Joint Bootstrapping Machines for High Confidence Relation Extraction

Pankaj Gupta[1,2], Benjamin Roth[2], Hinrich Schütze[2]
[1]Corporate Technology, Machine-Intelligence (MIC-DE), Siemens AG Munich, Germany
[2]CIS, University of Munich (LMU) Munich, Germany
pankaj.gupta@siemens.com | pankaj.gupta@campus.lmu.de
{beroth, inquiries}@cis.lmu.de

Abstract

Semi-supervised bootstrapping techniques for relationship extraction from text iteratively expand a set of initial seed instances. Due to the lack of labeled data, a key challenge in bootstrapping is semantic drift: if a false positive instance is added during an iteration, then all following iterations are contaminated. We introduce BREX, a new bootstrapping method that protects against such contamination by highly effective confidence assessment. This is achieved by using entity and template seeds jointly (as opposed to just one as in previous work), by expanding entities and templates in parallel and in a mutually constraining fashion in each iteration and by introducing higher-quality similarity measures for templates. Experimental results show that BREX achieves an F_1 that is 0.13 (0.87 vs. 0.74) better than the state of the art for four relationships.

1 Introduction

Traditional semi-supervised bootstrapping relation extractors (REs) such as BREDS (Batista et al., 2015), SnowBall (Agichtein and Gravano, 2000) and DIPRE (Brin, 1998) require an initial set of seed *entity pairs* for the target binary relation. They find occurrences of positive seed entity pairs in the corpus, which are converted into extraction patterns, i.e., *extractors*, where we define an extractor as a cluster of instances generated from the corpus. The initial seed entity pair set is expanded with the relationship entity pairs newly extracted by the extractors from the text iteratively. The augmented set is then used to extract new relationships until a stopping criterion is met.

Due to lack of sufficient labeled data, rule-based systems dominate commercial use (Chiticariu et al., 2013). Rules are typically defined by creating patterns around the entities (entity extraction) or entity pairs (relation extraction). Recently, supervised machine learning, especially deep learning techniques (Gupta et al., 2015; Nguyen and Grishman, 2015; Vu et al., 2016a,b; Gupta et al., 2016), have shown promising results in entity and relation extraction; however, they need sufficient hand-labeled data to train models, which can be costly and time consuming for web-scale extractions. Bootstrapping machine-learned rules can make extractions easier on large corpora. Thus, open information extraction systems (Carlson et al., 2010; Fader et al., 2011; Mausam et al., 2012; Mesquita et al., 2013; Angeli et al., 2015) have recently been popular for domain specific or independent pattern learning.

Hearst (1992) used hand written rules to generate more rules to extract hypernym-hyponym pairs, without distributional similarity. For entity extraction, Riloff (1996) used seed entities to generate extractors with heuristic rules and scored them by counting positive extractions. Prior work (Lin et al., 2003; Gupta et al., 2014) investigated different extractor scoring measures. Gupta and Manning (2014) improved scores by introducing expected number of negative entities.

Brin (1998) developed the bootstrapping relation extraction system DIPRE that generates extractors by clustering contexts based on string matching. SnowBall (Agichtein and Gravano, 2000) is inspired by DIPRE but computes a TF-IDF representation of each context. BREDS (Batista et al., 2015) uses word embeddings (Mikolov et al., 2013) to bootstrap relationships.

Related work investigated adapting extractor scoring measures in bootstrapping entity extraction with either entities or *templates* (Table 1) as seeds (Table 2). The state-of-the-art relation extractors bootstrap with only seed entity pairs and suffer due to a surplus of unknown extractions and the lack of labeled data, leading to low confidence extractors. This in turn leads to to low confidence in the system output. Prior RE sys-

Proceedings of NAACL-HLT 2018, pages 26–36
New Orleans, Louisiana, June 1 - 6, 2018.

BREE	Bootstrapping Relation Extractor with *Entity pair*
BRET	Bootstrapping Relation Extractor with *Template*
BREJ	Bootstrapping Relation Extractor in Joint learning
type	a named entity type, e.g., `person`
typed entity	a typed entity, e.g., <"Obama",`person`>
entity pair	a pair of two typed entities
template	a triple of vectors ($\vec{v}_{-1}$, $\vec{v}_0$, $\vec{v}_1$) and an entity pair
instance	entity pair and template (types must be the same)
γ	instance set extracted from corpus
i	a member of γ, i.e., an instance
$x(i)$	the entity pair of instance i
$\mathfrak{t}(i)$	the template of instance i
G_p	a set of positive seed entity pairs
G_n	a set of negative seed entity pairs
$\mathfrak{G}_p$	a set of positive seed templates
$\mathfrak{G}_n$	a set of negative seed templates
$\mathcal{G}$	$< G_p, G_n, \mathfrak{G}_p, \mathfrak{G}_n >$
k_{it}	number of iterations
λ_{cat}	cluster of instances (*extractor*)
cat	category of *extractor* λ
λ_{NNHC}	Non-Noisy-High-Confidence extractor (True Positive)
λ_{NNLC}	Non-Noisy-Low-Confidence extractor (True Negative)
λ_{NHC}	Noisy-High-Confidence extractor (False Positive)
λ_{NLC}	Noisy-Low-Confidence extractor (False Negative)

Table 1: Notation and definition of key terms

tems do not focus on improving the extractors' scores. In addition, SnowBall and BREDS used a weighting scheme to incorporate the importance of contexts around entities and compute a similarity score that introduces additional parameters and does not generalize well.

Contributions. **(1)** We propose a *Joint Bootstrapping Machine*[1] (JBM), an alternative to the entity-pair-centered bootstrapping for relation extraction that can take advantage of both entity-pair and template-centered methods to jointly learn extractors consisting of instances due to the occurrences of both entity pair and template seeds. It scales up the number of positive extractions for *non-noisy* extractors and boosts their confidence scores. We focus on improving the scores for *non-noisy-low-confidence* extractors, resulting in higher *recall*. The relation extractors bootstrapped with entity pair, template and joint seeds are named as *BREE*, *BRET* and *BREJ* (Table 1), respectively.

(2) Prior work on embedding-based context comparison has assumed that relations have *consistent syntactic expression* and has mainly addressed synonymy by using embeddings (e.g.,"acquired" – "bought"). In reality, there is *large variation in the syntax* of how relations are expressed, e.g., "MSFT to acquire NOK for $8B" vs. "MSFT earnings hurt by NOK acquisition". We introduce cross-context similarities that compare all parts of the context (e.g., "to acquire" and "acquisition") and show that these perform better (in terms of recall) than measures assuming consistent syntactic expression of relations.

(3) Experimental results demonstrate a 13% gain in $F1$ score on average for four relationships and suggest eliminating four parameters, compared to the state-of-the-art method.

The *motivation* and *benefits* of the proposed JBM for relation extraction is discussed in depth in section 2.3. The method is applicable for both entity and relation extraction tasks. However, in *context of relation extraction*, we call it *BREJ*.

2 Method

2.1 Notation and definitions

We first introduce the notation and terms (Table 1).

Given a relationship like "x acquires y", the task is to extract pairs of entities from a corpus for which the relationship is true. We assume that the arguments of the relationship are typed, e.g., x and y are organizations. We run a named entity tagger in preprocessing, so that the types of all candidate entities are given. The objects the bootstrapping algorithm generally handles are therefore *typed entities* (an entity associated with a type).

For a particular sentence in a corpus that states that the relationship (e.g., "acquires") holds between x and y, a *template* consists of three vectors that represent the context of x and y. $\vec{v}_{-1}$ represents the context before x, $\vec{v}_0$ the context between x and y and $\vec{v}_1$ the context after y. These vectors are simply sums of the embeddings of the corresponding words. A template is "typed", i.e., in addition to the three vectors it specifies the types of the two entities. An *instance* joins an entity pair and a template. The types of entity pair and template must be the same.

The first step of bootstrapping is to extract a set of instances from the input corpus. We refer to this set as γ. We will use i and j to refer to instances. $x(i)$ is the entity pair of instance i and $\mathfrak{t}(\mathrm{i})$ is the template of instance i.

A required input to our algorithm are sets of positive and negative seeds for either entity pairs (G_p and G_n) or templates ($\mathfrak{G}_p$ and $\mathfrak{G}_n$) or both. We define $\mathcal{G}$ to be a tuple of all four seed sets.

We run our bootstrapping algorithm for k_{it} iterations where k_{it} is a parameter.

[1] github.com/pgcool/Joint-Bootstrapping-Machines

A key notion is the similarity between two instances. We will experiment with different similarity measures. The baseline is (Batista et al., 2015)'s measure given in Figure 4, first line: the similarity of two instances is given as a weighted sum of the dot products of their before contexts ($\vec{v}_{-1}$), their between contexts ($\vec{v}_0$) and their after contexts ($\vec{v}_1$) where the weights w_p are parameters. We give this definition for instances, but it also applies to templates since only the context vectors of an instance are used, not the entities.

The similarity between an instance i and a cluster λ of instances is defined as the maximum similarity of i with any member of the cluster; see Figure 2, right, Eq. 5. Again, there is a straightforward extension to a cluster of templates: see Figure 2, right, Eq. 6.

The extractors Λ can be categorized as follows:

$$\Lambda_{NNHC} = \{\lambda \in \Lambda | \underbrace{\lambda \mapsto \Re}_{non-noisy} \wedge \text{cnf}(\lambda, \mathcal{G}) \geqslant \tau_{cnf}\} \quad (1)$$

$$\Lambda_{NNLC} = \{\lambda \in \Lambda | \lambda \mapsto \Re \wedge \text{cnf}(\lambda, \mathcal{G}) < \tau_{cnf}\} \quad (2)$$

$$\Lambda_{NHC} = \{\lambda \in \Lambda | \underbrace{\lambda \nmapsto \Re}_{noisy} \wedge \text{cnf}(\lambda, \mathcal{G}) \geqslant \tau_{cnf}\} \quad (3)$$

$$\Lambda_{NLC} = \{\lambda \in \Lambda | \lambda \nmapsto \Re \wedge \text{cnf}(\lambda, \mathcal{G}) < \tau_{cnf}\} \quad (4)$$

where $\Re$ is the relation to be bootstrapped. The λ_{cat} is a member of Λ_{cat}. For instance, a λ_{NNLC} is called as a *non-noisy-low-confidence* extractor if it represents the target relation (i.e., $\lambda \mapsto \Re$), however with the confidence below a certain threshold (τ_{cnf}). Extractors of types Λ_{NNHC} and Λ_{NLC} are desirable, those of types Λ_{NHC} and Λ_{NNLC} undesirable within bootstrapping.

2.2 The Bootstrapping Machines: BREX

To describe BREX (Figure 1) in its most general form, we use the term *item* to refer to an entity pair, a template or both.

The input to BREX (Figure 2, left, line 01) is a set γ of instances extracted from a corpus and $\mathcal{G}_{\text{seed}}$, a structure consisting of one set of positive and one set of negative seed items. $\mathcal{G}_{\text{yield}}$ (line 02) collects the items that BREX extracts in several iterations. In each of k_{it} iterations (line 03), BREX first initializes the cache $\mathcal{G}_{\text{cache}}$ (line 04); this cache collects the items that are extracted in this iteration. The design of the algorithm balances elements that ensure high recall with elements that ensure high precision.

High recall is achieved by starting with the seeds and making three "hops" that consecutively consider order-1, order-2 and order-3 neighbors

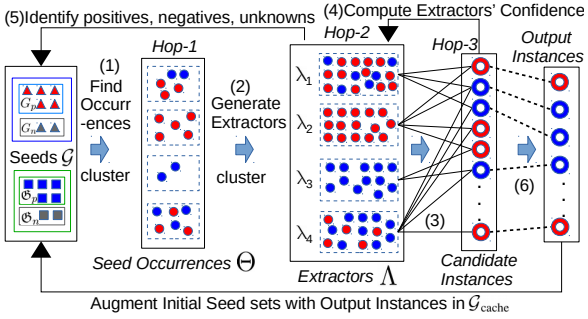

Figure 1: Joint Bootstrapping Machine. The red and blue filled circles/rings are the instances generated due to seed entity pairs and templates, respectively. Each dashed rectangular box represents a cluster of instances. Numbers indicate the flow. Follow the notations from Table 1 and Figure 2.

of the seeds. On line 05, we make the first hop: all instances that are similar to a seed are collected where "similarity" is defined differently for different BREX configurations (see below). The collected instances are then clustered, similar to work on bootstrapping by Agichtein and Gravano (2000) and Batista et al. (2015). On line 06, we make the second hop: all instances that are within τ_{sim} of a hop-1 instance are added; each such instance is only added to one cluster, the closest one; see definition of μ: Figure 2, Eq. 8. On line 07, we make the third hop: we include all instances that are within τ_{sim} of a hop-2 instance; see definition of ψ: Figure 2, Eq. 7. In summary, every instance that can be reached by three hops from a seed is being considered at this point. A cluster of hop-2 instances is named as *extractor*.

High precision is achieved by imposing, on line 08, a stringent check on each instance before its information is added to the cache. The core function of this check is given in Figure 2, Eq. 9. This definition is a soft version of the following hard max, which is easier to explain:

$$\text{cnf}(i, \Lambda, \mathcal{G}) \approx \max_{\{\lambda \in \Lambda | i \in \psi(\lambda)\}} \text{cnf}(i, \lambda, \mathcal{G})$$

We are looking for a cluster λ in Λ that licenses the extraction of i with high confidence. $\text{cnf}(i, \lambda, \mathcal{G})$ (Figure 2, Eq. 10), the *confidence* of a single cluster (i.e., extractor) λ for an instance, is defined as the product of the overall reliability of λ (which is independent of i) and the similarity of i to λ, the second factor in Eq. 10, i.e., $\text{sim}(i, \lambda)$. This factor $\text{sim}(i, \lambda)$ prevents an extraction by a cluster whose members are all distant from the instance – even if the cluster itself is highly reliable.

Algorithm: *BREX*

01 INPUT: γ, $\mathcal{G}_{\text{seed}}$
02 $\mathcal{G}_{\text{yield}} := \mathcal{G}_{\text{seed}}$
03 for k_{it} iterations:
04 $\quad \mathcal{G}_{\text{cache}} := \varnothing$
05 $\quad \Theta := \biguplus(\{i \in \gamma | \text{match}(i, \mathcal{G}_{\text{yield}})\}$
06 $\quad \Lambda := \{\mu(\theta, \Theta) | \theta \in \Theta\}$
07 $\quad$ for each $i \in \bigcup_{\lambda \in \Lambda} \psi(\lambda)$:
08 $\quad\quad$ if check$(i, \Lambda, \mathcal{G}_{\text{yield}})$:
09 $\quad\quad\quad$ add$(i, \mathcal{G}_{\text{cache}})$
10 $\quad \mathcal{G}_{\text{yield}} \cup= \mathcal{G}_{\text{cache}}$
11 OUTPUT: $\mathcal{G}_{\text{yield}}$, Λ

$$\text{sim}(i, \lambda) = \max_{i' \in \lambda} \text{sim}(i, i') \quad (5)$$

$$\text{sim}(i, \mathfrak{G}) = \max_{\mathfrak{t} \in \mathfrak{G}} \text{sim}(i, \mathfrak{t}) \quad (6)$$

$$\psi(\lambda) = \{i \in \gamma | \text{sim}(i, \lambda) \geqslant \tau_{\text{sim}}\} \quad (7)$$

$$\mu(\theta, \Theta) = \{i \in \gamma | \text{sim}(i, \theta) = d \wedge d = \max_{\theta \in \Theta} \text{sim}(i, \theta) \geqslant \tau_{\text{sim}}\} \quad (8)$$

$$\text{cnf}(i, \Lambda, \mathcal{G}) = 1 - \prod_{\{\lambda \in \Lambda | i \in \psi(\lambda)\}} (1 - \text{cnf}(i, \lambda, \mathcal{G})) \quad (9)$$

$$\text{cnf}(i, \lambda, \mathcal{G}) = \text{cnf}(\lambda, \mathcal{G}) \text{sim}(i, \lambda) \quad (10)$$

$$\text{cnf}(\lambda, \mathcal{G}) = \frac{1}{1 + w_n \frac{N_+(\lambda, \mathcal{G}_n)}{N_+(\lambda, \mathcal{G}_p)} + w_u \frac{N_0(\lambda, \mathcal{G})}{N_+(\lambda, \mathcal{G}_p)}} \quad (11)$$

$$N_0(\lambda, \mathcal{G}) = |\{i \in \lambda | x(i) \notin (G_p \cup G_n)\}| \quad (12)$$

Figure 2: BREX algorithm (left) and definition of key concepts (right)

	BREE	**BRET**	**BREJ**
Seed Type	*Entity pairs*	*Templates*	*Joint (Entity pairs + Templates)*
(i) $N_+(\lambda, \mathcal{G}_l)$	$\lvert\{i \in \lambda \lvert x(i) \in G_l\}\rvert$	$\lvert\{i \in \lambda \lvert \text{sim}(i, \mathfrak{G}_l) \geqslant \tau_{\text{sim}}\}\rvert$	$\lvert\{i \in \lambda \lvert x(i) \in G_l\}\rvert + \lvert\{i \in \lambda \lvert \text{sim}(i, \mathfrak{G}_l) \geqslant \tau_{\text{sim}}\}\rvert$
(ii) (w_n, w_u)	$(1.0, 0.0)$	$(1.0, 0.0)$	$(1.0, 0.0)$
05 match$(i, \mathcal{G})$	$x(i) \in G_p$	$\text{sim}(i, \mathfrak{G}_p) \geqslant \tau_{\text{sim}}$	$x(i) \in G_p \vee \text{sim}(i, \mathfrak{G}_p) \geqslant \tau_{\text{sim}}$
08 check$(i, \Lambda, \mathcal{G})$	$\text{cnf}(i, \Lambda, \mathcal{G}) \geqslant \tau_{\text{cnf}}$	$\text{cnf}(i, \Lambda, \mathcal{G}) \geqslant \tau_{\text{cnf}}$	$\text{cnf}(i, \Lambda, \mathcal{G}) \geqslant \tau_{\text{cnf}} \wedge \text{sim}(i, \mathfrak{G}_p) \geqslant \tau_{\text{sim}}$
09 add$(i, \mathcal{G})$	$G_p \cup= \{x(i)\}$	$\mathfrak{G}_p \cup= \{\mathfrak{x}(i)\}$	$G_p \cup= \{x(i)\}$, $\mathfrak{G}_p \cup= \{\mathfrak{x}(i)\}$

Figure 3: BREX configurations

The first factor in Eq. 10, i.e., $\text{cnf}(\lambda, \mathcal{G})$, assesses the reliability of a cluster λ: we compute the ratio $\frac{N_+(\lambda, \mathcal{G}_n)}{N_+(\lambda, \mathcal{G}_p)}$, i.e., the ratio between the number of instances in λ that match a negative and positive gold seed, respectively; see Figure 3, line (i). If this ratio is close to zero, then likely false positive extractions are few compared to likely true positive extractions. For the simple version of the algorithm (for which we set $w_n = 1$, $w_u = 0$), this results in $\text{cnf}(\lambda, \mathcal{G})$ being close to 1 and the reliability measure it not discounted. On the other hand, if $\frac{N_+(\lambda, \mathcal{G}_n)}{N_+(\lambda, \mathcal{G}_p)}$ is larger, meaning that the relative number of likely false positive extractions is high, then $\text{cnf}(\lambda, \mathcal{G})$ shrinks towards 0, resulting in progressive discounting of $\text{cnf}(\lambda, \mathcal{G})$ and leading to *non-noisy-low-confidence* extractor, particularly for a reliable λ. Due to lack of labeled data, the scoring mechanism cannot distinguish between noisy and non-noisy extractors. Therefore, an extractor is judged by its ability to extract more positive and less negative extractions. Note that we carefully designed this precision component to give good assessments while at the same time making maximum use of the available seeds. The reliability statistics are computed on λ, i.e., on hop-2 instances (not on hop-3 instances). The ratio $\frac{N_+(\lambda, \mathcal{G}_n)}{N_+(\lambda, \mathcal{G}_p)}$ is computed on instances that directly match a gold seed – this is the most reliable information we have available.

After all instances have been checked (line 08) and (if they passed muster) added to the cache (line 09), the inner loop ends and the cache is merged into the yield (line 10). Then a new loop (lines 03–10) of hop-1, hop-2 and hop-3 extensions and cluster reliability tests starts.

Thus, the algorithm consists of k_{it} iterations. There is a tradeoff here between τ_{sim} and k_{it}. We will give two extreme examples, assuming that we want to extract a fixed number of m instances where m is given. We can achieve this goal either by setting k_{it}=1 and choosing a small τ_{sim}, which will result in very large hops. Or we can achieve this goal by setting τ_{sim} to a large value and running the algorithm for a larger number of k_{it}. The flexibility that the two hyperparameters k_{it} and τ_{sim} afford is important for good performance.

$$\text{sim}_{\text{match}}(i,j) = \sum_{p\in\{-1,0,1\}} w_p \vec{v}_p(i)\vec{v}_p(j) \quad ; \quad \text{sim}_{cc}^{asym}(i,j) = \max_{p\in\{-1,0,1\}} \vec{v}_p(i)\vec{v}_0(j) \tag{13}$$

$$\text{sim}_{cc}^{sym1}(i,j) = \max\left(\max_{p\in\{-1,0,1\}} \vec{v}_p(i)\vec{v}_0(j), \max_{p\in\{-1,0,1\}} \vec{v}_p(j)\vec{v}_0(i)\right) \tag{14}$$

$$\text{sim}_{cc}^{sym2}(i,j) = \max\left(\left(\vec{v}_{-1}(i) + \vec{v}_1(i)\right)\vec{v}_0(j), \left(\vec{v}_{-1}(j) + \vec{v}_1(j)\right)\vec{v}_0(i), \vec{v}_0(i)\vec{v}_0(j)\right) \tag{15}$$

Figure 4: Similarity measures. These definitions for instances equally apply to templates since the definitions only depend on the "template part" of an instance, i.e., its vectors. (value is 0 if types are different)

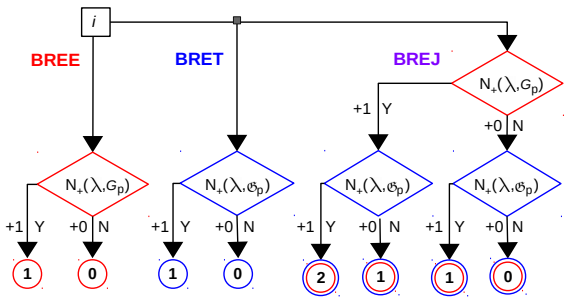

Figure 5: Illustration of Scaling-up Positive Instances. i: an instance in extractor, λ. Y: YES and N: NO

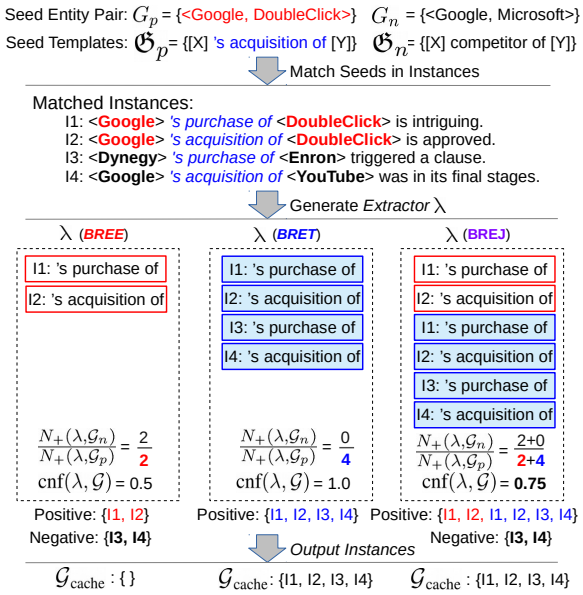

Figure 6: An illustration of scaling positive extractions and computing confidence for a non-noisy extractor generated for *acquired* relation. The dashed rectangular box represents an extractor λ, where λ (BREJ) is *hybrid* with 6 instances. Text segments matched with seed template are shown in italics. Unknowns (bold in black) are considered as negatives. $\mathcal{G}_{cache}$ is a set of output instances where $\tau_{cnf} = 0.70$.

2.3 BREE, BRET and BREJ

The main contribution of this paper is that we propose, as an alternative to entity-pair-centered BREE (Batista et al., 2015), template-centered BRET as well as BREJ (Figure 1), an instantiation of BREX that can take advantage of both entity pairs and templates. The differences and advantages of BREJ over BREE and BRET are:

(1) Disjunctive Matching of Instances: The first difference is realized in how the three algorithms match instances with seeds (line 05 in Figure 3). BREE checks whether the entity pair of an instance is one of the entity pair seeds, BRET checks whether the template of an instance is one of the template seeds and BREJ checks whether the disjunction of the two is true. The disjunction facilitates a higher hit rate in matching instances with seeds. The introduction of a few handcrafted templates along with seed entity pairs allows BREJ to leverage discriminative patterns and learn similar ones via distributional semantics. In Figure 1, the joint approach results in *hybrid* extractors Λ that contain instances due to seed occurrences Θ of both entity pairs and templates.

(2) Hybrid Augmentation of Seeds: On line 09 in Figure 3, we see that the bootstrapping step is defined in a straightforward fashion: the entity pair of an instance is added for BREE, the template for BRET and both for BREJ. Figure 1 demonstrates the hybrid augmentation of seeds via *red* and *blue* rings of *output instances*.

(3) Scaling Up Positives in Extractors: As discussed in section 2.2, a good measure of the quality of an extractor is crucial and N_+, the number of instances in an extractor λ that match a seed, is an important component of that. For BREE and BRET, the definition follows directly from the fact that these are entity-pair and template-centered instantiations of BREX, respectively. However, the disjunctive matching of instances for an extractor with entity pair and template seeds in BREJ (Figure 3 line "(i)") boosts the likelihood of finding positive instances. In Figure 5, we demonstrate computing the count of positive instances

Relationship	*Seed Entity Pairs*	*Seed Templates*
acquired	{Adidas;Reebok},{Google;DoubleClick}, {Widnes;Warrington},{Hewlett-Packard;Compaq}	{[X] acquire [Y]},{[X] acquisition [Y]},{[X] buy [Y]}, {[X] takeover [Y]},{[X] merger with [Y]}
founder-of	{CNN;Ted Turner},{Facebook;Mark Zuckerberg}, {Microsoft;Paul Allen},{Amazon;Jeff Bezos},	{[X] founded by [Y]},{[X] co-founder [Y]},{[X] started by [Y]}, {[X] founder of [Y]},{[X] owner of [Y]}
headquartered	{Nokia;Espoo},{Pfizer;New York}, {United Nations;New York},{NATO;Brussels},	{[X] based in [Y]},{[X] headquarters in [Y]},{[X] head office in [Y]}, {[X] main office building in [Y]},{[X] campus branch in [Y]}
affiliation	{Google;Marissa Mayer},{Xerox;Ursula Burns}, { Microsoft;Steve Ballmer},{Microsoft;Bill Gates},	{[X] CEO [Y]},{[X] resign from [Y]},{[X] founded by [Y]}, {[X] worked for [Y]},{[X] chairman director [Y]}

Table 2: Seed Entity Pairs and Templates for each relation. [X] and [Y] are slots for entity type tags.

$N_+(\lambda, \mathcal{G})$ for an extractor λ within the three systems. Observe that an instance i in λ can scale its $N_+(\lambda, \mathcal{G})$ by a factor of maximum 2 in BREJ if i is matched in both entity pair and template seeds. The reliability cnf$(\lambda, \mathcal{G})$ (Eq. 11) of an extractor λ is based on the ratio $\frac{N_+(\lambda,\mathcal{G}_n)}{N_+(\lambda,\mathcal{G}_p)}$, therefore suggesting that the scaling boosts its confidence.

In Figure 6, we demonstrate with an example how the joint bootstrapping scales up the positive instances for a *non-noisy* extractor λ, resulting in λ_{NNHC} for BREJ compared to λ_{NNLC} in BREE.

Due to unlabeled data, the instances not matching in seeds are considered either to be ignored/unknown N_0 or negatives in the confidence measure (Eq. 11). The former leads to high confidences for noisy extractors by assigning high scores, the latter to low confidences for non-noisy extractors by penalizing them. For a simple version of the algorithm in the illustration, we consider them as negatives and set $w_n = 1$. Figure 6 shows the three extractors (λ) generated and their confidence scores in BREE, BRET and BREJ. Observe that the scaling up of positives in BREJ due to BRET extractions (without w_n) discounts cnf$(\lambda, \mathcal{G})$ relatively lower than BREE. The discounting results in λ_{NNHC} in BREJ and λ_{NNLC} in BREE. The discounting in BREJ is adapted for *non-noisy* extractors facilitated by BRET in generating mostly non-noisy extractors due to stringent checks (Figure 3, line "(i)" and 05). Intuitively, the intermixing of non-noisy extractors (i.e., *hybrid*) promotes the scaling and boosts recall.

2.4 Similarity Measures

The before ($\vec{v}_{-1}$) and after ($\vec{v}_1$) contexts around the entities are highly sparse due to large variation in the syntax of how relations are expressed. SnowBall, DIPRE and BREE assumed that the between ($\vec{v}_0$) context mostly defines the syntactic expression for a relation and used weighted mechanism on the three contextual similarities in pairs, sim$_{match}$ (Figure 4). They assigned higher weights to the similarity in between ($p = 0$) contexts, that resulted in lower recall. We introduce attentive (max) similarity across all contexts (for example, $\vec{v}_{-1}(i)\vec{v}_0(j)$) to automatically capture the large variation in the syntax of how relations are expressed, without using any weights. We investigate asymmetric (Eq 13) and symmetric (Eq 14 and 15) similarity measures, and name them as *cross-context attentive* (sim$_{cc}$) similarity.

	ORG-ORG	*ORG-PER*	*ORG-LOC*
count	58,500	75,600	95,900

Table 3: Count of entity-type pairs in corpus

Parameter	*Description/ Search*	*Optimal*
$\|v_{-1}\|$	maximum number of tokens in before context	2
$\|v_0\|$	maximum number of tokens in between context	6
$\|v_1\|$	maximum number of tokens in after context	2
τ_{sim}	similarity threshold [0.6, 0.7, 0.8]	0.7
τ_{cnf}	instance confidence thresholds [0.6, 0.7, 0.8]	0.7
w_n	weights to negative extractions [0.0, 0.5, 1.0, 2.0]	0.5
w_u	weights to unknown extractions [0.0001, 0.00001]	0.0001
k_{it}	number of bootstrapping epochs	3
dim_{emb}	dimension of embedding vector, V	300
PMI	PMI threshold in evaluation	0.5
Entity Pairs	Ordered Pairs (OP) or Bisets (BS)	OP

Table 4: Hyperparameters in BREE, BRET and BREJ

3 Evaluation

3.1 Dataset and Experimental Setup

We re-run BREE (Batista et al., 2015) for **baseline** with a set of 5.5 million news articles from AFP and APW (Parker et al., 2011). We use processed dataset of 1.2 million sentences (released by BREE) containing at least two entities linked to FreebaseEasy (Bast et al., 2014). We extract four relationships: *acquired* (ORG-ORG), *founder-of* (ORG-PER), *headquartered* (ORG-LOC) and *affiliation* (ORG-PER) for Organization (ORG), Person (PER) and Location (LOC) entity types. We bootstrap relations in BREE, BRET and BREJ, each with 4 similarity measures using seed entity

	Relationships	$\#out$	P	R	$F1$	$\#out$	P	R	$F1$	$\#out$	P	R	$F1$	$\#out$	P	R	$F1$
		baseline: BREE+sim$_{match}$				**config$_2$**: BREE+sim$_{cc}^{asym}$				**config$_3$**: BREE+sim$_{cc}^{sym1}$				**config$_4$**: BREE+sim$_{cc}^{sym2}$			
BREE	acquired	2687	0.88	0.48	0.62	5771	0.88	0.66	0.76	3471	0.88	0.55	0.68	3279	0.88	0.53	0.66
	founder-of	628	0.98	0.70	0.82	9553	0.86	0.95	0.89	1532	0.94	0.84	0.89	1182	0.95	0.81	0.87
	headquartered	16786	0.62	0.80	0.69	21299	0.66	0.85	0.74	17301	0.70	0.83	0.76	9842	0.72	0.74	0.73
	affiliation	20948	0.99	0.73	0.84	27424	0.97	0.78	0.87	36797	0.95	0.82	0.88	28416	0.97	0.78	0.87
	avg	10262	0.86	0.68	0.74	16011	0.84	0.81	0.82	14475	0.87	0.76	0.80	10680	0.88	0.72	0.78
		config$_5$: BRET+sim$_{match}$				**config$_6$**: BRET+sim$_{cc}^{asym}$				**config$_7$**: BRET+sim$_{cc}^{sym1}$				**config$_8$**: BRET+sim$_{cc}^{sym2}$			
BRET	acquired	4206	0.99	0.62	0.76	15666	0.90	0.85	0.87	18273	0.87	0.86	0.87	14319	0.92	0.84	0.87
	founder-of	920	0.97	0.77	0.86	43554	0.81	0.98	0.89	41978	0.81	0.99	0.89	46453	0.81	0.99	0.89
	headquartered	3065	0.98	0.55	0.72	39267	0.68	0.92	0.78	36374	0.71	0.91	0.80	56815	0.69	0.94	0.80
	affiliation	20726	0.99	0.73	0.85	28822	0.99	0.79	0.88	44946	0.96	0.85	0.90	33938	0.97	0.81	0.89
	avg	7229	0.98	0.67	0.80	31827	0.85	0.89	0.86	35393	0.84	0.90	0.86	37881	0.85	0.90	0.86
		config$_9$: BREJ+sim$_{match}$				**config$_{10}$**: BREJ+sim$_{cc}^{asym}$				**config$_{11}$**: BREJ+sim$_{cc}^{sym1}$				**config$_{12}$**: BREJ+sim$_{cc}^{sym2}$			
BREJ	acquired	20186	0.82	**0.87**	**0.84**	35553	0.80	**0.92**	0.86	22975	0.86	0.89	0.87	22808	0.85	0.90	**0.88**
	founder-of	45005	0.81	**0.99**	**0.89**	57710	0.81	**1.00**	**0.90**	50237	0.81	0.99	0.89	45374	0.82	0.99	0.90
	headquartered	47010	0.64	**0.93**	**0.76**	66563	0.68	**0.96**	**0.80**	60495	0.68	0.94	0.79	57853	0.68	0.94	0.79
	affiliation	40959	0.96	**0.84**	**0.89**	57301	0.94	**0.88**	**0.91**	55811	0.94	0.87	0.91	51638	0.94	0.87	0.90
	avg	**38290**	0.81	**0.91**	**0.85**	**54282**	0.81	**0.94**	**0.87**	47380	0.82	0.92	0.87	44418	0.82	0.93	0.87

Table 5: Precision (P), Recall (R) and $F1$ compared to the state-of-the-art (*baseline*). $\#out$: count of output instances with cnf$(i, \Lambda, \mathcal{G}) \geqslant 0.5$. **avg**: average. **Bold** and underline: Maximum due to BREJ and sim$_{cc}$, respectively.

pairs and templates (Table 2). See Tables 3, 4 and 5 for the count of candidates, hyperparameters and different configurations, respectively.

Our evaluation is based on Bronzi et al. (2012)'s framework to estimate precision and recall of large-scale RE systems using FreebaseEasy (Bast et al., 2014). Also following Bronzi et al. (2012), we use Pointwise Mutual Information (PMI) (Turney, 2001) to evaluate our system automatically, in addition to relying on an external knowledge base. We consider only extracted relationship instances with confidence scores cnf$(i, \Lambda, \mathcal{G})$ equal or above 0.5. We follow the same approach as BREE (Batista et al., 2015) to detect the correct order of entities in a relational triple, where we try to identify the presence of passive voice using part-of-speech (POS) tags and considering any form of the verb to be, followed by a verb in the past tense or past participle, and ending in the word 'by'. We use GloVe (Pennington et al., 2014) embeddings.

3.2 Results and Comparison with baseline

Table 5 shows the experimental results in the three systems for the different relationships with *ordered* entity pairs and similarity measures (sim$_{match}$, sim$_{cc}$). Observe that BRET (config$_5$) is *precision-oriented* while BREJ (config$_9$) *recall-oriented* when compared to BREE (baseline). We see the number of output instances $\#out$ are also higher in BREJ, therefore the higher recall. The BREJ system in the different similarity configurations outperforms the baseline BREE and BRET in terms of $F1$ score. On an average for the four relations, BREJ in configurations config$_9$ and config$_{10}$ results in $F1$ that is 0.11 (0.85 vs 0.74) and 0.13 (0.87 vs 0.74) better than the baseline BREE.

We discover that sim$_{cc}$ improves $\#out$ and *recall* over sim$_{\text{match}}$ correspondingly in all three systems. Observe that sim$_{cc}$ performs better with BRET than BREE due to *non-noisy* extractors in BRET. The results suggest an alternative to the weighting scheme in sim$_{\text{match}}$ and therefore, the state-of-the-art (sim$_{cc}$) performance with the 3 parameters (w_{-1}, w_0 and w_1) ignored in bootstrap-

τ	k_{it}	$\#out$	P	R	$F1$
0.6	1	691	0.99	0.21	0.35
	2	11288	0.85	**0.79**	0.81
0.7	1	610	1.0	0.19	0.32
	2	7948	**0.93**	0.75	**0.83**
0.8	1	522	1.0	0.17	0.29
	2	2969	0.90	0.51	0.65

Table 6: Iterations (k_{it}) Vs Scores with thresholds (τ) for relation *acquired* in BREJ. τ refers to τ_{sim} and τ_{cnf}

	τ	$\#out$	P	R	$F1$	τ	$\#out$	P	R	$F1$
BREE	.60	1785	.91	.39	.55	.70	1222	.94	.31	.47
	.80	868	.95	.25	.39	.90	626	.96	.19	.32
BRET	.60	2995	.89	.51	.65	.70	1859	.90	.40	.55
	.80	1312	.91	.32	.47	.90	752	.94	.22	.35
BREJ	.60	18271	.81	.85	.83	.70	14900	.84	.83	.83
	.80	8896	.88	.75	.81	.90	5158	.93	.65	.77

Table 7: Comparative analysis using different thresholds τ to evaluate the extracted instances for *acquired*

BREX	*acquired*			*founder-of*			*headquartered*			*affiliation*		
	E	**T**	**J**	**E**	**T**	**J**	**E**	**T**	**J**	**E**	**T**	**J**
$\#hit$	71	682	743	135	956	1042	715	3447	4023	603	14888	15052

Table 8: Disjunctive matching of Instances. $\#hit$: the count of instances matched to positive seeds in $k_{it} = 1$

Attributes		$\|\Lambda\|$	*AIE*	*AES*	*ANE*	*ANNE*	*ANNLC*	*AP*	*AN*	*ANP*
acquired	BREE	167	12.7	0.51	0.84	0.16	0.14	37.7	93.1	2.46
	BRET	17	305.2	1.00	0.11	0.89	0.00	671.8	0.12	0.00
	BREJ	555	**41.6**	**0.74**	**0.71**	**0.29**	**0.03**	**313.2**	**44.8**	**0.14**
founder-of	BREE	8	13.3	0.46	0.75	0.25	0.12	44.9	600.5	13.37
	BRET	5	179.0	1.00	0.00	1.00	0.00	372.2	0.0	0.00
	BREJ	492	**109.1**	**0.90**	0.94	0.06	**0.00**	**451.8**	**79.5**	**0.18**
headquartered	BREE	655	18.4	0.60	0.97	0.03	0.02	46.3	82.7	1.78
	BRET	7	365.7	1.00	0.00	1.00	0.00	848.6	0.0	0.00
	BREJ	1311	**45.5**	**0.80**	0.98	0.02	**0.00**	**324.1**	**77.5**	**0.24**
affiliation	BREE	198	99.7	0.55	0.25	0.75	0.34	240.5	152.2	0.63
	BRET	19	846.9	1.00	0.00	1.00	0.00	2137.0	0.0	0.00
	BREJ	470	**130.2**	**0.72**	**0.21**	**0.79**	**0.06**	**567.6**	**122.7**	**0.22**

Table 9: Analyzing the attributes of extractors Λ learned for each relationship. Attributes are: number of extractors ($|\Lambda|$), avg number of instances in Λ (AIE), avg Λ score (AES), avg number of noisy Λ (ANE), avg number of non-noisy Λ (ANNE), avg number of Λ_{NNLC} below confidence 0.5 (ANNLC), avg number of positives (AP) and negatives (AN), ratio of AN to AP (ANP). The **bold** indicates comparison of BREE and BREJ with sim_{match}. avg: average

ping. Observe that sim_{cc}^{asym} gives higher recall than the two symmetric similarity measures.

Table 6 shows the performance of BREJ in different iterations trained with different similarity τ_{sim} and confidence τ_{cnf} thresholds. Table 7 shows a comparative analysis of the three systems, where we consider and evaluate the extracted relationship instances at different confidence scores.

3.3 Disjunctive Seed Matching of Instances

As discussed in section 2.3, BREJ facilitates disjunctive matching of instances (line 05 Figure 3) with seed entity pairs and templates. Table 8 shows $\#hit$ in the three systems, where the higher values of $\#hit$ in BREJ conform to the desired property. Observe that some instances in BREJ are found to be matched in both the seed types.

3.4 Deep Dive into Attributes of Extractors

We analyze the extractors Λ generated in BREE, BRET and BREJ for the 4 relations to demonstrate the impact of joint bootstrapping. Table 9 shows the attributes of Λ. We manually annotate the extractors as *noisy* and *non-noisy*. We compute $ANNLC$ and the lower values in BREJ compared to BREE suggest fewer non-noisy extractors with lower confidence in BREJ due to the scaled confi-

	Relationships	$\#out$	P	R	$F1$
BREE	acquired	387	0.99	0.13	0.23
	founder-of	28	0.96	0.09	0.17
	headquartered	672	0.95	0.21	0.34
	affiliation	17516	0.99	0.68	0.80
	avg	4651	0.97	0.28	0.39
BRET	acquired	4031	1.00	0.61	0.76
	founder-of	920	0.97	0.77	0.86
	headquartered	3522	0.98	0.59	0.73
	affiliation	22062	0.99	0.74	0.85
	avg	7634	0.99	0.68	0.80
BREJ	acquired	12278	0.87	0.81	**0.84**
	founder-of	23727	0.80	0.99	**0.89**
	headquartered	38737	0.61	0.91	**0.73**
	affiliation	33203	0.98	0.81	**0.89**
	avg	26986	0.82	0.88	**0.84**

Table 10: BREX+$\text{sim}_{\text{match}}$:Scores when w_n ignored

dence scores. *ANNE* (higher), *ANNLC* (lower), *AP* (higher) and *AN* (lower) collectively indicate that BRET mostly generates *NNHC* extractors. *AP* and *AN* indicate an average of $N_+(\lambda, \mathcal{G}_l)$ (line " (i)" Figure 3) for positive and negative seeds, respectively for $\lambda \in \Lambda$ in the three systems. Observe the impact of scaling positive extractions (*AP*) in BREJ that shrink $\frac{N_+(\lambda,\mathcal{G}_n)}{N_+(\lambda,\mathcal{G}_p)}$ i.e., *ANP*. It facilitates λ_{NNLC} to boost its confidence, i.e., λ_{NNHC} in BREJ suggested by *AES* that results in higher $\#out$ and recall (Table 5, BREJ).

3.5 Weighting Negatives Vs Scaling Positives

As discussed, Table 5 shows the performance of BREE, BRET and BREJ with the parameter $w_n = 0.5$ in computing extractors' confidence $\text{cnf}(\lambda, \mathcal{G})$(Eq. 11). In other words, config_9 (Table 5) is combination of both weighted negative and scaled positive extractions. However, we also investigate ignoring $w_n(= 1.0)$ in order to demonstrate the capability of BREJ with only scaling positives and without weighting negatives. In Table 10, observe that BREJ outperformed both BREE and BRET for all the relationships due to higher $\#out$ and recall. In addition, BREJ scores are comparable to config_9 (Table 5) suggesting that the scaling in BREJ is capable enough to remove the parameter w_n. However, the combination of both weighting negatives and scaling positives results in the state-of-the-art performance.

3.6 Qualitative Inspection of Extractors

Table 11 lists some of the non-noisy extractors (simplified) learned in different configurations to illustrate boosting extractor confidence $\text{cnf}(\lambda, \mathcal{G})$. Since, an extractor λ is a cluster of instances, therefore to simplify, we show one in-

config$_1$: BREE + sim$_{\text{match}}$	cnf$(\lambda, \mathbf{G})$	config$_5$: BRET + sim$_{\text{match}}$	cnf$(\lambda, \mathbf{G})$	config$_9$: BREJ + sim$_{\text{match}}$	cnf$(\lambda, \mathbf{G})$	config$_{10}$: BREJ + sim$_{cc}^{\text{asym}}$	cnf$(\lambda, \mathbf{G})$
acquired							
[X] acquired [Y]	0.98	[X] acquired [Y]	1.00	[X] acquired [Y]	1.00	acquired by [X] , [Y] †	0.93
[X] takeover of [Y]	0.89	[X] takeover of [Y]	1.00	[X] takeover of [Y]	0.98	takeover of [X] would boost [Y] 's earnings †	0.90
[X] 's planned acquisition of [Y]	0.87	[X] 's planned acquisition of[Y]	1.00	[X] 's planned acquisition of [Y]	0.98	acquisition of [X] by [Y] †	0.95
[X] acquiring [Y]	0.75	[X] acquiring [Y]	1.00	[X] acquiring [Y]	0.95	[X] acquiring [Y]	0.95
[X] has owned part of [Y]	0.67	[X] has owned part of [Y]	1.00	[X] has owned part of [Y]	0.88	owned by [X] 's parent [Y]	0.90
[X] took control of [Y]	0.49	[X] 's ownership of [Y]	1.00	[X] took control of [Y]	0.91	[X] takes control of [Y]	1.00
[X] 's acquisition of [Y]	0.35	[X] 's acquisition of [Y]	1.00	[X] 's acquisition of [Y]	0.95	acquisition of [X] would reduce [Y] 's share †	0.90
[X] 's merger with [Y]	0.35	[X] 's merger with[Y]	1.00	[X] 's merger with [Y]	0.94	[X] - [Y] merger between †	0.84
[X] 's bid for [Y]	0.35	[X] 's bid for [Y]	1.00	[X] 's bid for [Y]	0.97	part of [X] which [Y] acquired †	0.83
founder-of							
[X] founder [Y]	0.68	[X] founder [Y]	1.00	[X] founder [Y]	0.99	founder of [X] , [Y] †	0.97
[X] CEO and founder [Y]	0.15	[X] CEO and founder [Y]	1.00	[X] CEO and founder [Y]	0.99	co-founder of [X] 's millennial center , [Y] †	0.94
[X] 's co-founder [Y]	0.09	[X] owner [Y]	1.00	[X] owner [Y]	1.00	owned by [X] cofounder [Y]	0.95
		[X] cofounder [Y]	1.00	[X] cofounder [Y]	1.00	Gates co-founded [X] with school friend [Y] †	0.99
		[X] started by [Y]	1.00	[X] started by [Y]	1.00	who co-founded [X] with [Y] †	0.95
		[X] was founded by [Y]	1.00	[X] was founded by [Y]	0.99	to co-found [X] with partner [Y] †	0.68
		[X] begun by [Y]	1.00	[X] begun by [Y]	1.00	[X] was started by [Y] , cofounder	0.98
		[X] has established [Y]	1.00	[X] has established [Y]	0.99	set up [X] with childhood friend [Y] †	0.96
		[X] chief executive and founder [Y]	1.00	[X] co-founder and billionaire [Y] *	0.99	[X] co-founder and billionaire [Y]	0.97
headquartered							
[X] headquarters in [Y]	0.95	[X] headquarters in [Y]	1.00	[X] headquarters in [Y]	0.98	[X] headquarters in [Y]	0.98
[X] relocated its headquarters from [Y]	0.94	[X] relocated its headquarters from [Y]	1.00	[X] relocated its headquarters from [Y]	0.98	based at [X] 's suburban [Y] headquarters †	0.98
[X] head office in [Y]	0.84	[X] head office in [Y]	1.00	[X] head office in [Y]	0.87	head of [X] 's operations in [Y] †	0.65
[X] based in [Y]	0.75	[X] based in [Y]	1.00	[X] based in [Y]	0.98	branch of [X] company based in [Y]	0.98
[X] headquarters building in [Y]	0.67	[X] headquarters building in [Y]	1.00	[X] headquarters building in [Y]	0.94	[X] main campus in [Y]	0.99
[X] headquarters in downtown [Y]	0.64	[X] headquarters in downtown [Y]	1.00	[X] headquarters in downtown [Y]	0.94	[X] headquarters in downtown [Y]	0.96
[X] branch offices in [Y]	0.54	[X] branch offices in [Y]	1.00	[X] branch offices in [Y]	0.98	[X] 's [Y] headquarters represented †	0.98
[X] 's corporate campus in [Y]	0.51	[X] 's corporate campus in [Y]	1.00	[X] 's corporate campus in [Y]	0.99	[X] main campus in [Y]	0.99
[X] 's corporate office in [Y]	0.51	[X] 's corporate office in [Y]	1.00	[X] 's corporate office in [Y]	0.89	[X] , [Y] 's corporate †	0.94
affiliation							
[X] chief executive [Y]	0.92	[X] chief executive [Y]	1.00	[X] chief executive [Y]	0.97	[X] chief executive [Y] resigned monday	0.94
[X] secretary [Y]	0.88	[X] secretary [Y]	1.00	[X] secretary [Y]	0.94	worked with [X] manager [Y]	0.85
[X] president [Y]	0.87	[X] president [Y]	1.00	[X] president [Y]	0.96	[X] voted to retain [Y] as CEO †	0.98
[X] leader [Y]	0.72	[X] leader [Y]	1.00	[X] leader [Y]	0.85	head of [X] , [Y] †	0.99
[X] party leader [Y]	0.67	[X] party leader [Y]	1.00	[X] party leader [Y]	0.87	working with [X] , [Y] suggested †	1.00
[X] has appointed [Y]	0.63	[X] executive editor [Y]	1.00	[X] has appointed [Y]	0.81	[X] president [Y] was fired	0.90
[X] player [Y]	0.38	[X] player [Y]	1.00	[X] player [Y]	0.89	[X] 's [Y] was fired †	0.43
[X] 's secretary-general [Y]	0.36	[X] 's secretary-general [Y]	1.00	[X] 's secretary-general [Y]	0.93	Chairman of [X] , [Y] †	0.88
[X] hired [Y]	0.21	[X] director [Y]	1.00	[X] hired [Y]	0.56	[X] hired [Y] as manager †	0.85

Table 11: Subset of the non-noisy extractors (simplified) with their confidence scores cnf$(\lambda, \mathcal{G})$ learned in different configurations for each relation. $*$ denotes that the extractor was never learned in config$_1$ and config$_5$. † indicates that the extractor was never learned in config$_1$, config$_5$ and config$_9$. [X] and [Y] indicate placeholders for entities.

stance (mostly populated) from every λ. Each cell in Table 11 represents either a simplified representation of λ or its confidence. We demonstrate how the confidence score of a non-noisy extractor in BREE (config$_1$) is increased in BREJ (config$_9$ and config$_{10}$). For instance, for the relation *acquired*, an extractor $\{$*[X] acquiring [Y]*$\}$ is generated by BREE, BRET and BREJ; however, its confidence is boosted from 0.75 in BREE (config$_1$) to 0.95 in BREJ (config$_9$). Observe that BRET generates high confidence extractors. We also show extractors (marked by †) learned by BREJ with sim$_{cc}$ (config$_{10}$) but not by config$_1$, config$_5$ and config$_9$.

3.7 Entity Pairs: Ordered Vs Bi-Set

In Table 5, we use ordered pairs of typed entities. Additionally, we also investigate using entity sets and observe improved recall due to higher $\#out$ in both BREE and BREJ, comparing correspondingly Table 12 and 5 (*baseline* and config$_9$).

4 Conclusion

We have proposed a Joint Bootstrapping Machine for relation extraction (BREJ) that takes advantage of both entity-pair-centered and template-centered approaches. We have demonstrated that the joint approach scales up positive instances that boosts the confidence of NNLC extractors and improves recall. The experiments showed that the cross-context similarity measures improved recall and suggest removing in total four parameters.

Relationships	**BREE** + sim$_{\text{match}}$				**BREJ** + sim$_{\text{match}}$			
	$\#out$	P	R	$F1$	$\#out$	P	R	$F1$
acquired	2786	.90	.50	.64	21733	.80	.87	.83
founder-of	543	1.0	.67	.80	31890	.80	.99	.89
headquartered	16832	.62	.81	.70	52286	.64	.94	.76
affiliation	21812	.99	.74	.85	42601	.96	.85	.90
avg	10493	.88	.68	.75	37127	.80	.91	.85

Table 12: BREX+sim$_{\text{match}}$:Scores with entity *bisets*

Acknowledgments

We thank our colleagues Bernt Andrassy, Mark Buckley, Stefan Langer, Ulli Waltinger and Usama Yaseen, and anonymous reviewers for their review comments. This research was supported by Bundeswirtschaftsministerium (bmwi.de), grant 01MD15010A (Smart Data Web) at Siemens AG-CT Machine Intelligence, Munich Germany.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 15th ACM conference on Digital libraries*. Association for Computing Machinery, Washington, DC USA, pages 85–94.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Beijing, China, volume 1, pages 344–354.

Hannah Bast, Florian Bäurle, Björn Buchhold, and Elmar Haußmann. 2014. Easy access to the freebase dataset. In *Proceedings of the 23rd International Conference on World Wide Web*. Association for Computing Machinery, Seoul, Republic of Korea, pages 95–98.

David S. Batista, Bruno Martins, and Mário J. Silva. 2015. Semi-supervised bootstrapping of relationship extractors with distributional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 499–504.

Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *International Workshop on The World Wide Web and Databases*. Springer, Valencia, Spain, pages 172–183.

Mirko Bronzi, Zhaochen Guo, Filipe Mesquita, Denilson Barbosa, and Paolo Merialdo. 2012. Automatic evaluation of relation extraction systems on large-scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*. Association for Computational Linguistics, Montrèal, Canada, pages 19–24.

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 *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*. Atlanta, Georgia USA, volume 5, page 3.

Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. 2013. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington USA, pages 827–832.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland UK, pages 1535–1545.

Pankaj Gupta, Thomas Runkler, Heike Adel, Bernt Andrassy, Hans-Georg Zimmermann, and Hinrich Schütze. 2015. Deep learning methods for the extraction of relations in natural language text. Technical report, Technical University of Munich, Germany.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan, pages 2537–2547.

Sonal Gupta, Diana L. MacLean, Jeffrey Heer, and Christopher D. Manning. 2014. Induced lexico-syntactic patterns improve information extraction from online medical forums. *Journal of the American Medical Informatics Association* 21(5):902–909.

Sonal Gupta and Christopher Manning. 2014. Improved pattern learning for bootstrapped entity extraction. In *Proceedings of the 18th Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics, Baltimore, Maryland USA, pages 98–108.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 15th International Conference on Computational Linguistics*. Nantes, France, pages 539–545.

Winston Lin, Roman Yangarber, and Ralph Grishman. 2003. Bootstrapped learning of semantic classes from positive and negative examples. In *Proceedings of ICML 2003 Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*. Washington, DC USA, page 21.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, 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*. Association for Computational Linguistics, Jeju Island, Korea, pages 523–534.

Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. 2013. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington USA, pages 447–457.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the Workshop at the International Conference on Learning Representations*. ICLR, Scottsdale, Arizona USA.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Association for Computational Linguistics, Denver, Colorado USA, pages 39–48.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword. *Linguistic Data Consortium* .

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543.

Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI)*. Portland, Oregon USA, pages 1044–1049.

Peter D. Turney. 2001. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *Proceedings of the 12th European Conference on Machine Learning*. Springer, Freiburg, Germany, pages 491–502.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016a. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Association for Computational Linguistics, San Diego, California USA, pages 534–539.

Ngoc Thang Vu, Pankaj Gupta, Heike Adel, and Hinrich Schütze. 2016b. Bi-directional recurrent neural network with ranking loss for spoken language understanding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Shanghai, China, pages 6060–6064.