Collective Cross-Document Relation Extraction Without Labelled Data

Limin Yao **Sebastian Riedel** **Andrew McCallum**
University of Massachusetts, Amherst
{lmyao,riedel,mccallum}@cs.umass.edu

Abstract

We present a novel approach to relation extraction that integrates information across documents, performs global inference and requires no labelled text. In particular, we tackle relation extraction and entity identification jointly. We use distant supervision to train a factor graph model for relation extraction based on an existing knowledge base (Freebase, derived in parts from Wikipedia). For inference we run an efficient Gibbs sampler that leads to linear time joint inference. We evaluate our approach both for an in-domain (Wikipedia) and a more realistic out-of-domain (New York Times Corpus) setting. For the in-domain setting, our joint model leads to 4% higher precision than an isolated local approach, but has no advantage over a pipeline. For the out-of-domain data, we benefit strongly from joint modelling, and observe improvements in precision of 13% over the pipeline, and 15% over the isolated baseline.

1 Introduction

Relation Extraction is the task of predicting semantic relations over entities expressed in structured or semi-structured text. This includes, for example, the extraction of employer-employee relations mentioned in newswire, or protein-protein interactions expressed in biomedical papers. It also includes the prediction of entity types such as country, citytown or person, if we consider entity types as unary relations.

A particularly attractive approach to relation extraction is based on *distant supervision*.[1] Here in place of annotated text, only an existing knowledge base (KB) is needed to train a relation extractor (Mintz et al., 2009; Bunescu and Mooney, 2007; Riedel et al., 2010). The facts in the KB are heuristically aligned to an unlabelled training corpus, and the resulting alignment is the basis for learning the extractor.

Naturally, the predictions of a distantly supervised relation extractor will be less accurate than those of a supervised one. While facts of existing knowledge bases are inexpensive to come by, the heuristic alignment to text will often lead to noisy patterns in learning. When applied to unseen text, these patterns will produce noisy facts. Indeed, we find that extraction precision still leaves much room for improvement. This room is not as large as in previous work (Mintz et al., 2009) where target text and training KB are closely related. However, when we use the knowledge base Freebase (Bollacker et al., 2008) and the New York Times corpus (Sandhaus, 2008), we observe very low precision. For example, the precision of the top-ranked 50 `nationality` relation instances is only 28%.

On inspection, it turns out that many of the errors can be easily identified: they amount to violations of basic compatibility constraints between facts. In particular, we observe unsatisfied *selectional preferences* of relations towards particular entity types as types of their arguments. An example is the fact that the first argument of `nationality` is always a `person` while the second is a `country`. A simple way to address this is a pipeline: first predict entity types, and then condition on these when predicting relations. However, this neglects the fact that relations could as well be used to help entity type prediction.

[1] Also called *self training*, or *weak supervision*.

While there is some existing work on enforcing such constraints in a joint fashion (Roth and Yih, 2007; Kate and Mooney, 2010; Riedel et al., 2009), they are not directly applicable here. The difference is the amount of facts they take into account at the same time. They focus on single sentence extractions, and only consider very few interacting facts. This allows them to work with exact optimization techniques such as (Integer) Linear Programs and still remain efficient.[2] However, when working on a sentence level they fail to exploit the redundancy present in a corpus. Moreover, the fewer facts they consider at the same time, the lower the chance that some of these will be incompatible, and that modelling compatibility will make a difference.

In this work we present a novel approach that performs relation extraction across documents, enforces selectional preferences, and needs no labelled data. It is based on an undirected graphical model in which variables correspond to facts, and factors between them measure compatibility. In order to scale up, we run an efficient Gibbs-Sampler at inference time, and train our model using SampleRank (Wick et al., 2009). In practice this leads to a runtime behaviour that is linear in the size of the corpus. For example, 200,000 documents take less than three hours for training and testing.

For evaluation we consider two scenarios. First we follow Mintz et al. (2009), use Freebase as source of distant supervision, and employ Wikipedia as source of unlabelled text—we will call this an in-domain setting. This scenario is somewhat artificial in that Freebase itself is partially derived from Wikipedia, and in practice we cannot expect text and training knowledge base to be so close. Hence we also evaluate our approach on the New York Times corpus (out-of-domain setting).

For in-domain data we make the following finding. When we compare to an isolated baseline that makes no use of entity types, our joint model improves average precision by 4%. However, it does not outperform a pipelined system. In the out-of-domain setting, our collective model substantially outperforms both other approaches. Compared to the isolated baseline, we achieve a 15% increase in precision. With respect to the pipeline approach, the increase is 13%.

In the following we will first give some background information on relation extraction with distant supervision. Then we will present our graphical model as well as the inference and learning techniques we apply. After discussing related work, we present our empirical results and conclude.

2 Background

In this section we will introduce the terminology and concepts we use throughout the paper. We will also give a brief introduction to relation extraction, in particular in the context of distant supervision.

2.1 Relations

We seek to extract facts about entities. Example entities would be the company founder BILL GATES, the company MICROSOFT, and the country USA. A *relation* R is a set of tuples $\mathbf{c}$ over entities. We will follow (Mintz et al., 2009) and call the term $R(c_1, \ldots c_n)$ with $\mathbf{c} \in R$ a *relation instance*.[3] It denotes the membership of the tuple $\mathbf{c}$ in the relation R. For example, `founded` (BILL GATES, MICROSOFT) is a relation instance denoting that BILL GATES and MICROSOFT are related in the `founded` relation.

In the following we will always consider some set of *candidate tuples* C that may or may not be related. We define $C_n \subset C$ to be set of all n-ary tuples in C. Note that while our definition considers general n-nary relations, in practice we will restrict us to unary and binary relations C_1 and C_2.

Following previous work (Mintz et al., 2009; Zelenko et al., 2003; Culotta and Sorensen, 2004) we make one more simplifying assumption: every candidate tuple can be member of at most one relation.

2.2 Entity Types

An entity can be of one or several *entity types*. For example, BILL GATES is a `person`, and a company `founder`. Entity types correspond to the special case of relations with arity one, and will be treated as such in the following.

[2] The pyramid algorithm of Kate and Mooney (2010) may scale well, but it is not clear how to apply their scheme to cross-document extraction.

[3] Other commonly used terms are relational facts, ground facts, ground atoms, and assertions.

We care about entity types for two reasons. First, they can be important for downstream applications: if consumers of our extracted facts know the type of entities, they can find them more easily, visualize them more adequately, and perform operations specific to these types (write emails to persons, book a hotel in a city, etc.). Second, they are useful for extracting binary relations due to selectional preferences—see section 2.6.

2.3 Mentions

In natural language text spans of tokens are used to refer to entities. We call such spans *entity mentions*. Consider, for example, the following sentence snippet:

(1) Political opponents of President **Evo Morales** of **Bolivia** have in recent days stepped up...

Here "Evo Morales" is an entity mention of president EVO MORALES, and "Bolivia" a mention of the country BOLIVIA he is the president of.

People often express relations between entities in natural language texts by mentioning the participating entities in specific syntactic and lexical patterns. We will refer to any tuple of mentions of entities $(e_1, \ldots e_n)$ in a sentence as *candidate mention tuple*. If such a candidate expresses the relation R, then it is a *relation mention* of the relation instance $R(e_1, \ldots, e_n)$.

Consider again example 1. Here the pair of entity mentions ("Evo Morales", "Bolivia") is a *candidate mention tuple*. In fact, in this case the candidate is indeed a *relation mention* of the relation instance `nationality` (EVO MORALES, BOLIVIA).

2.4 Relation Extraction

We define the task of relation extraction as follows. We are given a corpus of documents and a set of target relations. Then we are asked to predict all relation instances I so that for each $R(\mathbf{c}) \in I$ there exists at least one relation mention in the given corpus.

The above definition covers a range of existing approaches by varying over what we define as target corpus. On one end, we have extractors that process text on a per sentence basis (Zelenko et al., 2003; Culotta and Sorensen, 2004). On the other end, we have methods that take relation mentions from several documents and use these as input features (Mintz et al., 2009; Bunescu and Mooney, 2007).

There is a compelling reason for performing relation extraction within a larger scope that considers mentions across documents: redundancy. Often facts are mentioned in several sentences and documents. Some of these mentions may be difficult to parse, or they use unseen patterns. But the more mentions we consider, the higher the probability that one does parse, and fits a pattern we have seen in the training data.

Note that for relation extraction that considers more than a single mention we have to solve the coreference problem in order to determine which mentions refer to the same entity. In the following we will assume that coreference clusters are provided by a preprocessing step.

2.5 Distant Supervision

In relation extraction we often encounter a lack of explicitly annotated text, but an abundance of structured data sources such as company databases or collaborative knowledge bases like Freebase. In order to exploit this, many approaches use simple but effective heuristics to align existing facts with unlabelled text. This labelled text can then be used as training material of a supervised learner.

One heuristic is to assume that each candidate mention tuple of a training fact is indeed expressing the corresponding relation (Bunescu and Mooney, 2007). Mintz et al. (2009) refer to this as the *distant supervision assumption*.

Clearly, this heuristic can fail. Let us again consider the `nationality` relation between EVO MORALES and BOLIVIA. In an 2007 article of the New York Times we find this relation mention candidate:

(2) ...the troubles faced by **Evo Morales** in **Bolivia**...

This sentence does not directly express that EVO MORALES is a citizen of BOLIVIA, and hence violates the distant supervision assumption. The problem with this observation is that at training time we may learn a relatively large weight for the feature "<Entity1> in <Entity2>" associated with

nationality. When testing our model we then encounter a sentence such as

(3) Arrest Warrant Issued for **Richard Gere** in **India**.

that leads us to extract that RICHARD GERE is a citizen of INDIA.

2.6 Global Consistency of Facts

As discussed above, distant supervision can lead to noisy extractions. However, such noise can often be easily identified by testing how compatible the extracted facts are to each other. In this work we are concerned with a particular type of compatibility: selectional preferences.

Relations require, or prefer, their arguments to be of certain types. For example, the nationality relation requires the first argument to be a person, and the second to be a country. On inspection, we find that these preferences are often not satisfied in a baseline distant supervision system akin to Mintz et al. (2009). This often results from patterns such as "<Entity1> in <Entity2>" that fire in many cases where <Entity2> is a location, but not a country.

3 Model

Our observations in the previous section suggest that we should (a) explicitly model compatibility between extracted facts, and (b) integrate evidence from several documents to exploit redundancy. In this work we choose a Conditional Random Field (CRF) to achieve this. CRFs are a natural fit for this task: They allow us to capture correlations in an explicit fashion, and to incorporate overlapping input features from multiple documents.

The hidden output variables of our model are $\mathbf{Y} = (Y_{\mathbf{c}})_{\mathbf{c} \in C}$. That is, we have one variable $Y_{\mathbf{c}}$ for each candidate tuple $\mathbf{c} \in C$. This variable can take as value any relation in C with the same arity as $\mathbf{c}$. See example relation variables in figure 1.

The observed input variables $\mathbf{X}$ consists of a family of variables $\mathbf{X_c} = \left(\mathbf{X}^1_{\mathbf{c}}, \ldots \mathbf{X}^m_{\mathbf{c}}\right)_{m \in M}$ for each candidate tuple $\mathbf{c}$. Here $\mathbf{X}^i_{\mathbf{c}}$ stores relevant observations we make for the i-th candidate mention tuple of $\mathbf{c}$ in the corpus. For example, $\mathbf{X}^1_{\text{BILL GATES,MICROSOFT}}$ in figure 1 would contain, among others, the pattern "[M2] was founded by [M1]".

3.1 Factor Templates

Our conditional probability distribution over variables $\mathbf{X}$ and $\mathbf{Y}$ is defined using using a set $\mathcal{T}$ of *factor templates*. Each template $T_j \in \mathcal{T}$ defines a set of factors $\{(\mathbf{y}_i, \mathbf{x}_i)\}$, a set K_j of feature indices, parameters $\left\{\theta^j_k\right\}_{k \in K_j}$ and feature functions $\left\{f^j_k\right\}_{k \in K_j}$. Together they define the following conditional distribution:

$$p\left(\mathbf{y}|\mathbf{x}\right) = \frac{1}{Z_{\mathbf{x}}} \prod_{T_j \in \mathcal{T}} \prod_{(\mathbf{y}_i, \mathbf{x}_i) \in T_j} e^{\sum_{k \in K_j} \theta^j_k f^j_k(\mathbf{y}_i, \mathbf{x}_i)} \tag{4}$$

In our case the set $\mathcal{T}$ consists of four templates we will describe below. We construct this graphical model using FACTORIE (McCallum et al., 2009), a probabilistic programming language that simplifies the construction process, as well as inference and learning.

3.1.1 Bias Template

We use a bias template T_{Bias} that prefers certain relations *a priori* over others. When the template is unrolled, it creates one factor per variable $Y_{\mathbf{c}}$ for candidate tuple $\mathbf{c} \in C$. The template also consists of one weight θ^{Bias}_r and feature function f^{Bias}_r for each possible relation r. f^{Bias}_r fires if the relation associated with tuple $\mathbf{c}$ is r.

3.1.2 Mention Template

In order to extract relations from text, we need to model the correlation between relation instances and their mentions in text. For this purpose we define the template T_{Men} that connects each relation instance variable $Y_{\mathbf{c}}$ with its observed mention variables $\mathbf{X_c}$. Crucially, this template gathers mentions from multiple documents, and enables us to exploit redundancy.

The feature functions of this template are taken from Mintz et al. (2009). This includes features that inspect the lexical content between entity mentions in the same sentence, and the syntactic path between them. One example is

$$f^{\text{Men}}_{101}\left(y_{\mathbf{c}}, \mathbf{x_c}\right) \stackrel{\text{def}}{=} \begin{cases} 1 & y_{\mathbf{c}} = \texttt{founded} \wedge \exists i \text{ with} \\ & \text{"M2 was founded by M1"} \in \mathbf{x}^i_{\mathbf{c}} \\ 0 & \text{otherwise} \end{cases}.$$

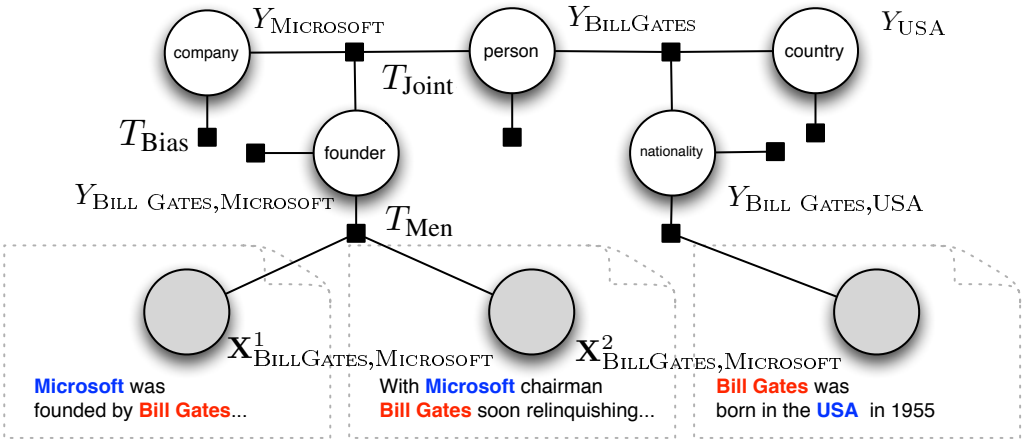


Figure 1: Factor Graph of our model that captures selectional preferences and functionality constraints. For readability we only label a subsets of equivalent variables and factors. Note that the graph shows an example assignment to variables.

It tests whether for any mentions of the candidate tuple the phrase "founded by" appears between the mentions of the argument entities.

3.1.3 Selectional Preferences Templates

To capture the correlations between entity types and relations the entities participate in, we introduce the template T_{Joint}. It connects a relation instance variable $Y_{e_1,\ldots,e_n}$ to the individual entity type variables $Y_{e_1},\ldots,Y_{e_n}$. To measure the compatibility between relation and entity variables, we use one feature $f^{\text{Joint}}_{r,t_1\ldots t_a}$ (and weight $\theta^{\text{Joint}}_{r,t_1\ldots t_a}$) for each combination of relation and entity types $r, t_1 \ldots t_a$.

$f^{\text{Joint}}_{r,t_1\ldots t_a}$ fires when the factor variables are in the state $r, t_1 \ldots t_a$. For example, $f^{\text{Joint}}_{\texttt{founded,person,company}}$ fires if Y_{e_1} is in state `person`, Y_{e_2} in state `company`, and Y_{e_1,e_2} in state `founded`.

We also add a template T_{Pair} that measures the pairwise compatibility between the relation variable $Y_{e_1,\ldots,e_a}$ and each entity variable Y_{e_i} in isolation. Here we use features $f^{\text{Pair}}_{i,r,t}$ that fire if e_i is the i-th argument of $\mathbf{c}$, has the entity type t and the candidate tuple $\mathbf{c}$ is labelled as instance of relation r. For example, $f^{\text{Pair}}_{1,\texttt{founded,person}}$ fires if Y_{e_1}(argument $i = 1$) is in state `person`, and Y_{e_1,e_2} in state `founded`, regardless of the state of Y_{e_2}.

3.2 Inference

There are two types of inference we have to perform: sampling from the posterior during training (see section 3.3), and finding the most likely configuration (aka MAP inference). In both settings we employ a Gibbs sampler (Geman and Geman, 1990) that randomly picks a variable $Y_{\mathbf{c}}$ and samples its relation value conditioned on its Markov Blanket. At test time we decrease the temperature of our sampler in order to find an approximation of the MAP solution.

3.3 Training

Most learning methods need to calculate the model expectations (Lafferty et al., 2001) or the MAP configuration (Collins, 2002) before making an update to the parameters. This step of inference is usually the bottleneck for learning, even when performed approximately.

SampleRank (Wick et al., 2009) is a rank-based learning framework that alleviates this problem by performing parameter updates *within* MCMC inference. Every pair of consecutive samples in the MCMC chain is ranked according to the model and the ground truth, and the parameters are updated when the rankings disagree. This update can follow different schemes, here we use MIRA (Crammer and Singer, 2003). This allows the learner to acquire more supervision per instance, and has led to efficient training for models in which inference

is expensive and generally intractable (Singh et al., 2009).

4 Related Work

Distant Supervision Learning to extract relations by using distant supervision has raised much interest in recent years. Our work is inspired by Mintz et al. (2009) who also use Freebase as distant supervision source. We also heuristically align our knowledge base to text by making the distant supervision assumption (Bunescu and Mooney, 2007; Mintz et al., 2009). However, in contrast to these previous approaches, and other related distant supervision methods (Craven and Kumlien, 1999; Weld et al., 2009; Hoffmann et al., 2010), we perform relation extraction collectively with entity type prediction.

Schoenmackers et al. (2008) use entailment rules on assertion extracted by TextRunner to increase recall. They also perform cross-document probabilistic inference based on Markov Networks. However, they do not infer the types of entities and work in an open IE setting.

Selectional Preferences In the context of supervised relation extraction, selectional preferences have been applied. For example, Roth and Yih (2007) have used Linear Programming to enforce consistency between entity types and extracted relations. Kate and Mooney (2010) use a pyramid parsing scheme to achieve the same. Riedel et al. (2009) use Markov Logic to model interactions between event-argument relations for biomedical event extraction. However, their work is (a) supervised, and (b) performs extraction on a per-sentence basis.

Carlson et al. (2010) also use selectional preferences. However, instead of exploiting them for training a graphical model using distant supervision, they use selectional preferences to improve a bootstrapping process. Here in each iteration of bootstrapping, extracted facts that violate compatibility constraints will not be used to generate additional patterns in the next iteration.

5 Experiments

We set up experiments to answer the following questions: (i) Does the explicit modelling of selectional preferences improve accuracy? (ii) Can we also perform joint entity and relation extraction in a pipeline and achieve similar results? (iii) How does our cross-document approach scale?

To answer these questions we carry out experiments on two data sets, Wikipedia and New York Times articles, and use Freebase as distant supervision source for both.

5.1 Experimental Setup

We follow Mintz et al. (2009) and perform two types of evaluation: held-out and manual. In both cases we have a training and a test corpus of documents, and training and test sets of entities. For held-out evaluation we split the set of entities in Freebase into training and test sets. For manual evaluation we use all Freebase entities during training. For testing we use all entities that appear in the test document corpus.

For both training and testing we then choose the candidate tuples C that may or may not be relation instances. To pick the entities C_1 we want to predict entity types for, we choose all entities that are mentioned at least once in the train/test corpus. To pick the entity pairs C_2 that we want to predict the relations of, we choose those that appear at least once together in a sentence.

The set of candidates C will contain many tuples which are not related in any Freebase relations. For efficiency, we filter out a large fraction of these *negative* candidates for training. The number of negative examples we keep is chosen to be about 10 times the number of positive candidates. This number stems from trading-off the accuracy it leads to and the increased training time it requires.

For both manual and held-out evaluation we rank extracted test relation instances in the MAP state of the network. This state is found by sampling 20 iterations with a low temperature of 0.00001. The ranking is done according to the log linear score that the assigned relation for a candidate tuple gets from the factors in its Markov Blanket. For optimal performance, the score is normalized by the number of relation mentions.

For manual evaluation we pick the top ranked 50 relation instances for the most frequent relations. We ask three annotators to inspect the mentions of these relation instances to decide whether they are correct. Upon disagreement, we use majority vote. To summarize precisions across relations, we take

their average, and their average weighted by the proportion of predicted instances for the given relation.

5.1.1 Data preprocessing

We preprocess our textual data as follows: We first use the Stanford named entity recognizer (Finkel et al., 2005) to find entity mentions in the corpus. The NER tagger segments each document into sentences and classifies each token into four categories: PERSON, ORGANIZATION, LOCATION and NONE. We treat consecutive tokens which share the same category as single entity mention. Then we associate these mentions with Freebase entities. This is achieved by performing a string match between entity mention phrases and the canonical names of entities as present in Freebase.

For each candidate tuple $\mathbf{c}$ with arity 2 and each of its mention tuples i we extract a set of features $\mathbf{X}_{\mathbf{c}}^{i}$ similar to those used in (Mintz et al., 2009): lexical, Part-Of-Speech (POS), named entity and syntactic features, i.e. features obtained from the dependency parsing tree of a sentence. We use the openNLP POS tagger[4] to obtain POS tags and employ the MaltParser (Nivre et al., 2004) for dependency parsing. For candidate tuples with arity 1 (entity types) we use the following features: the entity's word form, the POS sequence, the head of the entity in the dependency parse tree, the Stanford named entity tag, and the left and right words to the current entity mention phrase.

5.1.2 Configurations

We apply the following configurations of our factor graphs. As our baseline, and roughly equivalent to previous work (Mintz et al., 2009), we pick the templates T_{Bias} and T_{Men}. These describe a fully disconnected graph, and we will refer to this configuration as *isolated*. Next, we add the templates T_{Joint} and T_{Pair} to model selectional preferences, and refer to this setting as *joint*.

In addition, we evaluate how well selectional preferences can be captured with a simple *pipeline*. For this pipeline we first train an isolated system for entity type prediction. Then we use the output of the entity type prediction system as input for the relation extraction system.

[4] available at http://opennlp.sourceforge.net/

5.1.3 Entity types and Relation types

Freebase contains many relation types and only a subset of those relation types occur frequently in the corpus. Since classes with very few training instances are generally hard to learn, we restrict ourselves to the 54 most frequently mentioned relations. These include, for example, `nationality`, `contains`, `founded` and `place_of_birth`. Note that we convert two Freebase non-binary temporal relations to binary relations: `employment_tenure` and `place_lived`. In both cases we simply disregard the temporal information in the Freebase data.

As our main focus is relation extraction, we restrict ourselves to entity types compatible with our selected relations. To this end we inspect the Freebase schema information provided for each relation, and include those entity types that are declared as arguments of our relations. This leads to 10 entity types including `person`, `citytown`, `country`, and `company`.

Note that a Freebase entity can have several types. We pick one of these by choosing the most specific one that is a member of our entity type subset, or `MISC` if no such member exists.

5.2 Wikipedia

In our first set of experiments we train and test using Wikipedia as the text corpus. This is a comparatively easy scenario because the facts in Freebase are partly derived from Wikipedia, hence there is an increased chance of properly aligning training facts and text. This is similar to the setting of Mintz et al. (2009).

5.2.1 Held Out Evaluation

We split 1,300,000 Wikipedia articles into training and test sets. Table 1 shows the statistics for this split. The last row provides the number of negative relation instances (candidates which are not related according to Freebase) associated with each data set.

Figure 2 shows the precision-recall curves of relation extraction for held-out data of various configurations. We notice a slight advantage of the joint approach in the low recall area. Moreover, the joint model predicts more relation instances, as can be seen by its longer line in the graph.

For higher recall, the joint model performs slightly worse. On closer inspection, we find that

	Wikipedia		NYT	
	Train	Test	Train	Test
#Documents	900K	400K	177K	39K
#Entities	213K	137K	56K	27K
#Positive	36K	24K	5K	2K
#Negative	219K	590K	64K	94K

Table 1: The statistics of held-out evaluation on Wikipedia and New York Times.

	Isolated	Pipeline	Joint
Wikipedia	0.82	0.87	0.86
Wikipedia (w)	0.95	0.94	0.95
NYT	0.63	0.65	0.78
NYT (w)	0.78	0.82	0.94

Table 2: Average and weighted (w) average precision over frequent relations for New York Times and Wikipedia data, based on manual evaluation.

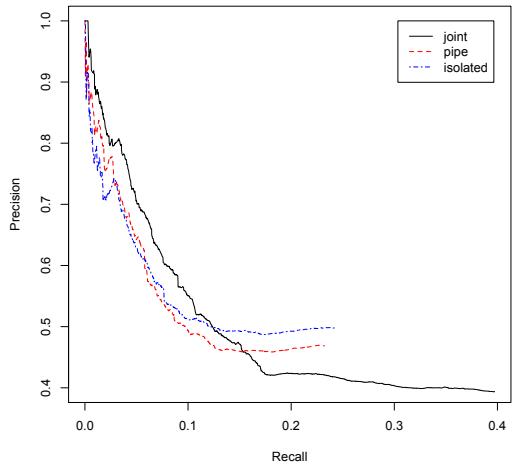


Figure 2: Precision-recall curves for various setups in Wikipedia held-out setting.

this observation is somewhat misleading. Many of the predictions of the joint model are not in the held-out test set derived from Freebase, but nevertheless correct. Hence, to understand if one system really outperforms another, we need to rely on manual evaluation.

Note that the figure only considers binary relations—for entity types all configurations perform similarly.

5.2.2 Manual Evaluation

As mentioned above, held-out evaluation in this context suffers from false negatives in Freebase. Table 2 therefore shows the results of our manual evaluation. They are based on the average, and weighted average, of the precisions for the relation instances of the most frequent relations. We notice that here all systems perform comparably for weighted average precision. For average precision we see an advantage for both the pipeline and the joint model over the isolated system.

One reason for similar weighted average precisions is the fact that all approaches accurately predict a large number of `contains` instances. This is due to very regular and simple patterns in Wikipedia. For example, most articles on towns start with "A is a municipality in the district of B in C, D." For these sentences, the relative position of two location mentions is a very good predictor of `contains`. When used as a feature, it leads to high precision for all models. And since `contains` instances are most frequent, and we take the weighted average, results are generally close to each other.

To summarize: in this in-domain setting, modelling compatibility between entity types and relations helps to improve average precision, but not weighted average precision. This holds for both the joint and the pipeline model. However, we will see how this changes substantially when moving to an out-of-domain scenario.

5.3 New York Times

For our second set of experiments we use New York Times data as training and test corpora. As we argued before, this is expected to be the more difficult—and more realistic—scenario.

5.3.1 Held-out Evaluation

We choose all articles of the New York times during 2005 and 2006 as training corpus. As test corpus we use the first 6 months of 2007.

Figure 3 shows precision-recall curves for our various setups. We see that jointly modelling entity

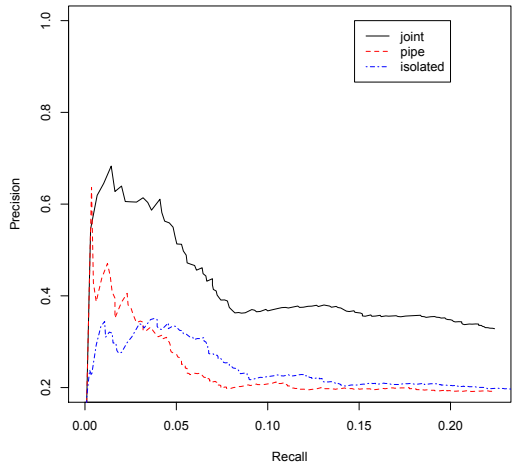


Figure 3: Precision-recall curves for various setups in New York Times held-out setting.

types and relations helps to improve precision.

Due to the smaller overlap between Freebase and NYT data, figure 3 also has to be taken with more caution. The systems may predict correct relation instances that just do not appear in Freebase. Hence manual evaluation is even more important.

When evaluating entity precision we find that for both models it is about 84%. This raises the question why the joint entity type and relation extraction model outperforms the pipeline on relations. We take a close look at the entities which participate in relations and find that joint model performs better on most entity types, for example, `country` and `citytown`. We also look at the relation instances which are predicted by both systems and find that the joint model does predict correct entity types when the pipeline mis-predicts. And exactly these mis-predictions lead the pipeline astray. Considering binary relation instances where the pipeline fails but the joint model does not, we observe an entity precision of 76% for the pipeline and 86% for our joint approach. The joint model fails to correctly predict some entity types that the pipeline gets right, but these tend to appear in contexts where relation instances are easy to extract without considering entity types.[5]

Relation Type	Iso.	Pipe	Joint
`contains`	0.92	0.98	0.96
`nationality`	0.28	0.64	0.82
`plc_lived`	0.88	0.70	0.96
`plc_of_birth`	0.32	0.20	0.25
`works_for`	0.96	0.98	0.98
`plc_of_death`	0.24	0.40	0.42
`children`	1.00	0.92	0.98
`founded`	0.42	0.34	0.71

Table 3: Precision at 50 for the most frequent relations on New York Times

5.3.2 Manual Evaluation

Manually evaluated precision for New York Times data can be seen in table 2. In contrast to the Wiki setting, here modelling entity types and relations jointly makes a substantial difference. For average precision, our joint model improves over the isolated baseline by 15%, and over the pipeline by 13%. Similar improvements can be observed for weighted average precision.

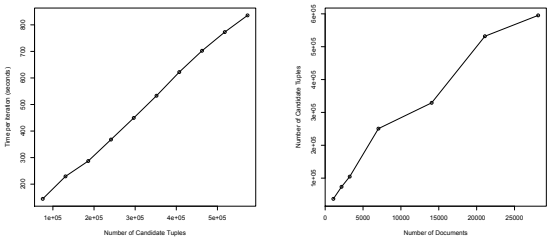
Let us look at a break-down of precisions with respect to different relations shown in table 3. We see dramatic improvements for `nationality` and `founded` when applying the joint model. Note that the `nationality` relation takes a larger part in the predicted relation instances of the joint model and hence contributes significantly to the weighted average precision.

5.4 Scalability

We propose to perform joint inference for large scale information extraction. An obvious concern in this scenario is scalability. In practice we find that inference (and hence learning) in our model scales linearly with the number of candidate tuples. This can be seen in figure 4a. It is to be expected since the number of candidates equals the number of variables the sampler has to process in each iteration.

The above observation also means that our approach scales linearly with corpus size. To illustrate

[5]Note that our learned preferences are soft, and hence can be violated in case of wrong entity type predictions.



(a) CPU time (b) Candidate tuples

Figure 4: CPU time for one iteration per candidate tuple, and candidate tuples per document.

this, figure 4b shows how the number of candidates scales with the number of documents. Again we observe a linear behavior. Since both are linear, we can say that our joint approach is linear in the number of documents.

Total training and test times are moderate, too. For example, the held-out experiments with 200,000 NYT documents finish within three hours.

6 Conclusion

This paper presents a novel approach to extracting relational facts from text. Akin to previous work in relation extraction with distant supervision, we require no annotated text. However, instead extracting facts in isolation, we model interactions between facts in order to improve precision. In particular, we capture selectional preferences of relations. These preferences are modelled in a cross-document fashion using a large scale factor graph. We show inference and learning can be efficiently performed in linear time by Gibbs Sampling and SampleRank. When applied to out-of-domain text, this approach leads to a 15% increase in precision over an isolated baseline, and a 13% improvement over a pipelined system.

A crucial aspect of our approach is its extensibility. Since it is exclusively framed in terms of an undirected graphical model, it is conceptually easy to extend it to other types of compatibilities, such as functionality constraints. It could also be extended to tackle coreference resolution. Eventually we seek to model the complete process of the automatic construction of KB within this framework, and capture dependencies between extractions in a joint and principled fashion. As we have seen here, in particular when learning is less supervised and extractions are noisy, capturing such interactions is paramount.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and in part by UPenn NSF medium IIS-0803847. The University of Massachusetts also gratefully acknowledges the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA. ACM.

Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45rd Annual Meeting of the Association for Computational Linguistics (ACL '07)*.

Andrew Carlson, Justin Betteridge, Richard Wang, Estevam Hruschka, and Tom Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Third ACM International Conference on Web Search and Data Mining (WSDM '10)*.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical methods in natural language processing (EMNLP '02)*, volume 10, pages 1–8.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

M. Craven and J. Kumlien. 1999. Constructing biological knowledge-bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86, Germany.

Aron Culotta and Jeffery Sorensen. 2004. Dependency tree kernels for relation extraction. In *42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pages 363–370, June.

S. Geman and D. Geman. 1990. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. pages 452–472.

Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. 2010. Learning 5000 relational extractors. In *ACL*.

Rohit J. Kate and Raymond J. Mooney. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL' 10)*.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*.

Andrew McCallum, Karl Schultz, and Sameer Singh. 2009. Factorie: Probabilistic programming via imperatively defined factor graphs. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1249–1257.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL '09)*, pages 1003–1011. Association for Computational Linguistics.

J. Nivre, J. Hall, and J. Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of CoNLL*, pages 49–56.

Sebastian Riedel, Hong-Woo Chun, Toshihisa Takagi, and Jun'ichi Tsujii. 2009. A markov logic approach to bio-molecular event extraction. In *Proceedings of the Natural Language Processing in Biomedicine NAACL 2009 Workshop (BioNLP '09)*, pages 41–49.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '10)*.

D. Roth and W. Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

Evan Sandhaus, 2008. *The New York Times Annotated Corpus*. Linguistic Data Consortium, Philadelphia.

Stefan Schoenmackers, Oren Etzioni, and Daniel S. Weld. 2008. Scaling textual inference to the web. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–88, Morristown, NJ, USA. Association for Computational Linguistics.

Sameer Singh, Karl Schultz, and Andrew McCallum. 2009. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 414–429.

Daniel S. Weld, Raphael Hoffmann, and Fei Wu. 2009. Using wikipedia to bootstrap open information extraction. In *ACM SIGMOD Record*.

Michael Wick, Khashayar Rohanimanesh, Aron Culotta, and Andrew McCallum. 2009. Samplerank: Learning preferences from atomic gradients. In *Neural Information Processing Systems (NIPS), Workshop on Advances in Ranking*.

Dimitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *JMLR*, 3(6):1083 – 1106.