Distant supervision for relation extraction without labeled data

Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky
Stanford University / Stanford, CA 94305
{mikemintz,sbills,rion,jurafsky}@cs.stanford.edu

Abstract

Modern models of relation extraction for tasks like ACE are based on supervised learning of relations from small hand-labeled corpora. We investigate an alternative paradigm that does not require labeled corpora, avoiding the domain dependence of ACE-style algorithms, and allowing the use of corpora of any size. Our experiments use Freebase, a large semantic database of several thousand relations, to provide *distant supervision*. For each pair of entities that appears in some Freebase relation, we find all sentences containing those entities in a large unlabeled corpus and extract textual features to train a relation classifier. Our algorithm combines the advantages of supervised IE (combining 400,000 noisy pattern features in a probabilistic classifier) and unsupervised IE (extracting large numbers of relations from large corpora of any domain). Our model is able to extract 10,000 instances of 102 relations at a precision of 67.6%. We also analyze feature performance, showing that syntactic parse features are particularly helpful for relations that are ambiguous or lexically distant in their expression.

1 Introduction

At least three learning paradigms have been applied to the task of extracting relational facts from text (for example, learning that a person is employed by a particular organization, or that a geographic entity is located in a particular region).

In supervised approaches, sentences in a corpus are first hand-labeled for the presence of entities and the relations between them. The NIST Automatic Content Extraction (ACE) RDC 2003 and 2004 corpora, for example, include over 1,000 documents in which pairs of entities have been labeled with 5 to 7 major relation types and 23 to 24 subrelations, totaling 16,771 relation instances. ACE systems then extract a wide variety of lexical, syntactic, and semantic features, and use supervised classifiers to label the *relation mention* holding between a given pair of entities in a test set sentence, optionally combining relation mentions (Zhou et al., 2005; Zhou et al., 2007; Surdeanu and Ciaramita, 2007).

Supervised relation extraction suffers from a number of problems, however. Labeled training data is expensive to produce and thus limited in quantity. Also, because the relations are labeled on a particular corpus, the resulting classifiers tend to be biased toward that text domain.

An alternative approach, purely unsupervised information extraction, extracts strings of words between entities in large amounts of text, and clusters and simplifies these word strings to produce relation-strings (Shinyama and Sekine, 2006; Banko et al., 2007). Unsupervised approaches can use very large amounts of data and extract very large numbers of relations, but the resulting relations may not be easy to map to relations needed for a particular knowledge base.

A third approach has been to use a very small number of seed instances or patterns to do bootstrap learning (Brin, 1998; Riloff and Jones, 1999; Agichtein and Gravano, 2000; Ravichandran and Hovy, 2002; Etzioni et al., 2005; Pennacchiotti and Pantel, 2006; Bunescu and Mooney, 2007; Rozenfeld and Feldman, 2008). These seeds are used with a large corpus to extract a new set of patterns, which are used to extract more instances, which are used to extract more patterns, in an iterative fashion. The resulting patterns often suffer from low precision and semantic drift.

We propose an alternative paradigm, *distant supervision*, that combines some of the advantages of each of these approaches. Distant supervision is an extension of the paradigm used by Snow et al. (2005) for exploiting WordNet to extract *hypernym* (is-a) relations between entities, and is similar to the use of *weakly labeled data* in bioinformatics (Craven and Kumlien, 1999; Morgan et al.,

Relation name	New instance
/location/location/contains	Paris, Montmartre
/location/location/contains	Ontario, Fort Erie
/music/artist/origin	Mighty Wagon, Cincinnati
/people/deceased_person/place_of_death	Fyodor Kamensky, Clearwater
/people/person/nationality	Marianne Yvonne Heemskerk, Netherlands
/people/person/place_of_birth	Wavell Wayne Hinds, Kingston
/book/author/works_written	Upton Sinclair, Lanny Budd
/business/company/founders	WWE, Vince McMahon
/people/person/profession	Thomas Mellon, judge

Table 1: Ten relation instances extracted by our system that did not appear in Freebase.

2004). Our algorithm uses Freebase (Bollacker et al., 2008), a large semantic database, to provide distant supervision for relation extraction. Freebase contains 116 million instances of 7,300 relations between 9 million entities. The intuition of distant supervision is that any sentence that contains a pair of entities that participate in a known Freebase relation is likely to express that relation in some way. Since there may be many sentences containing a given entity pair, we can extract very large numbers of (potentially noisy) features that are combined in a logistic regression classifier.

Thus whereas the supervised training paradigm uses a small labeled corpus of only 17,000 relation instances as training data, our algorithm can use much larger amounts of data: more text, more relations, and more instances. We use 1.2 million Wikipedia articles and 1.8 million instances of 102 relations connecting 940,000 entities. In addition, combining vast numbers of features in a large classifier helps obviate problems with bad features.

Because our algorithm is supervised by a database, rather than by labeled text, it does not suffer from the problems of overfitting and domain-dependence that plague supervised systems. Supervision by a database also means that, unlike in unsupervised approaches, the output of our classifier uses canonical names for relations.

Our paradigm offers a natural way of integrating data from multiple sentences to decide if a relation holds between two entities. Because our algorithm can use large amounts of unlabeled data, a pair of entities may occur multiple times in the test set. For each pair of entities, we aggregate the features from the many different sentences in which that pair appeared into a single feature vector, allowing us to provide our classifier with more information, resulting in more accurate labels.

Table 1 shows examples of relation instances extracted by our system. We also use this system to investigate the value of syntactic versus lexical (word sequence) features in relation extraction. While syntactic features are known to improve the performance of supervised IE, at least using clean hand-labeled ACE data (Zhou et al., 2007; Zhou et al., 2005), we do not know whether syntactic features can improve the performance of unsupervised or distantly supervised IE. Most previous research in bootstrapping or unsupervised IE has used only simple lexical features, thereby avoiding the computational expense of parsing (Brin, 1998; Agichtein and Gravano, 2000; Etzioni et al., 2005), and the few systems that have used unsupervised IE have not compared the performance of these two types of feature.

2 Previous work

Except for the unsupervised algorithms discussed above, previous supervised or bootstrapping approaches to relation extraction have typically relied on relatively small datasets, or on only a small number of distinct relations. Approaches based on WordNet have often only looked at the hypernym (is-a) or meronym (part-of) relation (Girju et al., 2003; Snow et al., 2005), while those based on the ACE program (Doddington et al., 2004) have been restricted in their evaluation to a small number of relation instances and corpora of less than a million words.

Many early algorithms for relation extraction used little or no syntactic information. For example, the DIPRE algorithm by Brin (1998) used string-based regular expressions in order to recognize relations such as *author-book*, while the SNOWBALL algorithm by Agichtein and Gravano (2000) learned similar regular expression patterns over words and named entity tags. Hearst (1992) used a small number of regular expressions over words and part-of-speech tags to find examples of the hypernym relation. The use of these patterns has been widely replicated in successful systems, for example by Etzioni et al. (2005). Other work

Relation name	Size	Example
/people/person/nationality	281,107	John Dugard, South Africa
/location/location/contains	253,223	Belgium, Nijlen
/people/person/profession	208,888	Dusa McDuff, Mathematician
/people/person/place_of_birth	105,799	Edwin Hubble, Marshfield
/dining/restaurant/cuisine	86,213	MacAyo's Mexican Kitchen, Mexican
/business/business_chain/location	66,529	Apple Inc., Apple Inc., South Park, NC
/biology/organism_classification_rank	42,806	Scorpaeniformes, Order
/film/film/genre	40,658	Where the Sidewalk Ends, Film noir
/film/film/language	31,103	Enter the Phoenix, Cantonese
/biology/organism_higher_classification	30,052	Calopteryx, Calopterygidae
/film/film/country	27,217	Turtle Diary, United States
/film/writer/film	23,856	Irving Shulman, Rebel Without a Cause
/film/director/film	23,539	Michael Mann, Collateral
/film/producer/film	22,079	Diane Eskenazi, Aladdin
/people/deceased_person/place_of_death	18,814	John W. Kern, Asheville
/music/artist/origin	18,619	The Octopus Project, Austin
/people/person/religion	17,582	Joseph Chartrand, Catholicism
/book/author/works_written	17,278	Paul Auster, Travels in the Scriptorium
/soccer/football_position/players	17,244	Midfielder, Chen Tao
/people/deceased_person/cause_of_death	16,709	Richard Daintree, Tuberculosis
/book/book/genre	16,431	Pony Soldiers, Science fiction
/film/film/music	14,070	Stavisky, Stephen Sondheim
/business/company/industry	13,805	ATS Medical, Health care

Table 2: The 23 largest Freebase relations we use, with their size and an instance of each relation.

such as Ravichandran and Hovy (2002) and Pantel and Pennacchiotti (2006) use the same formalism of learning regular expressions over words and part-of-speech tags to discover patterns indicating a variety of relations.

More recent approaches have used deeper syntactic information derived from parses of the input sentences, including work exploiting syntactic dependencies by Lin and Pantel (2001) and Snow et al. (2005), and work in the ACE paradigm such as Zhou et al. (2005) and Zhou et al. (2007).

Perhaps most similar to our distant supervision algorithm is the effective method of Wu and Weld (2007) who extract relations from a Wikipedia page by using supervision from the page's *infobox*. Unlike their corpus-specific method, which is specific to a (single) Wikipedia page, our algorithm allows us to extract evidence for a relation from many different documents, and from any genre.

3 Freebase

Following the literature, we use the term 'relation' to refer to an ordered, binary relation between entities. We refer to individual ordered pairs in this relation as 'relation instances'. For example, the *person-nationality* relation holds between the entities named 'John Steinbeck' and 'United States', so it has ⟨`John Steinbeck`, `United States`⟩ as an instance.

We use relations and relation instances from Freebase, a freely available online database of structured semantic data. Data in Freebase is collected from a variety of sources. One major source is text boxes and other tabular data from Wikipedia. Data is also taken from NNDB (biographical information), MusicBrainz (music), the SEC (financial and corporate data), as well as direct, wiki-style user editing. After some basic processing of the July 2008 link export to convert Freebase's data representation into binary relations, we have 116 million instances of 7,300 relations between 9 million entities. We next filter out nameless and uninteresting entities such as user profiles and music tracks. Freebase also contains the reverses of many of its relations (book-author v. author-book), and these are merged. Filtering and removing all but the largest relations leaves us with 1.8 million instances of 102 relations connecting 940,000 entities. Examples are shown in Table 2.

4 Architecture

The intuition of our *distant supervision* approach is to use Freebase to give us a training set of relations and entity pairs that participate in those relations. In the training step, all entities are identified in sentences using a named entity tagger that labels persons, organizations and locations. If a sentence contains two entities and those entities are an instance of one of our Freebase relations, features are extracted from that sentence and are added to the feature vector for the relation.

The distant supervision assumption is that if two entities participate in a relation, any sentence that contain those two entities might express that relation. Because any individual sentence may give an incorrect cue, our algorithm trains a multiclass logistic regression classifier, learning weights for each noisy feature. In training, the features for identical tuples (relation, entity1, entity2) from different sentences are combined, creating a richer feature vector.

In the testing step, entities are again identified using the named entity tagger. This time, every pair of entities appearing together in a sentence is considered a potential relation instance, and whenever those entities appear together, features are extracted on the sentence and added to a feature vector for that entity pair. For example, if a pair of entities occurs in 10 sentences in the test set, and each sentence has 3 features extracted from it, the entity pair will have 30 associated features. Each entity pair in each sentence in the test corpus is run through feature extraction, and the regression classifier predicts a relation name for each entity pair based on the features from all of the sentences in which it appeared.

Consider the *location-contains* relation, imagining that in Freebase we had two instances of this relation: ⟨`Virginia, Richmond`⟩ and ⟨`France, Nantes`⟩. As we encountered sentences like 'Richmond, the capital of Virginia' and 'Henry's Edict of Nantes helped the Protestants of France' we would extract features from these sentences. Some features would be very useful, such as the features from the Richmond sentence, and some would be less useful, like those from the Nantes sentence. In testing, if we came across a sentence like 'Vienna, the capital of Austria', one or more of its features would match those of the Richmond sentence, providing evidence that ⟨`Austria, Vienna`⟩ belongs to the *location-contains* relation.

Note that one of the main advantages of our architecture is its ability to combine information from many different mentions of the same relation. Consider the entity pair ⟨`Steven Spielberg, Saving Private Ryan`⟩ from the following two sentences, as evidence for the *film-director* relation.

> *[Steven Spielberg]'s film [Saving Private Ryan] is loosely based on the brothers' story.*
>
> *Allison co-produced the Academy Award-winning [Saving Private Ryan], directed by [Steven Spielberg]...*

The first sentence, while providing evidence for *film-director*, could instead be evidence for *film-writer* or *film-producer*. The second sentence does not mention that Saving Private Ryan is a film, and so could instead be evidence for the *CEO* relation (consider 'Robert Mueller directed the FBI'). In isolation, neither of these features is conclusive, but in combination, they are.

5 Features

Our features are based on standard lexical and syntactic features from the literature. Each feature describes how two entities are related in a sentence, using either syntactic or non-syntactic information.

5.1 Lexical features

Our lexical features describe specific words between and surrounding the two entities in the sentence in which they appear:

- The sequence of words between the two entities
- The part-of-speech tags of these words
- A flag indicating which entity came first in the sentence
- A window of k words to the left of Entity 1 and their part-of-speech tags
- A window of k words to the right of Entity 2 and their part-of-speech tags

Each lexical feature consists of the conjunction of all these components. We generate a conjunctive feature for each $k \in \{0, 1, 2\}$. Thus each *lexical* row in Table 3 represents a single lexical feature.

Part-of-speech tags were assigned by a maximum entropy tagger trained on the Penn Treebank, and then simplified into seven categories: nouns, verbs, adverbs, adjectives, numbers, foreign words, and everything else.

In an attempt to approximate syntactic features, we also tested variations on our lexical features: (1) omitting all words that are not verbs and (2) omitting all function words. In combination with the other lexical features, they gave a small boost to precision, but not large enough to justify the increased demand on our computational resources.

Feature type	Left window	NE1	Middle	NE2	Right window
Lexical	[]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[]
Lexical	[Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[,]
Lexical	[#PAD#, Astronomer]	PER	[was/VERB born/VERB in/CLOSED]	LOC	[, Missouri]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{lex-mod}$,]
Syntactic	[]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]
Syntactic	[Edwin Hubble $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]
Syntactic	[Astronomer $\Downarrow_{lex-mod}$]	PER	[$\Uparrow_s$ was $\Downarrow_{pred}$ born $\Downarrow_{mod}$ in $\Downarrow_{pcomp-n}$]	LOC	[$\Downarrow_{inside}$ Missouri]

Table 3: Features for 'Astronomer Edwin Hubble was born in Marshfield, Missouri'.

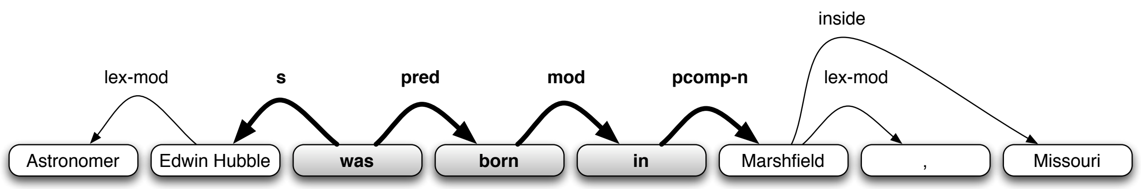

Figure 1: Dependency parse with dependency path from 'Edwin Hubble' to 'Marshfield' highlighted in boldface.

5.2 Syntactic features

In addition to lexical features we extract a number of features based on syntax. In order to generate these features we parse each sentence with the broad-coverage dependency parser MINIPAR (Lin, 1998).

A dependency parse consists of a set of words and chunks (e.g. 'Edwin Hubble', 'Missouri', 'born'), linked by directional dependencies (e.g. 'pred', 'lex-mod'), as in Figure 1. For each sentence we extract a dependency path between each pair of entities. A dependency path consists of a series of dependencies, directions and words/chunks representing a traversal of the parse. Part-of-speech tags are not included in the dependency path.

Our syntactic features are similar to those used in Snow et al. (2005). They consist of the conjunction of:

- A dependency path between the two entities
- For each entity, one 'window' node that is not part of the dependency path

A window node is a node connected to one of the two entities and not part of the dependency path. We generate one conjunctive feature for each pair of left and right window nodes, as well as features which omit one or both of them. Thus each *syntactic* row in Table 3 represents a single syntactic feature.

5.3 Named entity tag features

Every feature contains, in addition to the content described above, named entity tags for the two entities. We perform named entity tagging using the Stanford four-class named entity tagger (Finkel et al., 2005). The tagger provides each word with a label from {person, location, organization, miscellaneous, none}.

5.4 Feature conjunction

Rather than use each of the above features in the classifier independently, we use only conjunctive features. Each feature consists of the conjunction of several attributes of the sentence, plus the named entity tags. For two features to match, all of their conjuncts must match exactly. This yields low-recall but high-precision features. With a small amount of data, this approach would be problematic, since most features would only be seen once, rendering them useless to the classifier. Since we use large amounts of data, even complex features appear multiple times, allowing our high-precision features to work as intended. Features for a sample sentence are shown in Table 3.

6 Implementation

6.1 Text

For unstructured text we use the Freebase Wikipedia Extraction, a dump of the full text of all Wikipedia articles (not including discussion and

Relation	Feature type	Left window	NE1	Middle	NE2	Right window
/architecture/structure/architect	LEX$\curvearrowleft$		ORG	, the designer of the	PER	
	SYN	designed $\Uparrow_s$	ORG	$\Uparrow_s$ designed $\Downarrow_{by-subj}$ by $\Downarrow_{pcn}$	PER	$\Uparrow_s$ designed
/book/author/works_written	LEX		PER	s novel	ORG	
	SYN		PER	$\Uparrow_{pcn}$ by $\Uparrow_{mod}$ story $\Uparrow_{pred}$ is $\Downarrow_s$	ORG	
/book/book_edition/author_editor	LEX$\curvearrowleft$		ORG	s novel	PER	
	SYN		PER	$\Uparrow_{nn}$ series $\Downarrow_{gen}$	PER	
/business/company/founders	LEX		ORG	co - founder	PER	
	SYN		ORG	$\Uparrow_{nn}$ owner $\Downarrow_{person}$	PER	
/business/company/place_founded	LEX$\curvearrowleft$		ORG	- based	LOC	
	SYN		ORG	$\Uparrow_s$ founded $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	
/film/film/country	LEX		PER	, released in	LOC	
	SYN	opened $\Uparrow_s$	ORG	$\Uparrow_s$ opened $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ opened
/geography/river/mouth	LEX		LOC	, which flows into the	LOC	
	SYN	the $\Downarrow_{det}$	LOC	$\Uparrow_s$ is $\Downarrow_{pred}$ tributary $\Downarrow_{mod}$ of $\Downarrow_{pcn}$	LOC	$\Downarrow_{det}$ the
/government/political_party/country	LEX$\curvearrowleft$		ORG	politician of the	LOC	
	SYN	candidate $\Uparrow_{nn}$	ORG	$\Uparrow_{nn}$ candidate $\Downarrow_{mod}$ for $\Downarrow_{pcn}$	LOC	$\Uparrow_{nn}$ candidate
/influence/influence_node/influenced	LEX$\curvearrowleft$		PER	, a student of	PER	
	SYN	of $\Uparrow_{pcn}$	PER	$\Uparrow_{pcn}$ of $\Uparrow_{mod}$ student $\Uparrow_{appo}$	PER	$\Uparrow_{pcn}$ of
/language/human_language/region	LEX		LOC	- speaking areas of	LOC	
	SYN		LOC	$\Uparrow_{lex-mod}$ speaking areas $\Downarrow_{mod}$ of $\Downarrow_{pcn}$	LOC	
/music/artist/origin	LEX$\curvearrowleft$		ORG	based band	LOC	
	SYN	is $\Uparrow_s$	ORG	$\Uparrow_s$ is $\Downarrow_{pred}$ band $\Downarrow_{mod}$ from $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ is
/people/deceased_person/place_of_death	LEX		PER	died in	LOC	
	SYN	hanged $\Uparrow_s$	PER	$\Uparrow_s$ hanged $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ hanged
/people/person/nationality	LEX		PER	is a citizen of	LOC	
	SYN		PER	$\Downarrow_{mod}$ from $\Downarrow_{pcn}$	LOC	
/people/person/parents	LEX		PER	, son of	PER	
	SYN	father $\Uparrow_{gen}$	PER	$\Uparrow_{gen}$ father $\Downarrow_{person}$	PER	$\Uparrow_{gen}$ father
/people/person/place_of_birth	LEX$\curvearrowleft$		PER	is the birthplace of	PER	
	SYN		PER	$\Uparrow_s$ born $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	
/people/person/religion	LEX		PER	embraced	LOC	
	SYN	convert $\Downarrow_{appo}$	PER	$\Downarrow_{appo}$ convert $\Downarrow_{mod}$ to $\Downarrow_{pcn}$	LOC	$\Downarrow_{appo}$ convert

Table 4: Examples of high-weight features for several relations. Key: SYN = syntactic feature; LEX = lexical feature; $\curvearrowleft$ = reversed; NE# = named entity tag of entity.

user pages) which has been sentence-tokenized by Metaweb Technologies, the developers of Freebase (Metaweb, 2008). This dump consists of approximately 1.8 million articles, with an average of 14.3 sentences per article. The total number of words (counting punctuation marks) is 601,600,703. For our experiments we use about half of the articles: 800,000 for training and 400,000 for testing.

We use Wikipedia because it is relatively up-to-date, and because its sentences tend to make explicit many facts that might be omitted in newswire. Much of the information in Freebase is derived from tabular data from Wikipedia, meaning that Freebase relations are more likely to appear in sentences in Wikipedia.

6.2 Parsing and chunking

Each sentence of this unstructured text is dependency parsed by MINIPAR to produce a dependency graph.

In preprocessing, consecutive words with the same named entity tag are 'chunked', so that `Edwin/PERSON Hubble/PERSON` becomes `[Edwin Hubble]/PERSON`. This chunking is restricted by the dependency parse of the sentence, however, in that chunks must be contiguous in the parse (i.e., no chunks across subtrees). This ensures that parse tree structure is preserved, since the parses must be updated to reflect the chunking.

6.3 Training and testing

For held-out evaluation experiments (see section 7.1), half of the instances of each relation are not used in training, and are later used to compare against newly discovered instances. This means that 900,000 Freebase relation instances are used in training, and 900,000 are held out. These experiments used 800,000 Wikipedia articles in the training phase and 400,000 different articles in the testing phase.

For human evaluation experiments, all 1.8 million relation instances are used in training. Again, we use 800,000 Wikipedia articles in the training phase and 400,000 different articles in the testing phase.

For all our experiments, we only extract relation instances that do not appear in our training data, i.e., instances that are not already in Freebase.

Our system needs negative training data for the purposes of constructing the classifier. Towards this end, we build a feature vector in the training phase for an 'unrelated' relation by randomly selecting entity pairs that do not appear in any Freebase relation and extracting features for them. While it is possible that some of these entity pairs

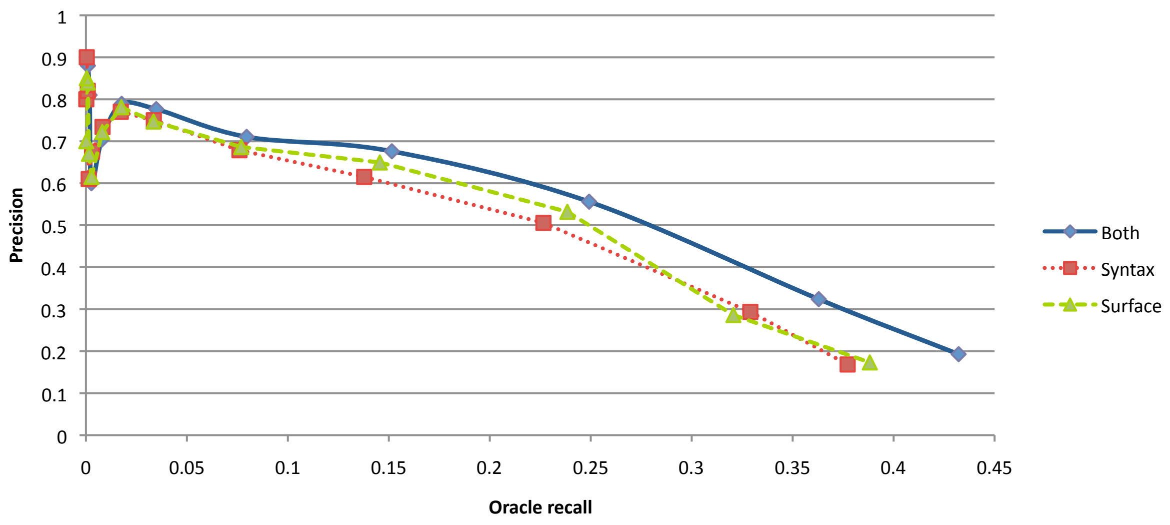

Figure 2: Automatic evaluation with 50% of Freebase relation data held out and 50% used in training on the 102 largest relations we use. Precision for three different feature sets (lexical features, syntactic features, and both) is reported at recall levels from 10 to 100,000. At the 100,000 recall level, we classify most of the instances into three relations: 60% as *location-contains*, 13% as *person-place-of-birth*, and 10% as *person-nationality*.

are in fact related but are wrongly omitted from the Freebase data, we expect that on average these false negatives will have a small effect on the performance of the classifier. For performance reasons, we randomly sample 1% of such entity pairs for use as negative training examples. By contrast, in the actual test data, 98.7% of the entity pairs we extract do not possess any of the top 102 relations we consider in Freebase.

We use a multi-class logistic classifier optimized using L-BFGS with Gaussian regularization. Our classifier takes as input an entity pair and a feature vector, and returns a relation name and a confidence score based on the probability of the entity pair belonging to that relation. Once all of the entity pairs discovered during testing have been classified, they can be ranked by confidence score and used to generate a list of the n most likely new relation instances.

Table 4 shows some high-weight features learned by our system. We discuss the results in the next section.

7 Evaluation

We evaluate labels in two ways: automatically, by holding out part of the Freebase relation data during training, and comparing newly discovered relation instances against this held-out data, and manually, having humans who look at each positively labeled entity pair and mark whether the relation indeed holds between the participants. Both evaluations allow us to calculate the precision of the system for the best N instances.

7.1 Held-out evaluation

Figure 2 shows the performance of our classifier on held-out Freebase relation data. While held-out evaluation suffers from false negatives, it gives a rough measure of precision without requiring expensive human evaluation, making it useful for parameter setting.

At most recall levels, the combination of syntactic and lexical features offers a substantial improvement in precision over either of these feature sets on its own.

7.2 Human evaluation

Human evaluation was performed by evaluators on Amazon's Mechanical Turk service, shown to be effective for natural language annotation in Snow et al. (2008). We ran three experiments: one using only syntactic features; one using only lexical features; and one using both syntactic and lexical features. For each of the 10 relations that appeared most frequently in our test data (according to our classifier), we took samples from the first 100 and 1000 instances of this relation generated in each experiment, and sent these to Mechanical Turk for

Relation name	100 instances			1000 instances		
	Syn	Lex	Both	Syn	Lex	Both
/film/director/film	**0.49**	0.43	0.44	**0.49**	0.41	0.46
/film/writer/film	**0.70**	0.60	0.65	**0.71**	0.61	0.69
/geography/river/basin_countries	0.65	0.64	**0.67**	**0.73**	0.71	0.64
/location/country/administrative_divisions	0.68	0.59	**0.70**	**0.72**	0.68	**0.72**
/location/location/contains	0.81	**0.89**	0.84	**0.85**	0.83	0.84
/location/us_county/county_seat	0.51	0.51	**0.53**	0.47	**0.57**	0.42
/music/artist/origin	0.64	0.66	**0.71**	0.61	**0.63**	0.60
/people/deceased_person/place_of_death	0.80	0.79	**0.81**	0.80	**0.81**	0.78
/people/person/nationality	0.61	0.70	**0.72**	0.56	0.61	**0.63**
/people/person/place_of_birth	**0.78**	0.77	**0.78**	0.88	0.85	**0.91**
Average	0.67	0.66	**0.69**	**0.68**	0.67	0.67

Table 5: Estimated precision on human-evaluation experiments of the highest-ranked 100 and 1000 results per relation, using stratified samples. 'Average' gives the mean precision of the 10 relations. Key: Syn = syntactic features only. Lex = lexical features only. We use stratified samples because of the overabundance of *location-contains* instances among our high-confidence results.

human evaluation. Our sample size was 100.

Each predicted relation instance was labeled as true or false by between 1 and 3 labelers on Mechanical Turk. We assigned the truth or falsehood of each relation according to the majority vote of the labels; in the case of a tie (one vote each way) we assigned the relation as true or false with equal probability. The evaluation of the syntactic, lexical, and combination of features at a recall of 100 and 1000 instances is presented in Table 5.

At a recall of 100 instances, the combination of lexical and syntactic features has the best performance for a majority of the relations, while at a recall level of 1000 instances the results are mixed. No feature set strongly outperforms any of the others across all relations.

8 Discussion

Our results show that the distant supervision algorithm is able to extract high-precision patterns for a reasonably large number of relations.

The held-out results in Figure 2 suggest that the combination of syntactic and lexical features provides better performance than either feature set on its own. In order to understand the role of syntactic features, we examine Table 5, the human evaluation of the most frequent 10 relations. For the top-ranking 100 instances of each relation, most of the best results use syntactic features, either alone or in combination with lexical features. For the top-ranking 1000 instances of each relation, the results are more mixed, but syntactic features still helped in most classifications.

We then examine those relations for which syntactic features seem to help. For example, syntactic features consistently outperform lexical features for the *director-film* and *writer-film* relations. As discussed in section 4, these two relations are particularly ambiguous, suggesting that syntactic features may help tease apart difficult relations. Perhaps more telling, we noticed many examples with a long string of words between the director and the film:

> *Back Street is a 1932 film made by Universal Pictures, directed by John M. Stahl, and produced by Carl Laemmle Jr.*

Sentences like this have very long (and thus rare) lexical features, but relatively short dependency paths. Syntactic features can more easily abstract from the syntactic modifiers that comprise the extraneous parts of these strings.

Our results thus suggest that syntactic features are indeed useful in distantly supervised information extraction, and that the benefit of syntax occurs in cases where the individual patterns are particularly ambiguous, and where they are nearby in the dependency structure but distant in terms of words. It remains for future work to see whether simpler, chunk-based syntactic features might be able to capture enough of this gain without the overhead of full parsing, and whether coreference resolution could improve performance.

Acknowledgments

We would like to acknowledge Sarah Spikes for her help in developing the relation extraction system, Christopher Manning and Mihai Surdeanu for their invaluable advice, and Fuliang Weng and Baoshi Yan for their guidance. Our research was partially funded by the NSF via award IIS-0811974 and by Robert Bosch LLC.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In Manuela M Veloso, editor, *IJCAI-07*, pages 2670–2676.

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*, pages 1247–1250, New York, NY. ACM.

Sergei Brin. 1998. Extracting patterns and relations from the World Wide Web. In *Proceedings World Wide Web and Databases International Workshop, Number 1590 in LNCS*, pages 172–183. Springer.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *ACL-07*, pages 576–583, Prague, Czech Republic, June.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In Thomas Lengauer, Reinhard Schneider, Peer Bork, Douglas L. Brutlag, Janice I. Glasgow, Hans W. Mewes, and Ralf Zimmer, editors, *ISMB*, pages 77–86. AAAI.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. *LREC-04*, pages 837–840.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

Jenny R. Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL-05*, pages 363–370, Ann Arbor, MI.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *HLT-NAACL-03*, pages 1–8, Edmonton, Canada.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING-92*, Nantes, France.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Workshop on the Evaluation of Parsing Systems*.

Metaweb. 2008. Freebase data dumps. `http://download.freebase.com/datadumps/`.

Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, and Jeff B. Colombe. 2004. Gene name identification and normalization using a model organism database. *J. of Biomedical Informatics*, 37(6):396–410.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In *COLING/ACL 2006*, pages 113–120, Sydney, Australia.

Marco Pennacchiotti and Patrick Pantel. 2006. A bootstrapping algorithm for automatically harvesting semantic relations. In *in Proceedings of Inference in Computational Semantics (ICoS-06)*, pages 87–96.

Deepak Ravichandran and Eduard H. Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL-02*, pages 41–47, Philadelphia, PA.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI-99*, pages 474–479.

Benjamin Rozenfeld and Ronen Feldman. 2008. Self-supervised relation extraction from the web. *Knowledge and Information Systems*, 17(1):17–33.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *HLT-NAACL-06*, pages 304–311, New York, NY.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *NIPS 17*, pages 1297–1304. MIT Press.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP 2008*, pages 254–263, Honolulu, HI.

Mihai Surdeanu and Massimiliano Ciaramita. 2007. Robust information extraction with perceptrons. In *Proceedings of the NIST 2007 Automatic Content Extraction Workshop (ACE07)*, March.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50, Lisbon, Portugal.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *ACL-05*, pages 427–434, Ann Arbor, MI.

Guodong Zhou, Min Zhang, Donghong Ji, and Qiaoming Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *EMNLP/CoNLL 2007*.