Joint Distant and Direct Supervision for Relation Extraction

Truc-Vien T. Nguyen and **Alessandro Moschitti**
Department of Information Engineering and Computer Science
University of Trento
38123 Povo (TN), Italy
{nguyenthi,moschitti}@disi.unitn.it

Abstract

Supervised approaches to Relation Extraction (RE) are characterized by higher accuracy than unsupervised models. Unfortunately, their applicability is limited by the need of training data for each relation type. Automatic creation of such data using Distant Supervision (DS) provides a promising solution to the problem. In this paper, we study DS for designing end-to-end systems of sentence-level RE. In particular, we propose a joint model between Web data derived with DS and manually annotated data from ACE. The results show (i) an improvement on the previous state-of-the-art in ACE, which provides important evidence of the benefit of DS; and (ii) a rather good accuracy on extracting 52 types of relations from Web data, which suggests the applicability of DS for general RE.

1 Introduction

Automatic Relation Extraction (RE) as defined in ACE (Doddington et al., 2004) achieves the highest accuracy when supervised approaches are applied, e.g., (Zelenko et al., 2002). Unfortunately, they require labeled data and tend to be domain-dependent as different domains involve different relations. Distant supervision (DS), e.g., using Wikipedia (Banko et al., 2007; Mintz et al., 2009; Hoffmann et al., 2010), can be applied for automatically acquiring relation types and their training data.

The main idea behind DS is to exploit (i) relation repositories, e.g., the *Infobox*, x, of Wikipedia to define a set of relation types $RT(x)$ and (ii) the text of the page associated with x to produce the training sentences, which are supposed to express instances of $RT(x)$.

Previous work has applied DS to RE at *corpus level*, e.g., (Banko et al., 2007; Mintz et al., 2009): relation extractors are (i) learned using such not completely accurate data and (ii) applied to extract relation instances from the whole corpus. The multiple pieces of evidence for each relation instance are then exploited to recover from errors of the automatic extractors. Additionally, a recent approach, i.e., (Hoffmann et al., 2010), has shown that DS can be also applied at level of Wikipedia article: given a target *Infobox* template, all its attributes[1] can be extracted from a given document matching such template.

Sentence-level RE (SLRE) has been typically modeled with the traditional supervised approach, e.g., using the data manually annotated in ACE (Culotta and Sorensen, 2004; Kambhatla, 2004; Bunescu and Mooney, 2005; Zhang et al., 2005; Zhang et al., 2006; Bunescu and Mooney, 2007; Nguyen et al., 2009). The resulting extractors are very valuable as they find rare relation instances that might be expressed in only one document. For example, the relation *President(Barrack Obama, United States*) can be extracted from thousands of documents thus there is a large chance of acquiring it. In contrast, *President(Eneko Agirre, SIGLEX)* is probably expressed in very few documents (if not just one sentence), increasing the complexity for obtaining it.

In (Nguyen and Moschitti, 2011), we firstly used DS from Wikipedia for SLRE by exploiting state-of-the-art models based on Support Vector Machines (SVMs) and kernel methods (KM). The experiments showed that our approach is robust to Web documents and can achieve high accuracy, i.e., an F1 of 74.29% on 52 YAGO relations.

In this paper, to accurately assess the benefit of using DS for SLRE, we manually mapped relations from YAGO to ACE based on their descrip-

[1] This is a simpler tasks as one of the two entity is fixed.

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 732–740,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

tions. Then, we designed a joint RE model combining DS and ACE data and tested it on ACE gold standard. This way the results are validated with the data provided by the expert linguistic annotators of ACE. The improvement produced by DS in these tests provides a strong evidence of the benefits of our joint model.

Additionally, since our aim is to produce RE for real-world applications, we experimented with end-to-end systems, which use Named Entity Recognizers (NERs). For this purpose, we also exploited Freebase for creating DS data for our robust NER (Nguyen et al., 2010). The results show that our RE systems can be applied to any document/sentence achieving an appreciable F1 of 67%.

In the remainder of this paper, Section 2 presents the related work, Section 3 describes the datasets for distant and direct supervision and the mapping between ACE and YAGO relations, Section 4 illustrates our RE models, including the joint ACE-Wikipedia model, Section 5 reports on all experiments with our models and finally Section 6 summarizes the conclusions.

2 Related Work

The extraction of relational data from text has drawn popularity for its potential application in a broad range of tasks. It refers to the automated extraction of relational facts, or world knowledge from the Web (Yates, 2009). To identify semantic relations using machine learning, three learning settings have mainly been applied, namely supervised methods (Zelenko et al., 2002; Culotta and Sorensen, 2004; Kambhatla, 2004; Zhou et al., 2005), semi supervised methods (Brin, 1998; Agichtein and Gravano, 2000), and unsupervised methods (Hasegawa et al., 2004; Banko et al., 2007).

Early work on Relation Extraction has mostly employed kernel-based approaches (Zelenko et al., 2002; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Zhang et al., 2005). Structural kernels on parse trees were proposed in (Collins and Duffy, 2001) for parse reranking and (Culotta and Sorensen, 2004) extended them for RE using augmented dependency trees. Recent literature has shown that efficient and appropriate kernels can be used to solve the RE problem, exploiting constituency trees (Zhang et al., 2006) and their combination with dependency trees (Nguyen et al., 2009)

Traditional relation classifiers use only labeled data for training. However, these are expensive to obtain, as they require efforts of experienced human annotators. In contrast, unlabeled data is relatively easy to collect, but its use is still an open problem. (Bunescu and Mooney, 2007) proposed a way of using a handful training set for RE. However, such model was applied to very few relation types. Distant supervised learning (Mintz et al., 2009) addresses this problem by using large amount of data to build classifiers.

The DS algorithm creates training data by selecting sentences that probably contain the target relation type. For example, suppose that $r(e1, e2)$ expresses one relation between pair of entities $e1$ and $e2$, then all sentences containing both $e1$ and $e2$ could be useful training examples. (Riedel et al., 2010) improved the DS assumption by only requiring that at least one of the sentences containing $e1$ and $e2$ expresses $r(e1; e2)$. They achieved a substantial improvement in extraction performance.

The most similar model to our DS algorithm is the method in (Hoffmann et al., 2010), which extracts relations from Wikipedia pages by using supervision from the page's infobox. In contrast, our approach allows for acquiring training data for relations defined in different sources.

3 Resources for designing and evaluating Generalized Distant Supervision

The resources we used to implement DS are YAGO, a large knowledge base of entities and relations, and Freebase, a collection of Wikipedia news articles. Our procedure uses entities and facts from YAGO to provide relation instances. For each pair of entities that appears in some YAGO relations, we retrieve all the sentences of the Freebase documents that contain such entities.

Additionally, as DS data is noisy, for accurately evaluating our extractors, we (i) manually annotated a small dataset and (ii) mapped some YAGO relations to ACE. This way we can measure the impact of Wikipedia training data on the ACE data.

3.1 ACE (Automatic Content Extraction)

The ACE effort (Doddington et al., 2004) aims at developing technology for automatically carrying out inference in natural language text. The data includes the entities being mentioned, the relations among these entities that are directly expressed, and the events in which these entities participate. Moreover, data includes various source types (image, audio, text) and languages (English, Arabic). We use the ACE 2004 corpus with seven relation types: Physical (PHYS), Person/Social (PER-SOC), Employment/Membership/Subsidiary (EMP-ORG), Agent-Artifact (ART), PER/ORG Affiliation (Other-AFF), GPE Affiliation (GPE-AFF), and Discourse (DISC). These relationships are explicitly described in the ACE document guidelines.

RE, as defined in ACE, is the task of finding relevant semantic relations between pairs of entities in texts. For example, the following sentence from the ACE 2004 corpus:

> ***Tara Singh Hayer****, editor of* ***The Indo-Canadian Times.***

expresses the employee/organization relation (EMP-ORG) between the first entity, i.e., *Tara Singh Hayer* (of type *person*) and the second entity, i.e., *The Indo-Canadian Times* (of type *organization*).

3.2 YAGO

This is a huge semantic knowledge base derived from WordNet and Wikipedia. It comprises about more than 2 million entities (like *persons*, *organizations*, *cities*, etc.) and 20 million facts connecting such entities. These include the taxonomic Is-A hierarchy as well as semantic relations between entities. The facts of YAGO have been extracted from the category system and the *Infoboxes* of Wikipedia and have been combined with taxonomic relations from Wordnet.

We use the YAGO ontology and the knowledge base, version *2008-w40-2*, whose validation has shown an accuracy of 95% for 99 relations. However, some of them are (a) rather trivial, e.g. *familyNameOf* or *givenNameOf*; (b) describe numerical attributes that change over time, e.g. *hasBudget*, *hasGDP* or *hasPopulation*; (c) symmetric, e.g. *hasPredecessor* and *hasSuccessor*; and (d) used for data management and do not convey semantics, e.g. *describes* or *foundIn*. Therefore, we removed trivial relations, unstable relations, and those used for data management. We obtained 1,489,156 instances of 52 relation types to be used with our DS approach. Some examples are shown in Table 1.

Algorithm 3.1: ACQUIRE_LABELED_DATA()

```
$DS = \emptyset$
$YAGO(R) : Instances\ of\ Relation\ R$
for each $\langle Wikipedia\ article : W \rangle \in Freebase$
  do
    $\mathcal{S} \leftarrow set\ of\ sentences\ from\ W$
    for each $s \in \mathcal{S}$
      do
        $\mathcal{E} \leftarrow set\ of\ entities\ from\ s$
        for each $E_1 \in \mathcal{E}$ and $E_2 \in \mathcal{E}$ and
        $R \in YAGO$
          do
            if $R(E_1, E_2) \in YAGO(R)$
              then $DS \leftarrow DS \cup \{s, R^+\}$
              else $DS \leftarrow DS \cup \{s, R^-\}$
return $(\mathcal{DS})$
```

3.3 Freebase

To access to the Wikipedia documents, we used Freebase (version March 27, 2010), which is a dump of the full text of all Wikipedia articles. It has been sentence-tokenized by Metaweb Technologies. For our experiments, we used 100,000 articles of which only 28,074 contain at least one relation for a total of 68,429 of relation instances. These connect 744,060 entities, 97,828 dates and 203,981 numerical attributes. Statistics are shown in Table 2.

In Freebase articles, Wikipedia entities like *Person*, *Organization* or *Location* are marked whereas numbers or dates are not. This prevents to extract interesting relations between entities and dates, e.g. *John F. Kennedy was born on May 29, 1917* or between entities and numerical attributes, e.g. *The novel Gone with the wind has 1037 pages*. Thus, we designed 18 regular expressions to extract dates and other 25 rules to extract numerical attributes, which range from integer numbers to ordinal numbers, percentage, monetary, speed, height, weight, area, time, and ISBN.

3.4 Distant Supervision

DS for RE is based on the following assumption, if (i) a sentence is connected *in some way* to a database of relations and (ii) it contains the pair of entities participating in such relation then it is likely that such sentence expresses the relation. For our DS, we relax (i) by allowing for the use of an external DB of relations such as YAGO and any document of Freebase. The alignment between YAGO and Freebase is implemented by the Wikipedia page link: for example the

Relation name	Size	Example
actedIn	28,836	George Clooney, Batman & Robin
bornIn	36,189	Alan Turing, London
created	95,248	Apple Inc., Dylan
diedIn	13618	Leonhard Euler, Saint Petersburg
directed	23,723	Mel Gibson, Braveheart
hasChild	4,454	Nero Claudius Drusus, Claudius
hasSuccessor	55,535	Jimmy Carter, Ronald Reagan
isAffiliatedTo	13,038	George W. Bush, Republican Party
isCitizenOf	4,865	Paul Cézanne, France
livesIn	14,710	Isaac Newton, England
locatedIn	60,261	Philadelphia, Pennsylvania
produced	41,747	Francis Ford Coppola, Apocalypse Now

Table 1: Some of selected YAGO relation types and their number of instances.

link *http://en.wikipedia.org/wiki/James_Cameron* refers to the entity *James_Cameron*.

A simplified version of our approach is the following: for any YAGO relation instance, scan all the sentences of all Wikipedia articles to test point (ii). Unfortunately, this procedure is impossible in practice since there are millions of relation instances in YAGO and millions of Wikipedia articles in Freebase, i.e. an order of magnitude of 10^{14} iterations[2]. Thus we use a more efficient procedure formally described in Alg. 3.1: for each Wikipedia article in Freebase, we scan all of its NEs. Then, for each pair of entities seen in the sentence, we query YAGO to retrieve the relation instance connecting these entities.

It should be noted that, our approach solves most of the problems for DS pointed out in (Bunescu and Mooney, 2007). Indeed, such issues are due to the sampling method used to acquire DS sentences: NEs were used as query to a search engine, whose weighting schemes introduce a bias. As, we utilize whole documents randomly drawn from Freebase and extract from them all possible positive and negative relation instances, no artificial feature (e.g. word) distribution is generated.

	Docs	Entities		Relations
ACE	443	Entities	12,037	5,784
DS	28,074	Entities	744,060	68,429
		Dates	97,828	
		Numbers	203,981	

Table 2: Statistics on the ACE and the DS datasets.

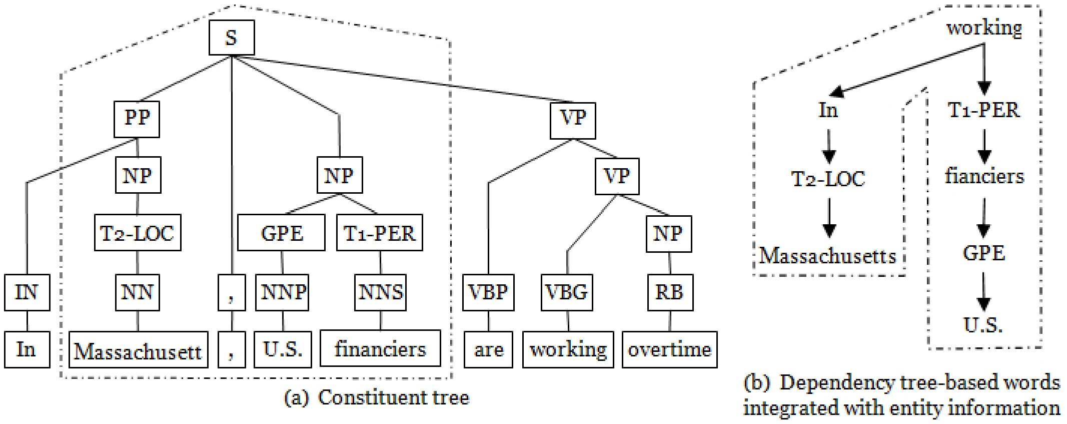
3.5 Mapping relations between YAGO-ACE

The YAGO knowledge base created from Wordnet and Wikipedia contains 99 relations whereas the ACE 2004 corpus only defines 7 relation types between 7 entity types. To further measure the impact of our Wikipedia dataset and the relations learnt, we mapped 33 relations of YAGO into those of ACE 2004. Surprisingly, we have found a fair correlation between the two different sources, which can help to validate our DS approach. The projection is shown in Table 3.

YAGO relations	Projection
actedIn	ART
bornIn	PHYS
created	ART
dealsWith	EMP-ORG
diedIn	PHYS
directed	ART
discovered	ART
graduatedFrom	EMP-ORG
happenedIn	PHYS
hasAcademicAdvisor	PER-SOC
hasCapital	PHYS
hasChild	PER-SOC
hasCurrency	ART
hasOfficialLanguage	ART
hasProduct	ART
hasProductionLanguage	ART
hasSuccessor	PER-SOC
hasWonPrize	ART
influences	PER-SOC
interestedIn	ART
isAffiliatedTo	EMP-ORG
isCitizenOf	GPE-AFF
isLeaderOf	EMP-ORG
isMarriedTo	PER-SOC
livesIn	PHYS
locatedIn	PHYS
madeCoverFor	ART
originatesFrom	PHYS
participatedIn	ART
politicianOf	Other-AFF
produced	ART
worksAt	EMP-ORG
wrote	ART

Table 3: 33 YAGO relation types projected into ACE.

[2] Assuming 100 sentences for each article.



(a) Constituent tree

(b) Dependency tree-based words integrated with entity information

Figure 1: The constituent and dependency parse trees integrated with entity information

4 Direct, distant and joint supervised learning

We model RE using state-of-the-art kernel methods: syntactic structures are used to represent relation instances whereas kernel functions measure the similarity between pairs of them. Such functions correspond to scalar products between implicit feature vectors in the space of substructures. Additionally, we define a joint model between the RE classifier trained on ACE and trained on DS data such that we can merge together the information from the two datasets on similar relation type.

4.1 RE based on Kernel Methods

State-of-the-art ACE RE, i.e. (Zhang et al., 2006; Nguyen et al., 2009), uses tree kernels applied to constituent and dependency syntactic structures, extracted from the sentences expressing the target relations. Given a parse tree, the path-enclosed tree (PET) is used as input of a tree kernel function. PET is the smallest common subtree including the two entities of a relation. Figure 1.a shows the constituent tree and figure 1.b shows a fragment of the dependency tree of the sentence: *In Massachussets, U.S. financiers are working overtime*. The dashed frame in Figure 1.a surrounds PET associated with the two mentions, *financiers* and *Massachussets*. Moreover, to improve the representation, two extra nodes T1-PER and T2-LOC, denoting the type PERSON and LOCATION, are added to the parse tree, above the two target NEs, respectively.

In our experiments, we use the model defined in (Zhang et al., 2006), which combines a syntactic tree kernel applied to constituent parse trees and a polynomial kernel over feature extracted from the entities:

$$CK_1 = \alpha \cdot K_P + (1 - \alpha) \cdot TK, \qquad (1)$$

where α is a coefficient to give more or less impact to the polynomial kernel, K_P, and TK is the syntactic tree kernel (Collins and Duffy, 2001) applied to PET.

We also use the best model in (Nguyen et al., 2009), which combines the advantages of the two parsing paradigms by adding six sequence kernels. These are applied to paths derived from the dependency tree and enriched with node labels of the constituent tree as follows:

$$CSK = \alpha \cdot K_P + (1 - \alpha) \cdot (TK + \sum_{i=1,..,6} SK_i), \qquad (2)$$

where SK_i are the sequence kernels applied to the structure i defined in (Nguyen et al., 2009).

In our application domain there are many different categories of name entities, e.g. Editor, President, Employer, and so on. Thus the typically available NE types, e.g. Person, Organization, Location, Time, Numbers, do not provide much selective information. For this purpose, we also provide adapted kernels by simply removing the category label in the nodes of the trees and in the sequences. This data transformation corresponds to define different kernel functions (Cristianini and Shawe-Taylor, 2000).

4.2 Joint Model for Distant and Direct Supervision

An interesting test of the quality of our DS data can be carried out by using it for ACE RE experiments. This way, we can use the gold and well annotated dataset of ACE to accurately measure the impact of DS data. For this purpose, we define a joint model as follows: first, we select the portion of hand-labeled ACE 2004 corpus containing common relations (see the mapping in Section 3.5).

Second, we create a huge labeled dataset under distant supervision assumption (described in Section 3.4) from Wikipedia news articles and YAGO knowledge base. Thanks to the projection from YAGO to ACE relations, we generate the two datasets under the same set of labels. This way, labeled data can be automatically acquired from a huge corpus and used to enrich ACE relation extractors.

Third, we train (i) the M_{ace} RE model on ACE dataset and (ii) the M_{mixed} model on ACE dataset mixed with the labeled data from Wikipedia (by using for example CSK).

Next, as standard SVM classifiers do not provide calibrated posterior probabilities we apply Platt transformation (Platt, 2000) improved by (Lin et al., 2007) with an additional sigmoid function. This allows us to map the SVM outputs of the two models M_{ace} and M_{mixed} into probabilities.

Finally, we linearly combine the probability of the two classifiers as follows:

$$P(C|r) = \alpha \cdot P(C|r, C_1) + \beta \cdot P(C|r, C_2), \quad (3)$$

where C_i is the output of classifier i, α and β are the weights learned from a validation set to encode the importance of the classifier for detecting the relation r. This combination provides a more robust model with respect to domain change.

5 Experiments

The aim of the experiments is to demonstrate that our DS produces reliable and practical usable relation extractors. For this purpose, we test SLRE trained with DS and with the joint DS and ACE data. We also test end-to-end RE, which also requires the experimentation of our automatic Named Entity Recognizer.

5.1 Experimental setting

We used the English portion of the ACE 2004 corpus including 443 documents, annotated with seven entity types and seven relation types. We obtained 5,784 positive and 55,650 negative examples when generating pairs of entity mentions as candidate relations. We employed the Stanford Parser (Klein and Manning, 2003) to produce parse trees. The candidate relations are generated by iterating all pairs of entity mentions in the same sentence.

Regarding the DS data extraction (see Table 2), we used two PCs, one with Intel X5270 3.50GHz CPU, 32GB RAM, another with 3.40GHz CPU and 8GB RAM to run the Algorithm 3.1. We processed about 25,000 Wikipedia documents per day per machine. When we added the generation of structures and features, the whole procedure required one day to process 5,000 Wikipedia documents (per machine). Thus, it took about 10 days to create the dataset and the computational learning files.

To train and test our binary relation classifier, we used SVMs, where relation detection is formulated as a multiclass classification problem. We employed *one vs. rest*, selecting the instance with largest margin as the final label. We used the Tree Kernel toolkit[3] (Moschitti, 2004; Moschitti, 2006; Moschitti, 2008) as SVM platform to implement CK_1 and CSK (see Section 4.1). The training phase with convolution kernels on syntactic parse tree and diverse sequence kernels on the large DS data took 3 days.

For testing on ACE data, we applied 5-fold cross-validation and evaluated single classifiers with the average of Precision, Recall and F1 on the 5-folds. The overall accuracy is measured with the mean of the Micro-Average (All) over the 5-folds.

For testing on Wikipedia, as DS data may be incorrect, we created a test set by sampling 200 articles from Freebase (these articles are not used for training). An expert annotator then examined one sentence at a time and took all possible pairs of entities, where the latter were already marked in the sentence. For each pair of entities, the considered 52 relations from YAGO (and used in our RE system) are marked as positive or negative, respectively. The annotator obtained 2,601 relation instances used for evaluation.

Regarding NE recognition, we applied CRFs to Wikipedia data but we could not use the whole amount of data. Thus we sampled 18,198 Wikipedia articles, selecting 4/5 for training and the rest for testing. The training phase took 14 hours and 30 minutes, whereas the classification took less than 10 minutes.

[3] http://disi.unitn.it/ moschitt/Tree-Kernel.htm

Class	PHYS		EMP-ORG		GPE-AFF		All	
Precision	72.06	72.46	85.71	90.00	78.95	83.33	72.41	80.16
Recall	67.12	68.49	80.00	81.82	75.00	75.00	75.54	72.66
F1	69.50	70.42	82.76	85.71	76.92	78.95	73.94	76.23

Table 4: RE from ACE 2004 of three relations between named entities: for each PHYS, EMP-ORG and GPE-AFF, the left and right columns report our best relation extractor only using ACE and ACE+Wikipedia data.

Class	PHYS	PER-SOC	EMP-ORG	ART	Other-AFF	GPE-AFF	DISC	All
ACE data								
Precision	56.28	88.12	80.82	80.68	62.73	76.55	80.15	74.47
Recall	44.51	59.80	76.73	39.20	17.11	32.32	59.85	57.26
F1	49.71	71.25	78.72	52.76	26.89	45.45	68.53	64.74
ACE + Wikipedia data								
Precision	58.22	91.06	81.76	80.68	62.73	78.49	80.15	77.65
Recall	48.44	64.74	76.66	37.14	17.11	32.26	59.85	59.84
F1	52.88	75.68	79.13	50.86	26.89	45.73	68.53	67.59

Table 5: Results on ACE 2004 considering all the type of entities and all the 7 ACE relations.

5.2 Using Wikipedia Relational Extractors to improve on ACE

In the ACE program, relations are defined between pairs of entities. These not only refer to NEs but also to mentions, e.g. indicated by a common noun or noun phrase, or represented by a pronoun. In contrast, Wikipedia instances mainly refer to NEs, e.g. *Leonardo Da Vinci*, *Canada* or *Titanic*, and we do not use pronominal references for building RE instances. Thus, we carried out two kinds of experiments: using (i) RE task as defined in ACE with all kind of entities and (ii) only relations between named entities. We have observed that the NE relations only exist for the classes: Physical (PHYS), Employment/Membership/Subsidiary (EMP-ORG) and GPE Affiliation (GPE-AFF).

Table 5 presents the combination results. Overall, using Wikipedia data improves the state-of-the-art of standard RE from 64.74% to 67.59%. Moreover, if we focus on *proper* NE relations, i.e. of the type indicated in point (ii), the relation extractors improve from 73.94% to 76.23%. These results are interesting as show that (a) we can improve the best systems with DS and (b) relations learned from Wikipedia can be mapped into those defined by expert linguists on ACE. We also tested a model learned from only DS data. For space reason, we do not report the complete results: as expected, its overall F1 is lower than the model trained on only ACE (about 10 absolute percent points less).

5.3 End-to-end Relation Extraction

In this section, we describe the experiments using automatic NEs. Previous work, e.g. (Zhang et al., 2006; Zhou et al., 2007; Nguyen et al., 2009) performed extraction using gold entity features such as entity types (*Person*, *Location*, *Organization*), entity subtypes (*Nation*, *Population-Center* for *GPE*). For example, in the sentence *Bush went to Washington*, the type of the first named entity, *Bush*, is PERSON and for the second named entity, *Washington*, is LOCATION. When accurate, such features improve performance. In case of fully automatic systems they introduce noise and in Wikipedia they are not available. Thus, we removed all gold entity features (entity type, entity subtype, mention type, and LDC mention type) from ACE annotations. We modeled tree and sequence kernels based on constituent and dependency parse trees along with a few features that can be extracted automatically such as the string and the head word of the entity. Note that in (Nguyen et al., 2009; Zhou et al., 2007; Zhang et al., 2006), even for tree kernels, the tree structures were also integrated with entity types (see Figure 1 as an example). Therefore, in the parse trees in Figure 1, we replaced entity types PER, ORG, LOC with a generic type ETYPE.

5.3.1 Entity Extraction from ACE and Wikipedia

For entity extraction, we followed the design in (Nguyen et al., 2010) by applying CRF++ [4]. We

[4] http://crfpp.sourceforge.net

Corpus	ACE	Wikipedia
Precision	77.84	68.84
Recall	70.26	64.56
F1	73.85	66.63

Table 6: Results of entity extraction from ACE and entity detection from Wikipedia.

performed automatic entity extraction from seven classes from ACE 2004 and entity detection from Wikipedia. While ACE documents have been annotated with seven classes *Person, Organization, Facility, Location, GPE, Vehicle, Weapon*, for Wikipedia, we used Freebase as learning source, where entities have been annotated in each Wikipedia article. Note that for Wikipedia, the entity detection has been done for only entities, like *Person, Organization, Location*. For dates and numerical attributes, we used the extraction patterns described in Section 3.4. The results reported in Table 6 are rather lower than in standard NE recognition. We should consider that our NER also tags mentions in ACE, which is a hard task whereas for Wikipedia, the entity instances from YAGO potentially belong to thousands of different categories. Although we do not categorize entities, it makes the complexity of detecting of NE boundaries higher.

5.3.2 RE from Automatic Entity Extraction

Web data entities are often not annotated and not available as in hand-labeled corpora like ACE or in Wikipedia pages. In this new experiment, we move to a novel task where entities are detected and classified automatically from a classifier. This way, we aim at designing an end-to-end RE system, where entities are not known beforehand. We also introduce a new task, that is the extraction of Wikipedia relations from any web text, i.e. detection of Wikipedia instances from any web page and not only from Wikipedia articles (where links often exist for Wikipedia instances).

Setting	Gold Features/ Gold NEs	No gold Features/ Gold NEs	No gold Features/ Auto NEs
Precision	76.60	74.47	70.27
Recall	67.00	57.26	47.52
F1	71.50	64.74	56.70

Table 7: Results on end-to-end RE from ACE.

The results are shown in Table 7 and Table 8. We note that the gold entity features lead to very good F1. When we remove these, the F1 decreases from 71.50% to 64.74%. Nevertheless, without gold entity features, RE from Wikipedia still achieves very good performance, i.e. an F1 of 74.29%.

Setting	Gold NEs	Automatic NEs
Precision	91.42	82.16
Recall	62.57	56.57
F1	74.29	67.00

Table 8: Results on end-to-end RE from Wikipedia.

6 Conclusion

In this paper, we proposed a study on novel training methods using semi-structured resources such as Wikipedia. As the NLP field always requires new methods to leverage the ever-increasing amounts of user-generated data available on the web, ours is a particularly important achievement for RE. We presented adaptation and experimentation of state-of-the-art RE models also exploiting a mapping between Wikipedia and ACE relations. We also extensively experimented with end-to-end systems applicable both to Wikipedia pages as well as to any natural language text. Our method is general and we suggest that it could be applicable to other external resources or other NLP tasks.

Acknowledgments

This research has been partly supported by the European Community Seventh Framework Programme (FP7/2007-2013) under the grant 247758: Trustworthy Eternal Systems via Evolving Software, Data and Knowledge (EternalS).

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI*, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Sergey Brin. 1998. Extracting patterns and relations from world wide web. In *Proceeding of WebDB Workshop at 6th International Conference on Extending Database Technology*, pages 172–183.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of HLT-EMNLP*, pages 724–731, Vancouver, British Columbia, Canada, October.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of ACL*, pages 576–583, Prague, Czech Republic, June.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of NIPS*, pages 625–632.

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, United Kingdom.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of ACL*, pages 423–429, Barcelona, Spain, July.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) programtasks, data, and evaluation. In *Proceedings of LREC*, pages 837–840, Barcelona, Spain.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of ACL*, pages 415–422, Barcelona, Spain, July.

Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. 2010. Learning 5000 relational extractors. In *Proceedings of ACL*, pages 286–295, Uppsala, Sweden, July.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of ACL*, pages 178–181, Barcelona, Spain.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*, pages 423–430, Sapporo, Japan, July.

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. 2007. A note on platts probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-AFNLP*, pages 1003–1011, Suntec, Singapore, August.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow statistic parsing. In *Proceedings of ACL*, pages 335–342, Barcelona, Spain, July.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of ECML*, pages 318–329, Berlin, Germany, September.

Alessandro Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *Proceedings of CIKM*, pages 253–262, New York, NY, USA. ACM.

Truc Vien T. Nguyen and Alessandro Moschitti. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of ACL-HLT*, pages 277–282, Portland, Oregon, USA, June.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of EMNLP*, pages 1378–1387, Singapore, August.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2010. Kernel-based reranking for named-entity extraction. In *Proceedings of COLING*, pages 901–909, Beijing, China, August.

John C. Platt. 2000. Probabilities for sv machines. *Advances in Large Margin Classifiers*, pages 61–74.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, pages 148–163.

Alexander Yates. 2009. Extracting world knowledge from the web. *IEEE Computer*, 42(6):94–97, June.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2002. Kernel methods for relation extraction. In *Proceedings of EMNLP*, pages 71–78, July.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proceedings of IJCNLP*, pages 378–389.

Min Zhang, Jie Zhang, Jian Su, and GuoDong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of COLING-ACL*, pages 825–832, Sydney, Australia, July.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL*, pages 427–434, Ann Arbor, USA, June.

GuoDong Zhou, Min Zhang, DongHong Ji, and QiaoMing Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of EMNLP-CoNLL*, pages 728–736, Prague, Czech Republic, June.