Distant Supervised Relation Extraction with Wikipedia and Freebase

Marcel Ackermann
TU Darmstadt
ackermann@tk.informatik.tu-darmstadt.de

Abstract

In this paper we discuss a new approach to extract relational data from unstructured text without the need of hand labeled data. So-called distant supervision has the advantage that it scales large amounts of web data and therefore fulfills the requirement of current information extraction tasks. As opposed to supervised machine learning we train generic, relation- and domain-independent extractors on the basis of data base entries. We use Freebase as a source of relational data and a Wikipedia corpus tagged with unsupervised word classes. In contrast to previous work in the field of distant supervision, we do not rely on preprocessing steps that involve supervised learning. This work consists of three parts, a distant supervised Named Entity Recognizer (NER), a distant supervised classifier to recognize sentences in which a certain relation between two objects is described and the combination of both, allowing us for example to contribute new instances to Freebase. The performance of the NER is too low, that the combined method produces usable results. Still the subcomponents can be used independently.

1 Introduction

Banko & Etzioni (2008) define Relation Extraction (RE) as "the task of recognizing the assertion of a particular relationship between two or more entities in text". For example, in the sentence: `Juan Ramón Jiménez was born in Moguer`, one relation to recognize is person/place_of_birth between Juan Ramón Jiménez and Moguer. Extracting relational facts from unstructured text is a highly relevant topic, as it has many applications, such as Information Retrieval, Information Extraction, Text Summarization, Question Answering, Paraphrasing and Word Sense Disambiguation.

The common practice is to use supervised machine learning methods, learning extractors for entities and their relations from hand-labeled corpora. For example in SemEval-2[1] task 8 (Multi-Way Classification of Semantic Relations Between Pairs of Nominals) there are nine relations, for each of them 1000 manually labeled example sentences. These relations can be learned using lexical, syntactic and semantic features. At this, any kind of resources are employed, such as large corpora, dictionaries or lexical-semantic resources like WordNet. Rink and Harabagiu (2010) achieved a macro-averaged F-score[2] of .82 in this task, using context words, hypernyms, parts-of-speech (POS), dependencies, semantic roles, paraphrases and more.

Good results in supervised relation extraction are only of limited value as it is not applicable in most contexts. Labeling training data is expensive and time consuming and therefore only available for a few relations on a small corpus. This does not scale to the amount of relations required for most NLP applications. Also it is highly domain-dependent and thus not applicable for heterogeneous texts like web corpora or narrow-domain company documents. (Mintz et al., 2009)

On the other side of the spectrum are unsupervised machine learning methods, for which Banko & Etzioni (2008) coined the term "Open IE". Their O-CRF system learns relation-independent lexico-syntactic patterns from a large web corpus. Zhu et al. (2009) learn patterns with Markov Logic Networks, achieving an F-score of .76 on the Sent500 data set. The main issue with the resulting relations is, that they are hard to map on existing knowledge bases. In addition to this most Open IE systems use subcom-

[1] ACL 2010, Proceedings of the 5th International Workshop on Semantic Evaluation.

[2] also F1, defined by the harmonic mean of precision (#correct/#found) and recall (#correct/#contained)

ponents, such as a tagger, parser or NER that are trained with supervised machine learning.

There is a broad range of methods between the two extremes. For example Yan et al. (2009) mine the article structure of Wikipedia. Another common approach is bootstrapping, where a small number of seed instances is used to extract new instances or patterns, which themselves function as new seeds in an iterative manner (Bunescu & Mooney, 2007), (Rozenfeld & Feldman, 2008). This often leads to semantic drift and low precision.

Mintz et al. (2009) present an alternative approach – distant supervision – combining advantages from the above methods. In short, distant supervision means that a training corpus is labeled with relational data from an external source. They build upon the work of Snow et al. (2005), mining WordNet relations and Morgan et al. (2004) using weakly labeled data in bioinformatics. Mintz et al. train a logistic regression classifier on a large amount of features, obtained from sentences containing instances from the Freebase database[3]. Their approach is based on the distant supervision assumption (Riedel et al., 2010): "If two entities participate in a relation, all sentences that mention these two entities express that relation."

Our approach has a similar setup as Mintz et al, we also use Wikipedia and Freebase as data sources and perform distant supervision – n-grams in Wikipedia sentences are labeled if they appear in Freebase, how exactly will be described in section 3. Our addition is that we do not depend on any supervised data at all, such as pre-trained POS tagger, NER or parsers. We use the unsupervised POS tagger of (Biemann, 2009), to tag a large amount of Wikipedia data. Then we filter for sentences in which instances of some Freebase relation occur. After that we have a hierarchical approach to obtain the entities of a relation and patterns that determine if a target relation exists in the seen sentence. This results in a system that is independent of language or domain, scales to Web size corpora and its output can directly mapped with canonical names to existing relations as defined in our database or ontology.

[3] www.freebase.com, an open triple store for real world entities and their relations

2 Related Work

After the overview of supervised, unsupervised and bootstrapping methods we now focus on work in the field of distant supervision.

Mintz et al. (2009) use the Stanford four-class named entity tagger (Finkel et al., 2005) which is supervised trained for the tags {person, location, organization, miscellaneous, none}. For sentences in which both entities from a Freebase relation occur and are tagged with a label other than none, they collect features. These features are lexical (words, word-window, and corresponding POS tags) and syntactic (dependency paths). Both POS tagger and parser are supervised, making their method domain dependent. Although they achieve a precision of .67 when returning 1000 results, practical usage is highly questionable, as they can only find relations consisting of the four Stanford categories.

Hoffmann et al. (2010) developed a "self-supervised, relation-specific IE system which learns 5025 relations". They apply dynamic lexicon learning to cope with noisy and sparse data. The lexicons are learned on lists obtained from the web. These lexicons are the input for a CRF tagger as well as Wikipedia data; it is left open how they tag their data. Other features used for the CRF are: Words, transitions between labels, capitalization, digits and dependency parses. For the last one they use the Stanford parser, which is again supervised, with all problems mentioned above. Also they use a labeled training set without further explaining it, so it is questionable if their F-score of 61% can be reached in real world scenarios.

Yao et al. (2010) present a similar setup as Mintz et al.; Freebase relational data for distant supervision on a Wikipedia corpus. Their method of training a factor graph model for relation extraction is more sophisticated then the pipeline approach of Mintz et al. The advantage of the factor graph model is that detecting entities and presence of a relation happens in one step and thus can mutually improve each other. For this they report an F1 increase from .31 to .4 in a Wikipedia held out setting. The other major contribution is an evaluation in a realistic scenario: Training on Freebase & Wikipedia and testing on a New York Time corpus. In this setting they achieve .25 F1, which resembles a drop of 37%. Although they built upon a large number of methods, including CRF, factor graph model, selectional preference templates and patterns, they also use supervised POS, NER and parser,

partially explaining the F1 drop in the out-of-domain setting.

3 Method

Our terminology is consistent with (Mintz et al., 2009). We use the term 'relation' in the meaning of an ordered, binary relation between two entities. We call these entities part A and part B, allowing us to specify the order. We refer to individual ordered pairs in the target relation as instances.

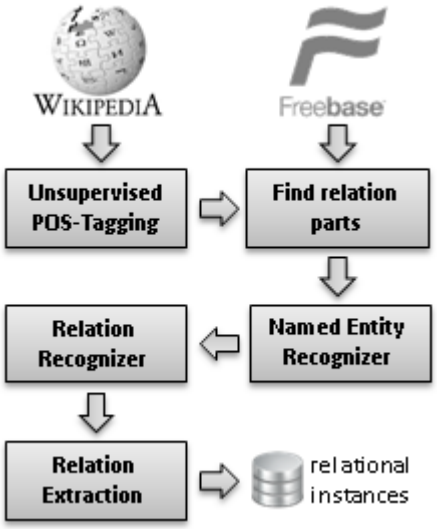

Figure 1: System architecture

Our method comprises the following steps, which can also be seen in Figure 1.

1. Read Freebase data, create training and test splits for validation.
2. Grab all Wikipedia sentences containing exactly one part A and one part B, not necessarily from the same instance of 1.
3. Train a named entity recognizer (NER), which is able to tag entities with three labels entity A, entity B, other O. (see 3.3)
4. Train a classifier (Relation Recognizer, RR) that separates sentences containing the target relation from those that do not. (see 3.4)
5. Use the classifier of step 3 and the NER of step 4 to find all sentences in Wikipedia that contain the target relation and extract both relation parts.
6. Compare the found relation instances of 5 with the held-out data of 1.

These steps will be explained in the following subsections.

3.1 Freebase Data

Freebase defines itself as "an open, Creative Commons licensed repository of structured data of almost 22 million entities."[4] It is collaboratively built out of different online sources as well as wiki-style contributions. According to Mintz et al. (2009) Freebase contains 116 million instances of 7300 relations between 9 million entities and a major source is text boxes and other tabular data from Wikipedia, as well as NNDB (biographical information), MusiBrainz (music) and SEC (financial and corporate data).

Although our method is generic for all relations and domains, we use a consistent example to explain the next steps. We chose the relation person/place_of_birth which has about 400 thousand instances. One of the instances is then (Juan Ramón Jiménez, Moguer).

For the evaluation we created ten randomized splits of the instances.

3.2 Wikipedia Data

We used a sample of 10 million sentences of the German and English Wikipedia, which was POS tagged with the unsupervised tagger "unsupos"[5]. Biemann (2009) shows that unsupervised POS tagging is possible at high quality and even can improve supervised methods. Also the quality for small training sets is improved, which can be traced to the fine-grained tagset that directly indicated entities. This is important as even large Freebase relations produce only a small amount of Wikipedia training data, which will be shown later. An example sentence of these POS tags looks like this:

```
Juan/6 Ramón/6 Jiménez/10 was/222
born/3 in/3 the/350 house/2 num-
ber/2 two/262 the/350 street/2
from/3 the/350 Ribera/1 de/157
Moguer/8
```

As training data for the classifiers in steps 3 and 4 we use all sentences in which exactly one entity A and one entity B occur.

We then assume that sentences in which A and B are from the same instance in our Freebase data show the target relation. This is true in the following example sentence: "Juan Ramón Jiménez was born in Moguer.", but can be also false if he also died there and we find a sentence like "Juan Ramón Jiménez died in Moguer."Nevertheless, while there might be sentences contributing wrong features, overall the large number of sentences contributing with right fea-

[4] http://wiki.freebase.com/wiki/What_is_Freebase?
[5] http://wortschatz.uni-leipzig.de/~cbiemann/software/unsupos.html

tures for the target relation outperforms small errors.

The sentences in which A and B are not from the same instance are used as negative examples. These are needed to train the RR classifier and also improve the quality of the NER. We tried using sentences in which only one entity or even none occurs as negative training data, but this only decreased the quality of the NER. The problem is, that there are many entities not being in the Freebase data for which the category O is learned. Same holds for the relation recognizer (RR), if we use sentences in which the target relation appears as negative examples, this decreases the quality of the RR.

For the 360k training instances in person/place_of_birth we found 2800 training sentences in Wikipedia, of which 250 are positive, meaning they contain both parts of one training instance.

3.3 Named Entity Recognizer (NER)

Current state of the art NER use Conditional Random Fields as theoretical basis (Lafferty et al., 2001). The Stanford NER[6] uses this basis and implements enough features, satisfying our purposes.

For describing the features we have the following abbreviations: w = word, t = tag, p = position, c = class, nw = next word, pt = previous tag and combinations, p(x) = probability of x and the colon "," meaning the conjunction.

The most important features we used for training the Stanford NER are:

- n-grams of classes, with classes A, B and O
- n-grams of words, with and without replacing A and B words with placeholders
- n-grams of letters, up to n=6
- probabilities of combinations of {previous, next} × {word, tag} and class, eg. `p(nw,c)`
- word pairs: `p(pw,w,c)` and `p(w,nw,c)`
- first, second, and third order class and tag sequence interaction features
- symbolic tags: `p(pt,t,nt,c)` and `p(t,nt,c)` and `p(pt,t,c)`
- symbolic word pairs: `(pw,nw,c)`
- disjunctions of words with distance four to the left or right, preserving the order but not the position
- combination of position in sentence and class

[6] http://www-nlp.stanford.edu/software/CRF-NER.shtml

The result of tagging our example sentence, which is actually also recognized by the RR is:

```
Juan/A Ramón/A Jiménez/A was/O
born/O in/O the/O house/O num-
ber/O two/O the/O street/O from/O
the/O Ribera/O de/O Moguer/B
```

3.4 Relation Recognizer (RR)

In order to decide whether or not a sentence contains the target relation we tried two different approaches. In the first variant we let the RR decide on the relation before tagging entities, in the second we tagged all sentences with our NER and then classified with the RR.

Classifying before tagging has the advantage that classifying is significantly faster than tagging. Yet, the disadvantage is that we can only use words and POS-tags as features, not the presenece/absence of named entities. Characterizing or discriminating n-grams are learnt with the following approach. At first all n-grams in the positive examples are counted and then ordered by their frequency. A second ranking list is generated for the negative examples in the same way. Then the rank differences for all entries are calculated. For example when "Childhood" appears at place thirty in the positive example ranking and at place hundred in the negative ranking list, then the rank difference would be seventy. At last the n-grams are ordered by their rank difference. This method essentially ranks common words lower as they appear in both input lists, as well as n-grams appearing more often in the negative example list. See Table 1 to get an impression which word unigrams, bigrams and trigrams were learned distinguishing well between positive and negative examples.

unigrams	**bigrams**
Childhood	Hall of
Poetry	the University
Career	Early life
inducted	was born
trigrams	
in the village	
the University of	
grew up in	
was born in	

Table 1: Characterizing n-grams

The second classifier also takes the classes assigned by the NER as features. This improves the classification but strongly depends on the quality of the NER. The output of our second classifier is a list of patterns which indicate a high probability that the sentence contains the target rela-

tion in case of a pattern match. As features for patterns, we use word and pos n-grams under the condition that they appear before A, between A and B, after A as well as for the case when B appears before A. Regarding only unigrams we get 12 features: {word, pos} × {AB, BA} × {before, between, after}

The top three word unigrams are listed in Table 2.

before	between	after
A before B		
Franciscan	Lovejoy	Harington
Childhood	born	Eartham
Biography	uprising	Doraly
B before A		
Kiltimagh	rue	Poetry
Noted	Arkham	governor
Residents	birthplace	Carolina

Table 2: Characterizing unigrams relative to relations

From these features we learn significant patterns in the way, that we take the 25% n-grams with the highest rank difference per section to build patterns in the form `before (A,B) between (A,B) after`, joining word and tag n-grams. We allow only one place to be empty and if there is a pattern with gaps that is part of a pattern without, we only take the one without gaps. After that we count the support in the positive sentences. The top unigram patterns with their support are shown in Table 3.

pos	words	mixed
1 A 3 B 2:32	A born B of:6	1 A in B :16
1 A 3 B:29	A born B was:5	1 A born B 2:16
A 3 B 2:23	Personal A was B 6:5	2 A 3 B _NUM_:17
1 A B 2:20	life A was B:5	1 A was B 2:16
1 A 3 B 6:20	Early A was B:5	1 A 3 B of:15

Table 3: Unigram patterns with highest support

The final step is to combine word/pos n-grams respectively the patterns with the named entity recognizer to extract entities from positive classified sentences.

4 Evaluation

At first we evaluate the two subsystems and then provide statistics for overall system performance.

Our test data for the complete evaluation is the person/place_of_birth relation. Although we chose one of the larger Freebase relations, it turned out that the representation in the Wikipedia data is very low: Out of 40k test instances in Freebase, only 28 appear in our corpus of 10 million Wikipedia sentences. To this we add the 182 sentences in which A and B appear from a different instance. The results of the evaluation are presented in Table 4.

P	R	F1	acc
NER			
.16	.091	.116	.077
RR with Patterns			
.136	.893	.236	.225
RR with n-grams			
.23	.5	.315	.708

Table 4: Precision, recall, F1 & accuracy for NER & RR

In the evaluation of the NER a true positive is the case in which an entity is found completely, the true negatives are the ones in which all words are correctly labeled as other O. Table 4 shows that, although we biased the NER to use A and B more often, we have very low recall and precision values. For finding single parts A and B this might be suitable, but in our case the low recall is the reason why we do not find any correct Freebase instance at all.

In order to avoid the dependency on the NER, the RR is evaluated using the correct labels. From Table 4 we see the high recall of the RR using patterns. This is something we aimed at, as the idea was to have two independent systems with a high recall, as in our hierarchical setup the recalls of the subsystems are multiplied, resulting in a medium recall for the complete setup. The difference in the F1 measure between the two RR variants can be explained by their different feature structure. In the case of patterns the entities have to be matched before and afterwards a match in unigrams in the right parts is required. In the other case only uni- to trigrams are matched.

The overall RE system performance cannot be evaluated as the data we used is on the one hand too spare and on the other hand our subsystem performance is too low. Nevertheless we summarize related work's performance in this task in order to compare future systems.

As shown in Yao et al. (2010) evaluation must be performed on out-of-domain corpora in order to be realistic for real world scenarios. We approximate this by applying the same loss Yao et al. perceive in their system when comparing other work in Table 5.

system	intra-domain F1	extra-domain F1
(Mintz et al.)	.34	.21
(Hoffmann et al.)	.61	.38
(Yao et al.)	.40	.25

Table 5: RE system comparison

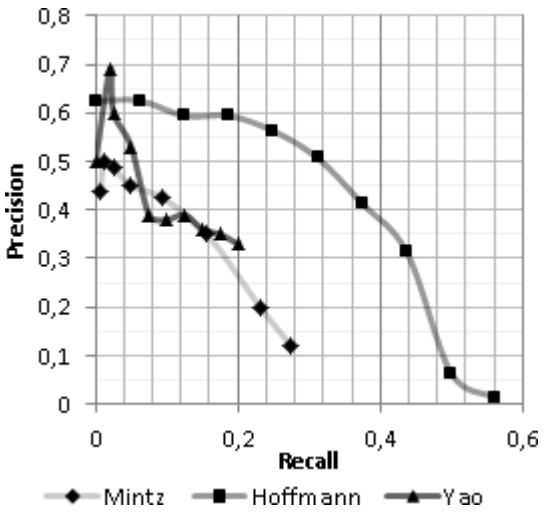

Figure 2: Recall & precision for current RE systems

A comparison by recall and precision is shown in Figure 2. The F1 performance of the systems is shown in Figure 3.

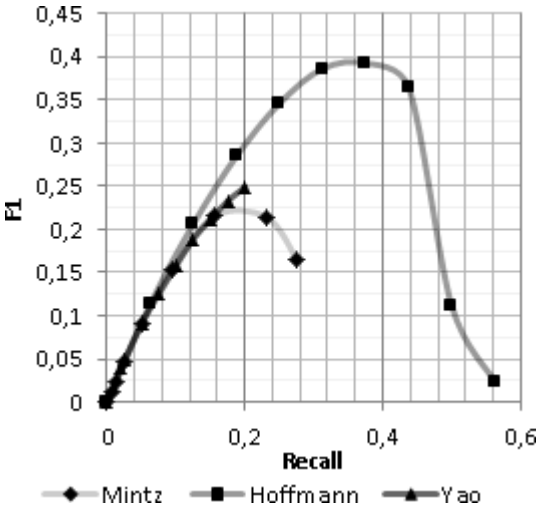

Figure 3: Recall and F1 for current RE systems

This shows that current systems do not exceed a recall of .56, while their best F1 performance is at a recall level of .38.

5 Conclusion & Future Work

In this paper we presented a novel approach to distant supervision. Current state of the art distant supervision systems (see section 2) exploit plenty of features gathered by supervised systems, such as POS tagger, NER and parser. We developed a completely unsupervised system, based on state-of-the-art unsupervised POS tagging (Biemann, 2009). Our contribution is a distant supervised NER and Relation Recognizer. Combining those leads to a Relation Extraction system – which should generalize enough to work on out-of-domain corpora without a vast performance loss, as current systems suffer from.

The main problem with our work is, that the subsystems perform too badly, with the consequence that the RE system does not extract any usable results.

Clearly, work has to be done in improving the subsystems. Most promising is the approach of Yao et al. (2010), combining the two steps of NER and RR in one, with a factor graph model. Also the performance of the NER might be improved using a classifier designed for partially labeled data.

If the system performs well enough on the current Wikipedia data, the next step will be to evaluate it on out-of-domain corpora, proving that there is no significant decrease in performance.

Reference

Banko, M. & Etzioni, O., 2008. The tradeoffs between open and traditional relation extraction. *Proceedings of ACL*.

Biemann, C., 2009. Unsupervised Part-of-Speech Tagging in the Large. *Res. Lang. Comput. 7*.

Bunescu, R. & Mooney, R., 2007. Learning to extract relations from the web using minimal supervision. *ACL-07*, pp.576-83.

Finkel, J.R., Grenager, T. & Manning, C., 2005. Incorporating non-local information into information extraction systems by gibbs sampling. *ACL-05*, pp.363-70.

Hoffmann, R., Zhang, C. & Weld, D.S., 2010. Learning 5000 Relational Extractors. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pp.286-95.

Lafferty, J., McCallum, A. & Pereira, F., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.

Mintz, M., Bills, S., Snow, R. & Jurafsky, D., 2009. Distant supervision for relation extraction without labeled data. *Proceedings of ACL-IJCNLP 2009*.

Morgan, A.A. et al., 2004. Gene name identification and normalization using a model organism database. *J. of Biomedical Informatics, 37*, pp.396-410.

Riedel, S., Yao, L. & Mccallum, A., 2010. Modeling Relations and Their Mentions without Labeled

Text. *ECML PKDD 2010, Part III, LNAI 6323*, pp.148-63.
Rink, B. & Harabagiu, S., 2010. UTD: Classifying Semantic Relations by Combining Lexical and Semantic Resources. *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp.256-59.
Rozenfeld, B. & Feldman, R., 2008. Self-supervised relation extraction from the web. *Knowledge and Information Systems*, pp.17-33.
Snow, R., Jurafsky, D. & Ng, A.Y., 2005. Learning syntactic patterns for automatic hypernym discovery. *NIPS 17*, pp.1297-304.
Yan, Y. et al., 2009. Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from theWeb. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pp.1021-29.
Yao, L., Riedel, S. & McCallum, A., 2010. Collective Cross-Document Relation ExtractionWithout Labelled Data. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp.1013-23.
Zhu, J. et al., 2009. StatSnowball: a Statistical Approach to Extracting Entity Relationships. *ACM 978-1-60558-487-4/09/04*.