuples in $\mathcal{T}$, where $n(t)$ is the tuple of notable types corresponding to the tuple of entities t as before. We associate each tuple of notable types $t' \in \mathcal{T}'$ with a latent feature vector $v'_{t'}$ of dimensionality K. Then we define the notable type extension to the latent feature model as:

$$\theta^{\mathrm{F}'}_{r,t} \;=\; \boldsymbol{a}_r \cdot (\boldsymbol{v}_t + \boldsymbol{v}'_{n(t)})\,.$$

This can be understood as associating each entity tuple $t \in \mathcal{T}$ with a pair of latent feature vectors $\boldsymbol{v}_t$ and $\boldsymbol{v}_{n(t)}$. The vector $\boldsymbol{v}_{n(t)}$ is based on the notable types of the entities, so it can capture generalisations over those notable types. The L_2 regularisation employed during inference prefers latent feature vectors in which $\boldsymbol{v}_t$ and $\boldsymbol{v}'_{n(t)}$ are small, thus encouraging generalisations which can be stated in terms of notable types to be captured by $\boldsymbol{v}'_{n(t)}$.

3.3 A notable type extension of the entity model

The entity model represents an entity e with a K-dimensional ($K = 100$) feature vector u_e. Similarly, the ith argument position of a relation r is also represented by a K-dimensional feature vector $d_{r,i}$. The entity model associates a score $\theta^{\mathrm{E}}_{r,t}$ with a relation $r \in \mathcal{R}$ and entity tuple $t \in \mathcal{T}$ as follows: $\theta^{\mathrm{E}}_{r,t} = \sum_{i=1}^{|t|} \boldsymbol{d}_{r,i} \cdot \boldsymbol{u}_{t_i}$, where $|t|$ is the arity of (i.e., number of elements in the entity tuple t), t_i is the ith entity in the entity tuple t, and $\boldsymbol{d}_{r,i}$ and $\boldsymbol{u}_{t_i}$ are K-dimensional vectors associated with the ith argument slot of relation r and the entity t_i respectively. The intuition is that the latent feature vectors of co-occurring entities and argument slots should be close to each other in the K-dimensional latent feature space, while entities and argument slots that do not co-occur should be far apart.

Our notable type extension of the entity model is similar to our notable type extension of the latent feature model. We associate each notable type m with a K-dimensional feature vector $\boldsymbol{u}'_m$, and use those vectors to define the entity model score. Specifically, the entity model score is defined as:

$$\theta^{\mathrm{E}'}_{r,t} \;=\; \sum_{i=1}^{|t|} \boldsymbol{d}_{r,i} \cdot \left(\boldsymbol{u}_{t_i} + \boldsymbol{u}'_{n(t_i)}\right)\,,$$

where $n(e)$ is the notable type for entity e and $|t|$ is the length of tuple t. The L2 regularisation again should encourage generalisations that can be expressed in terms of notable types to be encoded in the $\boldsymbol{u}'_{n(t_i)}$ latent feature vectors.

4 Inference for model parameters

The goal of inference is to identify the values of the model's parameters, i.e., $\boldsymbol{w}, \boldsymbol{a}, \boldsymbol{v}, \boldsymbol{d}$ and $\boldsymbol{u}$ in the case of the Riedel et al model, and these plus $\boldsymbol{w}', \boldsymbol{v}'$ and $\boldsymbol{u}'$ in the case of the notable type extensions. The inference procedure is inspired by Bayesian Personalised Ranking (Rendle et al., 2009). Specifically, while the true value of $\theta_{r,t}$ is unknown, it's reasonable to assume that if $\langle r, t^+\rangle \in \mathcal{O}$ (i.e., is observed in the training data) then $\theta_{r,t^+} > \theta_{r,t^-}$ for all $\langle r, t^-\rangle \notin \mathcal{O}$ (i.e., not observed in the training data). Thus the training objective is to maximise

$$\ell \;=\; \sum_{\langle r,t^+\rangle \in \mathcal{O}} \; \sum_{\langle r,t^-\rangle \notin \mathcal{O}} \ell_{\langle r,t^+\rangle, \langle r,t^-\rangle}$$

where: $\ell_{\langle r,t^+\rangle,\langle r,t^-\rangle} = \log \sigma(\theta_{r,t^+} - \theta_{r,t^-})$, and $\theta_{r,t} = \theta^{\mathrm{N}}_{r,t} + \theta^{\mathrm{F}}_{r,t} + \theta^{\mathrm{E}}_{r,t}$ or $\theta_{r,t} = \theta^{\mathrm{N}'}_{r,t} + \theta^{\mathrm{F}'}_{r,t} + \theta^{\mathrm{E}'}_{r,t}$, depending on whether the submodels with notable type extensions are used. The objective function ℓ is then maximised by using stochastic gradient ascent. The stochastic gradient procedure sweeps through the training data, and, for each observed tuple $\langle r, t^+\rangle \in \mathcal{O}$, samples a negative evidence tuple $\langle r, t^-\rangle \notin \mathcal{O}$ not in the training data, adjusting weights to prefer the observed tuple.

In our experiments below we ran stochastic gradient ascent with a step size of 0.05 and an L2 regulariser constant of 0.1 for the neighbourhood model and 0.01 for the latent feature and entity models (we used the same regulariser constants for models both with and without the notable type extensions). We ran 2,000 sweeps of stochastic gradient ascent.

5 Experimental evaluation

We used a set of controlled experiments to see to what extent the notable type information improves the state-of-the-art relation extraction system. We used the New York Times corpus (Sandhaus, 2008) in our experiments, assigning articles from the year 2000 as the training corpus and the articles from 1990 to 1999 for testing. The entity tuples $\mathcal{T}$ were extracted from the New York Times corpus (tuples that did not appear at least 10 times and also appear in one of the FreeBase relations were discarded). The relations $\mathcal{R}$ are either syntactic patterns found in the New York Times corpus, FreeBase relations, or (in our extension) notable types extracted from FreeBase. Our evaluation focuses on 19 FreeBase relations, as in Riedel et al. (2013).

5.1 Notable type identification

Our extension requires a FreeBase notable type for every entity mention, which in turn requires a Freebase entity id because a notable type is a property associated with entities in FreeBase. We found the entity id for each named entity as follows. We used the FreeBase API to search for the notable type for each named entity mentioned in the training or test data. In cases where several entities were returned, we used the notable type of the first entity returned by the API. For example, the FreeBase API returns two entities for the string "Canada:" a country and a wine (in that order), so we use the notable type "country" for "Canada" in our experiments. This heuristic is similar to the method of choosing the most likely entity id for a string, which provides a competitive baseline for entity linking (Hoffart et al., 2011).

5.2 Evaluation procedure

After the training procedure is complete and we have estimates for the model's parameters, we can use these to compute estimates for the log odds $\theta_{r,t}$ for the test data. These values quantify how likely it is that the FreeBase relation r holds of an entity tuple t from the test set, according to the trained model.

In evaluation we follow Riedel et al. (2013) and treat each of the 19 relations r as a query, and evaluate the ranking of the entity tuples t returned according to $\theta_{r,t}$. For each relation r we pool the highest-ranked 100 tuples produced by each of the models and manually evaluate their accuracy (e.g., by inspecting the original document if necessary). This gives a set of results that can be used to calculate a precision-recall curve. Averaged precision (AP) is a measure of the area under that curve (higher is better), and mean average precision (MAP) is average precision averaged over all of the relations we evaluate on. Weighted MAP is a version of MAP that weights each relation by the true number of entity tuples for that relation (so more frequent relations count more).

An unusual property of this evaluation is that increasing the number of models being evaluated generally decreases their MAP scores: as we evaluate more models, the pool of "true" entity tuples for each relation grows in size and diversity (recall that this pool is manually constructed by manually annotating the highest-scoring tuples returned by each model). Thus in general the recall scores of the existing models are lowered as the number of models increases.

Relation	#	NF	NF^T	NFE	NFE^T
person/company	131	0.83	0.89	0.83	0.86
location/containedby	88	0.68	0.69	0.68	0.69
person/nationality	51	0.11	0.55	0.15	0.45
author/works_written	38	0.51	0.53	0.57	0.53
person/parents	34	0.14	0.31	0.11	0.28
parent/child	31	0.48	0.58	0.49	0.58
person/place_of_birth	30	0.51	0.48	0.56	0.57
person/place_of_death	22	0.75	0.77	0.75	0.77
neighbourhood/neighbourhood_of	17	0.48	0.55	0.52	0.54
broadcast/area_served	8	0.21	0.41	0.26	0.30
company/founders	7	0.46	0.27	0.40	0.28
team_owner/teams_owned	6	0.21	0.25	0.25	0.25
team/arena_stadium	5	0.06	0.07	0.06	0.09
film/directed_by	5	0.21	0.25	0.24	0.35
person/religion	5	0.20	0.28	0.21	0.23
composer/compositions	4	0.42	0.44	0.06	0.42
sports_team/league	4	0.70	0.62	0.63	0.64
film/produced_by	3	0.17	0.30	0.12	0.26
structure/architect	2	*1.00*	*1.00*	*1.00*	*1.00*
MAP		0.43	**0.49**	0.42	0.48
Weighted MAP		0.55	**0.64**	0.56	0.62

Table 1: Averaged precision and mean average precision results. The rows correspond to FreeBase relations, and the columns indicate the combination of sub-models (N = neighbourhood model, F = latent feature model, E = entity model). The superscript "T" indicates the combined models that incorporate the notable type extensions, and the # column gives the number of true facts.

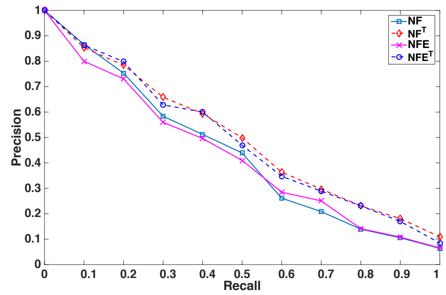

Figure 1: Averaged 11-point precision-recall curve for the four models shown in Table 1.

5.3 Experiments with Notable Types

We found we obtained best performance from the model that incorporates all submodels (which we call NFE^T) and from the model that only incorporates the Neighbourhood and Latent Feature submodels (which we call NF^T), so we concentrate on them here. Table 1 presents the MAP and weighted MAP scores for these models on the 19 FreeBase relations in the testing set.

The MAP scores are 6% higher for both NF^T and NFE^T, and the weighted MAP scores are 9% and 6% higher for NF^T and NFE^T respectively.

Relation	#	NE	NFE^T	NE+P	NE+L	NE+O	NE+M
person/place_of_birth	30	0.52	**0.57**	0.54	0.50	0.50	0.54
author/works_written	38	0.57	0.53	**0.61**	0.56	0.57	0.49
team/arena_stadium	5	0.08	0.09	**0.10**	0.09	0.07	0.09
composer/compositions	4	0.35	0.42	**0.51**	0.37	0.35	0.45
person/company	131	0.81	**0.86**	0.84	0.82	0.83	**0.86**
film/directed_by	5	0.30	0.35	**0.41**	0.27	0.27	**0.41**
neighbourhood/neighbourhood_of	17	0.59	0.54	0.59	0.49	0.59	**0.62**
film/produced_by	3	0.20	0.26	0.29	0.18	0.19	**0.40**
person/religion	5	0.22	0.23	0.21	0.22	0.28	**0.53**
location/containedby	88	0.66	0.69	0.68	0.64	0.64	**0.70**
sports_team/league	4	0.53	0.64	0.54	0.52	**0.75**	**0.75**
person/parents	34	0.33	0.28	0.30	0.32	**0.35**	0.34
parent/child	31	0.55	0.58	0.56	0.55	**0.59**	0.56
person/place_of_death	22	0.71	0.77	0.74	0.74	**0.78**	0.72
company/founders	7	0.22	0.28	0.28	0.21	**0.29**	0.22
team_owner/teams_owned	6	0.34	0.25	0.27	0.34	**0.36**	0.35
person/nationality	51	0.19	0.45	0.23	**0.50**	0.20	0.21
broadcast/area_served	8	0.32	0.30	0.33	**0.38**	0.31	0.29
structure/architect	2	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*
MAP		0.45	0.48	0.48	0.46	0.47	**0.50**
Weighted MAP		0.57	**0.62**	0.59	0.60	0.58	0.60

Table 2: Results of ablation experiments on the NFE^T model. The columns correspond to experiments, and the column labels are explained in Table 3.

A sign test shows that the difference between the models with notable types and those without the notable types is statistically significant ($p < 0.05$). Clearly, the notable type extensions significantly improve the accuracy of the existing relation extraction models. Figure 1 shows an averaged 11-point precision-recall curve for these four models. This makes clear that across the range of precision-recall trade-offs, the models with notable types offer the best performance.

5.4 Ablation Experiments

We performed a set of ablation experiments to determine exactly how and where the notable type information improves relation extraction. In these experiments entities are divided into 4 "named entity" (NE) classes, and we examine the effect of just providing notable type information for the entities of a single NE class. The 4 NE classes we used were PERSON, LOCATION, ORGANISATION, and MISC (miscellaneous). We classified all entities into these four categories using their FreeBase types, which provide a more coarse-grained classification than notable types. For example, if an entity has a FreeBase "*people/person*" type, then we assigned it to the NE class PERSON; if an entity has a "*location/location*" type, then its NE class is LOCATION; and if an entity has a "*organisation/organisation*" type, then its NE class is ORGANISATION. All entities not classified as PERSON, LOCATION, or ORGANISATION were labelled MISC.

We ran a set of ablation experiments as follows. For each NE class c in turn, we replaced the notable type information for entities not classified as c with their NE class. For example, when c = PERSON, only entities with the NE label PERSON had notable type information, and the notable types of all other entities was replaced with their NE labels. Table 3 lists the different ablation experiments. The ablation experiments are designed to study which NE classes the notable types help most on. The results are reported in Table 2. The results clearly indicate that different relations benefit from the different kinds of notable type information about entities.

Ablation setting	Description
NE	All entities are labelled with their NE class instead of their notable type.
NE+P	Only PERSON entities have notable type information; the notable type of other entities is replaced with their NE class.
NE+L	Only LOCATION entities have notable type information; the notable type of other entities is replaced with their NE class.
NE+O	Only ORGANISATION entities have notable type information; the notable type of other entities is replaced with their NE class.
NE+M	Only MISC entities have notable type information; the notable type of other entities is replaced with their NE class.

Table 3: Descriptions of the ablation experiments in Table 2.

Column "NE+P" shows that relations such as "*author/works_written*", "*composer/compositions*" and "*film/directed_by*" benefit the most from notable type information about PERSONs. We noticed that there are about 43K entities classified as PERSON, which includes 8,888 book authors, 802 music composers, 1212 film directors, etc. These entities have 214 distinct notable types. Our results show that it is helpful to distinguish the PERSON entities with their notable types for relations involving professions. For example, not all people are authors, so knowing that a person is an author increases the accuracy of extracting "*author/works_written*". Similarly, Column "NE+L" shows that "*person/nationality*" and "*broadcast/area_served*" gain the most from the notable type information about locations. There are about 8.5K entities classified as LOCATION, which includes 4807 city towns, 301 countries, and so on. There are 170 distinct notable types for LOCATION entities.

Column "NE+O" shows that the notable type information about ORGANISATION entities improves the accuracy of extracting relations involving organisations. Indeed, there are more than

3K business companies and 200 football teams. Notable type information about organisations improves extraction of the "*parent/child*" relation because this relation involves entities such as companies. For example, in our corpus the sentence "CNN, a unit of the Turner Broadcasting, says that 7000 schools have signed up for The Newsroom" expresses the *parent/child(Turner Broadcasting, CNN)* relation.

The ablation results in Column "NE+M" show that information about MISC entities is most useful of all, as this ablation experiment yielded the highest overall MAP score. There are about 13.5K entities labelled MISC. The most frequent notable types for entities in the MISC NE class are "*film/film*" and "*book/book*". Therefore it is reasonable that notable type information for MISC entities would improve AP scores for relations such as "*film/directed_by*" and "*person/religion*". For example, "George Bush reached a turning point in his life and became a born-again Christian" is an example of the "*person/religion*" relation, and it's clear that it is useful to know that "born-again Christian" belongs to the religion notable type. The "sports_team/league" relation is interesting because it performs best with notable type information for entities in the ORGANISATION or MISC NE classes. It turns out that roughly half the sports teams are classified as ORGANISATIONs and half are classified as MISC. The sports teams that are classified as MISC are missing the "organisation/organisation" type in their FreeBase entries, otherwise they would be classified as ORGANISATIONs.

In summary, the ablation results show that the contribution of notable type information depends on the relation being extracted. The result demonstrates that relations involving organisations benefits from the notable type information about these organisations. It also demonstrates that certain relations benefit more from notable type information than others. Further research is needed understand some of the ablation experiment results (e.g., why does person/place of death perform best with notable type information about ORGANISATIONs?)

6 Conclusion and future work

In this paper we investigated the hypothesis that background information about entities present in a large database such as FreeBase can be useful for relation extraction. We modified a state-of-the-art relation extraction system (Riedel et al., 2013) by extending each of its submodels to exploit the "notable type" information about entities available in FreeBase. We demonstrated that these extensions improve the MAP score by 6% and the weighted MAP score by 7.5%, which is a significant improvement over a strong baseline. Our ablation experiments showed that the notable type information improves relation extraction more than NER tags across a wide range of entity types and relations.

In future work we would like to develop methods for exploiting other information available in FreeBase to improve a broad range of natural language processing and information extraction tasks. We would like to explore ways of exploiting entity information beyond (distant) supervision approaches, for example, in the direction of OpenIE (Wu and Weld, 2010; Fader et al., 2011; Mausam et al., 2012). The temporal information in a large database like FreeBase might be especially useful for named entity linking and relation extraction: e.g., someone that has died is less likely to release a hit single. In summary, we believe that there are a large number of ways in which the rich and diverse information present in FreeBase might be leveraged to improve natural language processing and information retrieval, and exploiting notable types is just one of many possible approaches.

Acknowledgments

This research was supported by a Google award through the Natural Language Understanding Focused Program, and under the Australian Research Council's *Discovery Projects* funding scheme (project numbers DP110102506 and DP110102593).

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 576–583.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1579.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 423–429.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*, pages 1535–1545.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)* , pages 782–792.

Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1891–1901.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *AAAI*.

Yudong Liu, Zhongmin Shi, and Anoop Sarkar. 2007. Exploiting rich syntactic information for relation extraction from biomedical articles. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-Short'07, pages 97–100.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. 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 1003–1011.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III*, ECML PKDD'10, pages 148–163.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Evan Sandhaus. 2008. The new york times annotated corpus. Linguistic Data Consortium, Philadelphia.

Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, pages 1–6.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems (NIPS 26)*, pages 926–934.

Michael Spivey-Knowlton and Julie Sedivy. 1995. Resolving attachment ambiguities with multiple constraints. *Cognition*, 55:227–267.

Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371.

Fei Wu and Daniel S. Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL'10, pages 118–127.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, pages 79–84.