Mapping Verbs In Different Languages to Knowledge Base Relations using Web Text as Interlingua

Derry Tanti Wijaya
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213
dwijaya@cs.cmu.edu

Tom M. Mitchell
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213
tom.mitchell@cs.cmu.edu

Abstract

In recent years many knowledge bases (KBs) have been constructed, yet there is not yet a verb resource that maps to these growing KB resources. A resource that maps verbs in different languages to KB relations would be useful for extracting facts from text into the KBs, and to aid alignment and integration of knowledge across different KBs and languages. Such a multi-lingual verb resource would also be useful for tasks such as machine translation and machine reading. In this paper, we present a scalable approach to automatically construct such a verb resource using a very large web text corpus as a kind of interlingua to relate verb phrases to KB relations. Given a text corpus in any language and any KB, it can produce a mapping of that language's verb phrases to the KB relations. Experiments with the English NELL KB and ClueWeb corpus show that the learned English verb-to-relation mapping is effective for extracting relation instances from English text. When applied to a Portuguese NELL KB and a Portuguese text corpus, the same method automatically constructs a verb resource in Portuguese that is effective for extracting relation instances from Portuguese text.

1 Introduction

In recent years a variety of large knowledge bases (KBs) have been constructed e.g., Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007), NELL (Carlson et al., 2010), and Yago (Suchanek et al., 2007). These KBs consist of (1) an ontology that defines a set of categories (e.g., *SportsTeam*, *City*), (2) another part of the ontology that defines relations with these categories as argument types (e.g., ***teamPlaysInCity****(SportsTeam, City)*), (3) KB entities which instantiate these categories (e.g., *Steelers* $\in$ *SportsTeam*), and (4) KB entity pairs which instantiate these relations (e.g., (*Steelers*, *Pittsburgh*) $\in$ ***teamPlaysInCity***). The KB ontology also specifies constraints (e.g., mutual exclusion, subset) among KB categories and relations.

Despite recent progress in KB construction, there is not yet a verb resource that maps to these KBs: one that contains verb phrases[1] that identify KB relations. Such a verb resource can be useful to aid KB relation extraction. A distribution of verb phrases associated with any given KB relation is also a KB-independent representation of that relation's semantics which can form the basis of aligning ontologies across arbitrary KBs (Wijaya et al., 2013). Given a KB and verb resources in different languages that map to the KB, we can also begin to align knowledge expressed in different languages.

We introduce here an approach to mapping verb phrases to KB relations using a very large ClueWeb corpus (Callan et al., 2009) as a kind of interlingua. Our approach grounds each KB relation instance (e.g., ***teamPlaysInCity****(Steelers, Pittsburgh)*) in mentions of its argument pair in this text, then represents the relation in terms of the verb phrases that connect these paired mentions (see Fig. 1). For a high coverage mapping, we train on both labelled and unlabelled data using expectation maximization (EM). We introduce argument type checking during

[1] In this paper we use the term "verb phrase" and "verb" interchangeably; both referring to either verb or verb+preposition

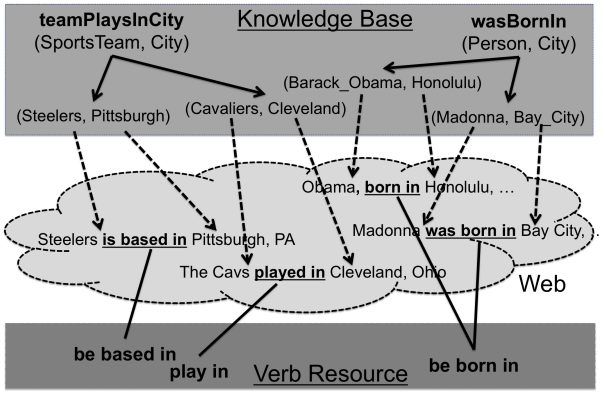

Figure 1: Mapping verb phrases to relations in KB through Web-text as interlingua. Each relation instance is grounded by its mentions in the Web-text. The verbs that co-occur with mentions of the relation's instances are mapped to that relation.

the EM process to ensure only verbs whose argument types match the relation's argument types are mapped to the relation. We also incorporate constraints defined in the KB ontology to find a verb to relation mapping consistent with these constraints.

Our contributions are: (1) We propose a scalable EM-based method that automatically maps verb phrases to KB relations by using the mentions of the verb phrases with the relation instances in a very large unlabeled text corpus. (2) We demonstrate the effectiveness of the resource for extracting relation instances in NELL KB. Specifically, it improves the recall of both the supervised- and the unsupervised- verb-to-relation mapping; demonstrating the benefit of semi-supervised learning on unlabeled Web-scale text. (3) We demonstrate the flexibility of the method, which is both KB- and language-independent, by using the same method for constructing English verb resource to automatically construct a Portuguese verb resource. (4) We make our verb resources publicly available [2].

2 Method

2.1 Terminology

We define a NELL KB to be a 6-tuple $(C, I_C, R, I_R, Subset, Mutex)$. C is the set of categories e.g., *SportsTeam* i.e., $c_j \in C = \{c_1, ..., c_{|C|}\}$. I_C is the set of category instances which are entity-category pairs e.g., (*Cleveland*, *City*) i.e., $I_C = \{(e_m, c_j) \mid e_m \in c_j,\ c_j \in C\}$.

R is the set of relations e.g., ***teamPlaysInCity*** i.e., $r_i \in R = \{r_1, ..., r_{|R|}\}$. We also define f_{type} to be a function that when applied to a relation r_i returns the argument type signature of the relation $f_{type}(r_i) = (c_j, c_k)$ for some $c_j, c_k \in C$ e.g., f_{type}(***teamPlaysInCity***) = (*SportsTeam*, *City*).

I_R is the set of relation instances which are entity-relation-entity triples e.g., *(Cavaliers,* ***teamPlaysInCity****, Cleveland)* i.e., $I_R = \{(e_m,r_i,e_n) \mid (e_m, e_n) \in r_i,\ r_i \in R,\ e_m \in c_j,\ e_n \in c_k,\ f_{type}(r_i) = (c_j, c_k)\}$; $I_R = I_{r_1} \cup I_{r_2} \cup ... I_{r_{|R|}}$.

$Subset$ is the set of all subset constraints among relations in R i.e., $Subset = \{(i,k) : I_{r_i} \subseteq I_{r_k}\}$. For example {*(person,* ***ceoOf****, company)*} $\subseteq$ {*(person,* ***worksFor****, company)*}.

$Mutex$ is the set of all mutual exclusion constraints among relations in R i.e., $Mutex = \{(i,k) : I_{r_i} \cap I_{r_k} = \phi\}$. For example {*(drug,* ***hasSideEffect****, physiologicalCondition)*} $\cap$ {*(drug,* ***possiblyTreats****, physiologicalCondition)*} = ϕ.

Each KB entity e_m can be referred to by one or more noun phrases (NPs). For example, the entity *Cavaliers*, can be referred to in text using either the NP *"Cleveland Cavaliers"* or the NP *"The Cavs"*[3]. We define $N_{en}(e_m)$ to be the set of English NPs corresponding to entity e_m.

We define SVO to be the English Subject-Verb-Object (SVO) interlingua[4] consisting of tuples of the form (np_s, v_p, np_o, w), where np_s and np_o are noun phrases (NP) corresponding to subject and object, respectively, v_p is a verb phrase that connects them, and w is the count of the tuple.

2.2 Data Construction

We construct a dataset D for mapping English verbs to NELL KB relations. First, we convert each tuple in SVO to its equivalent entity pair tuple(s) in $SVO' = \{(e_m, v_p, e_n, w) \mid np_s \in N_{en}(e_m),\ np_o \in N_{en}(e_n),\ (np_s,\ v_p,\ np_o,\ w) \in SVO\}$. Then, we construct D from SVO' as a collection of labeled and unlabeled instances.

[2] http://www.cs.cmu.edu/%7Edwijaya/mapping.html

[3] defined by the ***canReferTo*** relation in NELL KB

[4] We use 600 million SVO triples collected from the entire ClueWeb (Callan et al., 2009) of about 230 billion tokens with some filtering described in Section 3.1.

The set of labeled instances is $D^\ell = \{(\mathbf{y}_{(e_m,e_n)}, \mathbf{v}_{(e_m,e_n)})\}$ where $\mathbf{y}_{(e_m,e_n)} \in \{0,1\}^{|R|}$ is a bit vector of label assignment, each bit representing whether the instance belongs to a particular relation i.e., $y^i_{(e_m,e_n)} = 1 \iff (e_m, e_n) \in r_i$ and 0 otherwise. $\mathbf{v}_{(e_m,e_n)} \in \mathbb{R}^{|V|}$ is a $|V|$-dimensional vector of verb phrase counts that connect e_m and e_n in SVO' (V is the set of all verb phrases) i.e., $v^p_{(e_m,e_n)}$ is the number of times the verb phrase v_p connects e_m and e_n in SVO'.

The collection of unlabeled instances is constructed from entity pairs in SVO' whose label assignment $\mathbf{y}$ are unknown (its bits are all zero) i.e., $D^u = \{(\mathbf{y}_{(e_m,e_n)}, \mathbf{v}_{(e_m,e_n)}) \mid (e_m, *, e_n, *) \in SVO', (e_m, *, e_n) \notin I_R\}$.

An instance in our dataset $d_{(e_m,e_n)} \in D$ is therefore either a labeled or unlabeled tuple i.e., $d_{(e_m,e_n)} = (\mathbf{y}_{(e_m,e_n)}, \mathbf{v}_{(e_m,e_n)})$.

We let $f_{type}(d_{(e_m,e_n)})$ return the argument type of the instance i.e., $f_{type}(d_{(e_m,e_n)}) = (c_j, c_k)$ where (e_m, c_j) and $(e_n, c_k) \in I_C$.

We let $f_{verb}(d_{(e_m,e_n)})$ return the set of all verb phrases that co-occur with the instance in SVO' i.e., $f_{verb}(d_{(e_m,e_n)}) = \{v_p \mid (e_m, v_p, e_n, *) \in SVO'\}$.

When applied to a relation r_i, we let $f_{verb}(r_i)$ return the set of all verb phrases that co-occur with instances in D whose types match that of the relation i.e., $f_{verb}(r_i) = \{v_p \mid \exists\ d_{(e_m,e_n)} \in D,\ v_p \in f_{verb}(d_{(e_m,e_n)}),\ f_{type}(d_{(e_m,e_n)}) = f_{type}(r_i)\}$.

2.3 Model

We train a Naive Bayes classifier on our dataset. Given as input a collection D^ℓ of labeled instances and D^u of unlabeled instances, it outputs a classifier, $\hat{\theta}$, that takes an unlabeled instance and predicts its label assignment i.e., for each unlabeled instance $d_{(e_m,e_n)} \in D^u$ the classifier predicts the label assignment $\mathbf{y}_{(e_m,e_n)}$ using $\mathbf{v}_{(e_m,e_n)}$ as features:

$$
\begin{aligned}
&P(y^i_{(e_m,e_n)} = 1 \mid d_{(e_m,e_n)};\ \hat{\theta}) \\
&\qquad = \frac{P(r_i|\hat{\theta}) P(d_{(e_m,e_n)} \mid r_i; \hat{\theta})}{P(d_{(e_m,e_n)}|\hat{\theta})} \\
&\qquad = \frac{P(r_i|\hat{\theta}) \prod_{p=1}^{|V|} P(v_p|r_i;\hat{\theta})^{v^p_{(e_m,e_n)}}}{\sum_{k=1}^{|R|} P(r_k|\hat{\theta}) \prod_{p=1}^{|V|} P(v_p|r_k;\hat{\theta})^{v^p_{(e_m,e_n)}}}
\end{aligned} \tag{1}
$$

If the task is to classify the unlabeled instance into a single relation, only the bit of the relation with the highest posterior probability is set i.e, $y^k_{(e_m,e_n)} = 1$ where $k = \arg\max_i P(y^i_{(e_m,e_n)} = 1 \mid d_{(e_m,e_n)}; \hat{\theta})$.

2.3.1 Parameter Estimation

To estimate model parameters (the relation prior probabilities $\hat{\theta}_{r_i} \equiv P(r_i|\hat{\theta})$ and probabilities of a verb given a relation $\hat{\theta}_{v_p|r_i} \equiv P(v_p|r_i;\hat{\theta})$) from both labeled and unlabeled data, we use an Expectation Maximization (EM) algorithm (Nigam et al., 2006). The estimates are computed by calculating a maximum a posteriori estimate of θ, i.e. $\hat{\theta} = \arg\max_\theta \mathcal{L}(\theta|D) = \arg\max_\theta log(P(D \mid \theta)P(\theta))$.

The first term, $P(D \mid \theta)$ is calculated by the product of all the instance likelihoods:

$$
\begin{aligned}
&P(D \mid \theta) \\
&\quad = \prod_{d_{(e_m,e_n)} \in D^u} \sum_{i=1}^{|R|} P(r_i|\theta) P(d_{(e_m,e_n)}|r_i;\theta) \\
&\quad \times \prod_{d_{(e_m,e_n)} \in D^\ell} \sum_{\{i|y^i_{(e_m,e_n)}=1\}} P(r_i|\theta) P(d_{(e_m,e_n)}|r_i;\theta)
\end{aligned} \tag{2}
$$

The second term, $P(\theta)$, the prior distribution over parameters is represented by Dirichlet priors: $P(\theta) \propto \prod_{i=1}^{|R|}((\theta_{r_i})^{\alpha_1 - 1} \prod_{p=1}^{|V|}(\theta_{v_p|r_i})^{\alpha_2 - 1})$ where α_1 and α_2 are parameters that effect the strength of the priors. In this paper we set $\alpha_1 = 2$ and $\alpha_2 = 1 + \sigma(P^e(v_p|r_i))$, where $P^e(v_p|r_i)$ is the initial bias of the verb-to-relation mapping. Thus, in this paper we define $P(\theta)$ as:

$$
P(\theta) = \prod_{i=1}^{|R|}(P(r_i|\theta) \prod_{p=1}^{|V|}(P(v_p|r_i;\theta)^{\sigma(P^e(v_p|r_i))}) \tag{3}
$$

We can see from this that $\sigma(P^e(v_p|r_i))$ is a conjugate prior on $P(v_p|r_i;\theta)$ with σ as the confidence parameter. This conjugate prior allows incorporation of any existing knowledge (Section 2.3.2) we may have about the verb-to-relation mapping.

From Equation 2, we see that $log\ P(D|\theta)$ contains a log of sums, which makes a maximization by partial derivatives computationally intractable. Using EM, we instead maximize the *expected* log likelihood of the data with respect to

the posterior distribution of the y labels given by: $\arg\max_\theta E_{(\mathbf{y}|D;\theta)}[log\ P(D|\theta)]$.

In the E-step, we use the current estimates of the parameters $\hat{\theta}^t$ to compute $\hat{\mathbf{y}}^t = E[\mathbf{y}|D;\hat{\theta}^t]$ the expected label assignments according to the current model. In practice it corresponds to calculating the posterior distribution over the y labels for unlabeled instances $P(y^i_{(e_m,e_n)} = 1 \mid d_{(e_m,e_n)};\hat{\theta}^t)$ (Equation 1) and using the estimates to compute its expected label assignment $\hat{\mathbf{y}}^t_{(e_m,e_n)}$.

In the M-step, we calculate a new maximum a posteriori estimate for $\hat{\theta}^{(t+1)}$ which maximizes the expected log likelihood of the complete data, $\mathcal{L}_c(\theta|D;\hat{\mathbf{y}}^t) = log(P(\theta^t)) + \ \hat{\mathbf{y}}^t\ [log\ P(D|\theta^t)]$:

$$\mathcal{L}_c(\theta|D;\hat{\mathbf{y}}^\mathbf{t}) = log(P(\theta^t)) + \sum_{d_{(e_m,e_n)}\in D}\ \sum_{i=1}^{|R|} y^{ti}_{(e_m,e_n)}\ log\ P(r_i|\theta)P(d_{(e_m,e_n)}|r_i;\theta) \tag{4}$$

$\mathcal{L}_c(\theta|D;\mathbf{y})$ bounds $\mathcal{L}(\theta|D)$ from below (by application of Jensen's inequality $E[log(X)] \leq log(EX)$). The EM algorithm produces parameter estimates $\hat{\theta}$ that correspond to a local maximum of $\mathcal{L}_c(\theta|D;\mathbf{y})$. The relation prior probabilities are thus estimated using current label assignments as:

$$P(r_i|\hat{\theta}^{(t+1)}) = \frac{1 + \sum_{d_{(e_m,e_n)}\in D} y^{ti}_{(e_m,e_n)}}{|R| + |D|} \tag{5}$$

The verb-to-relation mapping probabilities are estimated in the same manner:

$$P(v_p \mid r_i;\ \hat{\theta}^{(t+1)}) = \frac{\sigma_i^{(t+1)}\ (P^e(v_p \mid r_i)) \ + \sum_{d_{(e_m,e_n)}\in D} v^p_{(e_m,e_n)}\ y^{ti}_{(e_m,e_n)}}{\sigma_i^{(t+1)} \ + \sum_{s=1}^{|V|}\ \sum_{d_{(e_m,e_n)}\in D}\ v^s_{(e_m,e_n)}\ y^{ti}_{(e_m,e_n)}} \tag{6}$$

We start with $\sigma = |V|$ and gradually reduce the impact of prior by decaying σ with a decay parameter of 0.8 at each iteration in the manner of (Lu and Zhai, 2008)). This will allow the EM to gradually pick up more verbs from the data to map to relations.

EM iteratively computes parameters θ^1, ..., θ^t using the above E-step and M-step update rule at each iteration t, halting when there is no further improvement in the value of $\mathcal{L}_c(\theta|D;\mathbf{y})$.

2.3.2 Prior Knowledge

In our prior $P(\theta)$, we incorporate knowledge about verb-to-relation mappings from the text patterns learned by NELL to extract relations. This is our way of *aligning* our verb-to-relation mappings with NELL's current extractions. Coupled Pattern Learner (CPL) (Carlson et al., 2010) is a component in NELL that learns these contextual patterns for extracting instances of relations and categories.

We consider only CPL's extraction patterns that contain verb phrases. Given a set E_{r_i} of CPL's extraction patterns for a relation r_i, and E_{r_i,v_p} as the set of extraction patterns in E_{r_i} that contains the verb phrase v_p, we compute $P^e(v_p \mid r_i) = \frac{\mid E_{r_i,v_p} \mid}{\mid E_{r_i} \mid}$ and use them as priors in our classifier (Equation 3).[5]

2.3.3 Argument Type Checking

Although some verbs are ambiguous (e.g., the verb "play" may express several relations: ***musicianPlaysMusicalInstrument***, ***athletePlaysSport***, ***actorPlaysMovie***, etc), knowing the types of the verbs' arguments can help disambiguate the verbs (e.g., the verb "play" that takes a *musicalInstrument* type as object is more likely to express the ***musicianPlaysMusicalInstrument*** relation). Therefore, we incorporate argument type checking in our EM process to ensure that it maps verbs to relations whose argument types match:

- In the E-Step, we make sure that unlabeled instances are only labeled with relations that have the same argument types as the instance *and* that share some verbs with the instance. In other words, in the E-step we compute $P(y^i_{(e_m,e_n)} = 1 \mid d_{(e_m,e_n)})$ if $f_{type}(r_i) = f_{type}(d_{(e_m,e_n)})$ *and* $\left(f_{(verb)}(r_i) \cup \{v_p | E_{r_i,v_p} \neq \emptyset\}\right) \cap f_{(verb)}(d_{(e_m,e_n)}) \neq \emptyset$.
- In the M-step, we make sure that verbs are only mapped to relations whose argument types match at least one of the instances that co-occur with the verbs in SVO'. In other words, in the M-step we compute $P(v_p \mid r_i)$ if $v_p \in f_{verb}(r_i)$ or $E_{r_i,v_p} \neq \emptyset$.

[5] We manually add a few verb phrases for relations whose E_r is an empty set when possible, to set the EM process on these relations with good initial guesses of the parameters. In average, each relation has about 6 verb patterns in total as priors.

2.3.4 Incorporating Constraints

In the E-step, for each unlabeled instance, given the probabilities over relation labels $P(y^i_{(e_m,e_n)} = 1 \mid d_{(e_m,e_n)}; \hat{\theta}^t)$, and $Subset$ and $Mutex$ constraints[6], similar to (Dalvi et al., 2015), we use a Mixed-Integer Program (MIP) to produce its bit vector of label assignment as output: $\hat{\mathbf{y}}^t_{(e_m,e_n)}$.

The constraints among relations are incorporated as constraints on bits in this bit vector. For example, if for an unlabeled instance (*Jeff Bezos*, *Amazon*), a bit corresponding to the relation ***ceoOf*** is set then the bit corresponding to the relation ***worksFor*** should also be set due to the subset constraint: ***ceoOf*** $\subseteq$ ***worksFor***. For the same instance, the bit corresponding to ***competesWith*** should not be set due to the mutual exclusion constraint ***ceoOf*** $\cap$ ***competesWith*** = ϕ. The MIP formulation for each unlabeled instance thus tries to maximize the sum of probabilities of selected relation labels after penalizing for violation of constraints (Equation 7), where ζ_{ik} are slack variables for $Subset$ constraints and δ_{ik} are slack variables for $Mutex$ constraints:

$$\underset{\mathbf{y}_{(e_m,e_n)},\zeta_{ik},\delta_{ik}}{\text{maximize}} \left(\sum_{i=1}^{|R|} y^i_{(e_m,e_n)} \times P(y^i_{(e_m,e_n)} = 1 | d_{(e_m,e_n)}; \hat{\theta}^t) - \sum_{(i,k)\in Subset} \zeta_{ik} - \sum_{(i,k)\in Mutex} \delta_{ik} \right)$$

$$\text{subject to,}$$

$$y^i_{(e_m,e_n)} \leq y^k_{(e_m,e_n)} + \zeta_{ik}, \forall (i,k) \in Subset$$

$$y^i_{(e_m,e_n)} + y^k_{(e_m,e_n)} \leq 1 + \delta_{ik}, \forall (i,k) \in Mutex$$

$$\zeta_{ik}, \delta_{ik} \geq 0, y^i_{(e_m,e_n)} \in \{0,1\}, \forall i,k \quad (7)$$

Our algorithm that includes argument type checking and constraints is summarized in Algorithm 1.

2.4 Portuguese Verb Mapping

To map Portuguese verbs to relations in Portuguese NELL, which is an automatically and independently constructed KB separate from English NELL, we use the Portuguese NELL and Portuguese text corpus SVO_{pt}[7] and construct a dataset D_{pt}. Given

[6]The $Subset$ and $Mutex$ constraints are obtained as part of the NELL KB ontology, which is publicly available at the NELL Read The Web project website: http://rtw.ml.cmu.edu/resources/.

[7]We obtain the Portuguese SVO from the NELL-Portuguese team at Federal University of Sao Carlos.

Algorithm 1 The EM Algorithm for Verb-to-Relation Mapping

Input: $D = D^\ell \cup D^u$ and an initial naive Bayes classifier θ^1 from labeled documents D^ℓ only (using Equations 5 and 6)
Output: θ^T that include verbs to relations mappings given by $P(v_p|r_i;\theta^T)$

for t = 1 ... T **do**
  E-Step:
  for $d_{(e_m,e_n)} \in D^u$ **do**
    Compute $P(y^i_{(e_m,e_n)} = 1|d_{(e_m,e_n)};\theta^t)\ \forall r_i \in R$ that satisfy argument types checking (Equation 1)
    Find a consistent label assignment $\mathbf{y}^t_{(e_m,e_n)}$ by solving MIP (Equation 7)
  end for
  M-step: Recompute model parameters θ^{t+1} based on current label assignments (Equation 5 and 6) respecting argument type checking
  if convergence ($\mathcal{L}_c(\theta^{t+1})$, $\mathcal{L}_c(\theta^t)$) **then**
    break
  end if
end for
return θ^T

	English NELL	Portuguese NELL	Portuguese NELL^{+en}
$\lvert R\rvert$	317	302	302
$\lvert I_R\rvert$	135,267	5,675	12,444
$\lvert D^\ell\rvert$	85,192	2,595	5,412
$\lvert D^u\rvert$	240,490	595,274	1,186,329

Table 1: Statistics of KB facts and dataset constructed

D_{pt}, we follow the same approach as before to find a mapping of Portuguese verbs to relations. Since Portuguese NELL is newly constructed, it contains fewer facts (category and relation instances) than English NELL, and hence its dataset D^ℓ_{pt} has fewer labeled instances (see Table 1).

Adding more relation instances to Portuguese NELL can result in more labeled instances in the dataset D_{pt}, a more productive EM, and a better verb-to-relation mapping. Since each category and each relation in Portuguese NELL ontology has a one-to-one mapping in English NELL ontology, we can add relation instances to Portuguese NELL from the corresponding English NELL relations.

English NELL however, has only English noun phrases (NPs) to refer to entities in its relation instances. To add more labeled instances in D_{pt} using English relation instances, we need to find instantiations of these English relation instances in Portuguese SVO_{pt}, which translates to finding Portuguese NPs that refer to English NELL entities. For example, Portuguese NP: "Artria torcica interna" for English NELL entity: *internal mammary artery*.

To automatically translate English NELL enti-

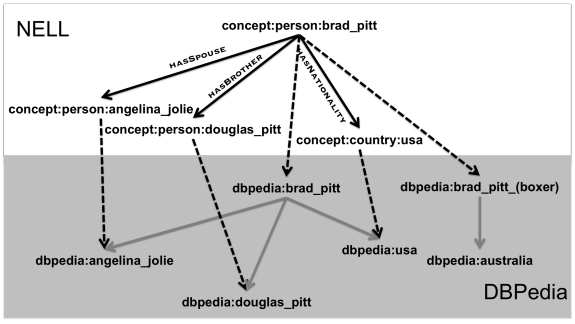

Figure 2: Mapping NELL entity *Brad Pitt* to DBPedia.

ties to Portuguese NPs, we use DBPedia (Auer et al., 2007) which has structured information about Wikipedia pages in many languages. The idea is to map each English NELL entity e_m to its corresponding English DBPedia page and therefore its Portuguese DBPedia page[8]. We use the structured information of the Portuguese page in DBPedia: its title and label as the set of Portuguese NPs corresponding to the English entity, $N_{pt}(e_m)$.

More specifically, for each English NELL entity e_m with English NPs that can refer to it, $N_{en}(e_m)$, we find *candidate* English DBPedia pages that can refer to the entity. We do this by computing Jaccard similarities (Jaccard, 1912; Chapman, 2009) of the entity's NPs with titles and labels of English DBPedia pages. We select pages with Jaccard similarities of more than 0.6 as candidates e.g., for English NELL entity *Brad Pitt* we find candidate English pages: `http://dbpedia.org/page/Brad_Pitt` (*Brad Pitt*, the US actor) and `http://dbpedia.org/page/Brad_Pitt_(boxer)` (*Brad Pit*, the Australian boxer).

Then, we construct a graph containing nodes that are: (1) the NELL entity that we want to map to DBPedia, (2) its candidate DBPedia pages, (3) other entities that have relations to the entity in NELL KB, and (4) the candidate DBPedia pages of these other entities (see Fig. 2 for the NELL entity *Brad Pitt*).

We add as edges to this graph: (1) the can-refer-to edges between entities in NELL and their candidate pages in DBPedia (dashed edges in Fig. 2), (2) the relation edges between the entities in NELL KB (black edges), and (3) the hyperlink edges between the pages in DBPedia (gray edges). In this graph we want to use the knowledge that NELL has already learned about the entity to narrow its candidates down to the page that the entity refers to. The idea is that relatedness among the entities in NELL implies relatedness among the DBPedia pages that refer to the entities. We use Personalized Page Rank (Page et al., 1999) to rank candidate DBPedia pages in this graph and pick the top ranked page as the page that can refer to the NELL entity.

For example, to find the DBPedia page that can refer to our NELL entity *Brad Pitt*, we use NELL's knowledge about this entity to rank its candidate pages. As seen in Fig. 2, DBPedia page of *Brad Pitt*, the US actor (*dbpedia:brad_pitt*) is highly connected to other pages (*dbpedia:angelina_jolie*, *dbpedia:douglas_pitt*, *dbpedia:usa*) that are in turn connected to the NELL entity *Brad Pitt*. *dbpedia:brad_pitt* is thus ranked highest and picked as the page that can refer to the NELL entity *Brad Pitt*.

Once we have an English DBPedia page that can refer to the NELL entity e_m, we can obtain the corresponding Portuguese page from DBPedia. The title and label of the Portuguese page becomes the set of Portuguese NPs that can refer to the NELL entity i.e., $N_{pt}(e_m)$ (see Table 2 for examples). Using $N_{pt}(e_m)$ we find instantiations of English relation instances in SVO_{pt} to add as labeled instances in D_{pt}. Portuguese NELL enriched with English NELL (i.e., Portuguese NELL^{+en}) has more than double the amount of relation instances, labeled and unlabeled instances (Table 1) than Portuguese NELL. In the experiments, we observe that this translates to a better verb-to-relation mapping.

Mapping NELL to DBPedia is also useful because it can align existing knowledge and add new knowledge to NELL. For example, by mapping to DBPedia, we can resolve abbreviations (e.g., the NELL entity: *COO* as "Chief Operations Officer" in English or "Diretor de Operaç*õ*es" in Portuguese), or resolve a person entity (e.g., the NELL entity: *Utamaro* as "Kitagawa Utamaro", the virtual artist).

3 Experiments

3.1 Pre-processing

For better coverage of verbs, we lemmatize verbs in the English SVO (using Stanford CoreNLP (Man-

[8] Almost every DBPedia English page has a corresponding Portuguese page

English NELL entity	Portuguese NPs
Amazonian Brown Brocket	"Veado-Roxo", "Fuboca"
COO	"Diretor de Operações"
Utamaro	"Kitagawa Utamaro"
Notopteridae	"Peixe-faca"
1967 Arab Israeli War	"Guerra dos Seis Dias", "Guerra de 1967"
Food Products	"Produtos Alimenticios", "Alimento", "Comida", ...

Table 2: Example Portuguese NPs learned for NELL entities

ning et al., 2014)). We lemmatize verbs in Portuguese SVO_{pt} (using LemPORT (Rodrigues et al., 2014)) and expand contracted prepositions.

For better precision and to make our method scale to a large text corpus, we focus on mapping verbs that are important for a relation based on how often the verbs co-occur with entity pairs that match the relation's argument type. For each argument type in the English SVO we consider only the top 50 verbs (in terms of *tf-idf* scores) for mapping. We use *tf-idf* scores to adjust for the fact that some verbs appear more frequently in general. For each of these verbs, we also use only the top 50 entity pairs that co-occur with the verb in the SVO (in terms of co-occurrence counts) to construct our dataset D.

For Portuguese verb-to-relation mapping, since SVO_{pt} is much smaller than the English SVO (i.e., it contains only about 22 million entity pair-verb triples compared to the 600 million triples in the English SVO), we use all the Portuguese entity pairs and verbs for mapping. To adjust for the fact that some verbs appear more frequently in general, we use *tf-idf* scores instead of co-occurrence counts for the values of $\mathbf{v}_{(e_m,e_n)}$ in the M-step (Equation 6).

3.2 Evaluation

We set aside 10% of D^{ℓ} for testing. Given a test instance $t_{(e_m,e_n)}$ and the trained model, we can predict the label assignment $\mathbf{y}_{(e_m,e_n)}$ using Eq. 1. This simulates the task of relation extraction where we predict relation(s) that exist between the entity pair in $t_{(e_m,e_n)}$.

We compare predicted labels of these test instances to the actual labels and measure precision, recall and F1 values of the prediction. We evaluate NELL relations that have more than one labeled instances in D^{ℓ} (constructed using the method described in section 2.2). For experiments on the English NELL, we evaluate 77 relations, with an average of 23 (and a median of 11) *training* instances per relations. For experiments on the Portuguese NELL^{+en}, which is Portuguese NELL enriched with relation instances from English NELL, we evaluate 85 relations, with an average of 31 (and a median of 10) *training* instances per relations. We compare the prediction produced by our approach: **EM** with that of other systems: **CPL**, **DIRT**, and **NB**.

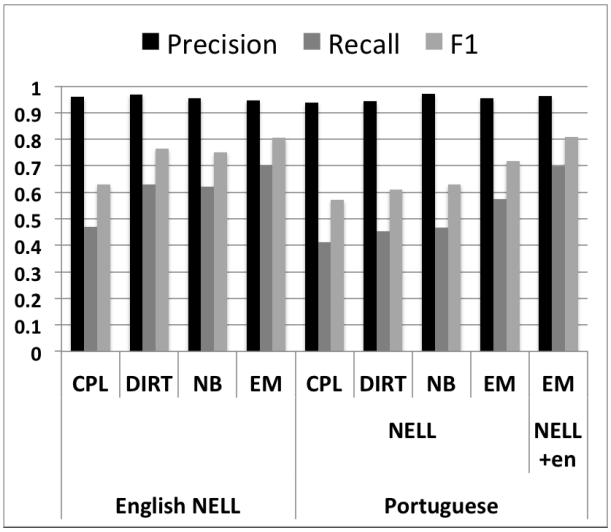

Figure 3: Performance on leaf relations.

In **CPL**, we obtain verb-to-relation mapping weights from NELL's CPL patterns and hand-labeled verb phrases (see Section 2.3.2). In **DIRT**, we obtain verb-to-relation mapping weights in an unsupervised manner (Lin and Pantel, 2001) based on their mutual information over labeled training instances. In Naive Bayes (**NB**) we learn the verb-to-relation mapping weights from labeled training instances. In contrast to the other systems, **EM** allows learning from both labeled and unlabeled instances.

To make other systems comparable to our proposed method, For **NB** and **DIRT** we add **CPL** weights as priors to their verb-to-relation mapping weights. For all these other systems, we also incorporate type-checking during prediction in that unlabeled instances are only labeled with relations that have the same argument types as the instance.

We show the micro-averaged performance of the systems on *leaf* relations of English NELL and Portuguese NELL (Fig. 3), where we do not incorporate constraints and classify each test instance into a single relation. We observe in both English and

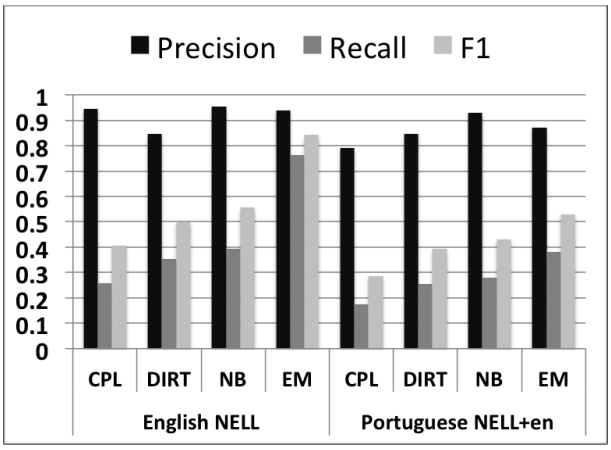

Figure 4: Performance on all relations.

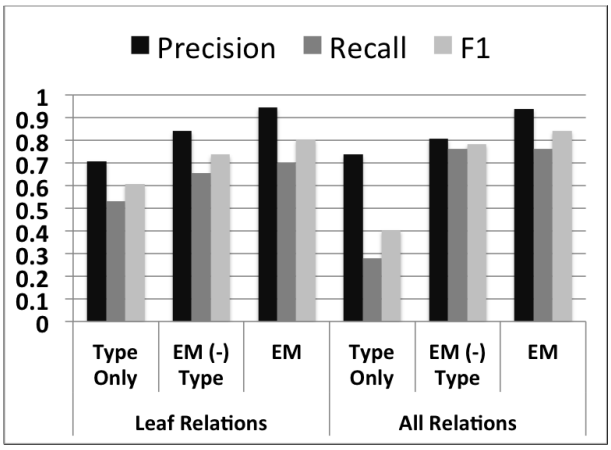

Figure 5: Performance on English NELL relations with and without type-checking.

Portuguese NELL that the verb-to-relation mapping obtained by **EM** results in predictions that have a much higher recall and comparable precision.

In Figure 3, we also observe a gain in performance when we run **EM** on Portuguese NELL^{+en} which is Portuguese NELL enriched with relation instances from English NELL obtained using our DBPedia linking in section 2.4. More labeled instances results in higher recall and precision. This shows the usefulness of aligning and merging knowledge from many different KBs to improve verb-to-relation mapping and relation extraction in general.

We show the micro-averaged performance of the systems on *all* relations of English NELL and Portuguese NELL (Fig. 4). Here, we incorporate hierarchical and mutual exclusive constraints between relations in our **EM**, allowing a test instance to be classified into more than one relation while respecting these constraints. Like before, we observe that the verb-to-relation mapping obtained by **EM** results in predictions with a much higher recall and comparable precision to other systems which do not incorporate constraints between relations.

In the experiments we also observe that **NB** performs comparably or better than **DIRT**. We hypothesize that it is because **NB** obtains its verb-to-relation mapping in a supervised manner while **DIRT** obtains its mapping in an unsupervised manner.

We also conduct experiments to investigate how much influence type-checking has on prediction. We show performance over instances whose types alone are not enough to disambiguate their assignments

Relation	Verbs	Proposed New Instances
bookWriter	a_1 be written by a_2, a_2 write a_1	(*Dracula*, *Bram Stoker*), (*Divine Comedy*, *Dante*)
city-Also-KnownAs	a_1 be known as a_2, a_2 be known as a_1, a_2 be renamed a_1,	(*Amman*, *Philadelphia*), (*Chennai*, *Madras*), (*Southport*, *Smithville*)
liderDe-Organizacao	a_1 fundador a_2, a_1 ceo de/em a_2	(*Jimmy Wales*, *Wikipedia*), (*Chad Hurley*, *Youtube*)
pessoa-Acusada-DoCrime	a_1 ser condenar aa_2, a_1 ser acusar de a_2, a_1 ser prender por a_2	(*Pedrinho Matador*, *Homicidios*), (*Omid Tahvili*, *Trafico de Drogaso*)

Table 3: Some relations' verbs and proposed new instances

(i.e., when more than one relation shares their argument type signatures) to see the merits of verb-to-relation mapping on prediction (Fig. 5). We observe that verbs learned by **EM** results in a better prediction even when used without type-checking (**EM** (-) **Type**) than using type-checking alone (by picking majority class among relations that have the correct type) (**Type Only**). Adding type checking improves performance even further (**EM**). This shows how verbs learning is complementary to type-checking.

The results of our experiments also highlight the merit of learning from a large, though unlabeled corpus to improve the coverage of verb-to-relation mapping and hence the recall of predictions. We also observe the usefulness of incorporating constraints and for merging knowledge from multiple KBs to improve performance. Another advantage of **EM** is that it produces relation labels for unlabeled data not yet in NELL KB. We show some of these new proposed relation instances as well as some of the verb-

to-relation mapping obtained by **EM** (Table 3).

EM learns in average 177 English verbs and 3310 Portuguese verbs per relation; and propose in average 1695 new instances per relation for English NELL, and 6426 new instances per relation for Portuguese NELL. It learns less English verbs than Portuguese due to the filtering of English data (Section 3.1) and a high degree of inflection in Portuguese verbs. The smaller size of Portuguese KB also means more of its proposed instances are new.

4 Related Work

Existing verb resources are limited in their ability to map to KBs. Some existing resources classify verbs into semantic classes either manually (e.g. WordNet (Miller et al., 1990)) or automatically (e.g. DIRT (Lin and Pantel, 2001)). However, these classes are not directly mapped to KB relations. Other resources provide relations between verbs and their arguments in terms of semantic roles (e.g. PropBank (Kingsbury and Palmer, 2002), VerbNet (Kipper et al., 2000), FrameNet (Ruppenhofer et al., 2006)). However, it is not directly clear how the verbs map to relations in specific KBs.

Most existing verb resources are also manually constructed and not scalable. A verb resource that maps to KBs should grow in coverage with the KBs, possibly by leveraging large corpora such as the Web for high coverage mapping. One system that leverages Web-text as an interlingua is (Wijaya et al., 2013). However, they use it to map KBs to KBs, and obtain a verb-to-relation mapping only indirectly. They also compute heuristic confidences in verb-to-relation mappings from label propagation scores, which are not probabilities. In contrast, we map verbs directly to relations, and obtain $P(v_p|r_i)$ as an integral part of our EM process.

In terms of systems that learn mappings of textual patterns to KB relations, CPL (Carlson et al., 2010) is one system that is most similar to our proposed approach in that it also learns text patterns for KB relations in a semi-supervised manner and uses constraints in KB ontology to couple the learning to produce extractors consistent with these constraints. However, CPL uses a combination of heuristics in its learning, while we use EM. In our experiments, we use CPL patterns that contain verbs as priors and show that our approach outperforms CPL in terms of effectiveness for extracting relation instances.

In terms of the relation extraction, there are distantly-supervised methods that can produce verb groupings as a by product of relation extraction. One state-of-the-art uses matrix factorization and universal schemas to extract relations (Riedel et al., 2013). In this work, they populate a database of a universal schema (which involves surface form predicates and relations from pre-existing KBs such as Freebase) by using matrix factorization models that learn latent feature vectors for relations and entity tuples. One can envision obtaining a verb grouping for a particular relation by predicting verb surface forms that occur between entity tuples that are instances of the relation. However, unlike our proposed method that learns mapping between typed-verbs to relations, they do not incorporate argument types in their learning, preferring to learn latent entity representation from data. Although this improves relation extraction, they observe that it hurts performance of surface form prediction because a single surface pattern (like "visit") can have multiple argument types (person-visit-location, person-visit-person, etc). Unlike our method, it is not clear in their method how argument types of surface patterns can be dealt with. Furthermore, it is not clear how useful prior constraints between relations (subset, mutex, etc.) can be incorporated in their method.

5 Conclusion

In this paper, we introduce an EM-based approach with argument type checking and ontological constraints to automatically map verb phrases to KB relations. We demonstrate that our verb resource is effective for extracting KB relation instances while improving recall; highlighting the value of learning from large scale unlabeled Web text. We also show the flexibility of our method. Being KB-, and language-independent, our method is able to construct a verb resource for any language, given a KB and a text corpus in that language. We illustrate this by building a verb resource in Portuguese and in English which are both effective for extracting KB relations. Future work will explore the use of our multilingual verb resource for relation extraction by reading natural language text in multiple languages.

Acknowledgments

We thank members of the NELL team at CMU and Federal University of Sao Carlos for their helpful datasets, comments, and suggestions. This research was supported by DARPA under contract number FA8750-13-2-0005.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

J. Callan, M. Hoy, C. Yoo, and L. Zhao. 2009. Clueweb09 data set. *boston.lti.cs.cmu.edu*.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3.

Sam Chapman. 2009. Simmetrics. *URL http://sourceforge. net/projects/simmetrics/. SimMetrics is a Similarity Metric Library, eg from edit distance's (Levenshtein, Gotoh, Jaro etc) to other metrics,(eg Soundex, Chapman). Work provided by UK Sheffield University funded by (AKT) an IRC sponsored by EPSRC, grant number GR N*, 15764.

Bhavana Dalvi, Einat Minkov, Partha P Talukdar, and William W Cohen. 2015. Automatic gloss finding for a knowledge base using ontological constraints. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 369–378. ACM.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*. Citeseer.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *AAAI/IAAI*, pages 691–696.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(04):343–360.

Yue Lu and Chengxiang Zhai. 2008. Opinion integration through semi-supervised topic modeling. In *Proceedings of the 17th international conference on World Wide Web*, pages 121–130. ACM.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244.

Kamal Nigam, Andrew McCallum, and Tom Mitchell. 2006. Semi-supervised text classification using em. *Semi-Supervised Learning*, pages 33–56.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: bringing order to the web.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas.

Ricardo Rodrigues, Hugo Gonçalo Oliveira, and Paulo Gomes. 2014. Lemport: a high-accuracy cross-platform lemmatizer for portuguese. *Maria João Varanda Pereira José Paulo Leal*, page 267.

Josef Ruppenhofer, Michael Ellsworth, Miriam RL Petruck, Christopher R Johnson, and Jan Scheffczyk. 2006. Framenet ii: Extended theory and practice.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

Derry Wijaya, Partha Pratim Talukdar, and Tom Mitchell. 2013. Pidgin: ontology alignment using web text as interlingua. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 589–598. ACM.