Open Information Extraction

- Traditional *relation extraction* systems learn how to identify instances of a specific relation, usually given labeled examples of that relation.
- In contrast, the goal of *open information extraction (OpenIE)* systems is to extract instances of any relation (i.e., an open set of relations).
  - Essentially, extract everything you can find!
- Open IE systems typically learn from the Web, benefiting from the Web's vast amount of text.
  - need shallow methods, for robustness and speed

Traditional Relation Extraction vs. Open IE

- Since no specific relation is targeted, the relation phrases need to be identified. (As opposed to, say, a sequential tagger for relation extraction that identifies the entities participating in a relation based on contextual features.)
- The relation phrases need to be clustered/normalized to determine which instances represent the same relation.

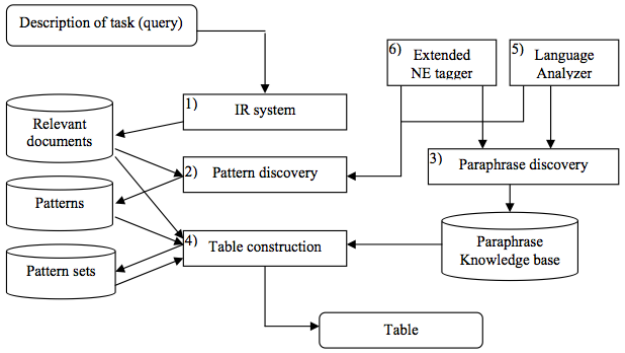
  "*is headquartered in*" = "*is based in*"
- Very general feature sets are needed to:
  1. cover an unlimited & unknown set of relations. Even anchoring on Named Entities will be a problem for many relations.
  2. robustly and efficiently process large amounts of WWW text.

Open IE Research

- Several research efforts focus on developing Open IE systems to automatically acquire knowledge from the Web. This is also sometimes called *Machine Reading*.
- Generally speaking, the goals of Open IE are to create systems that can:
  - robustly process and extract knowledge from massive amounts of Web text.
  - populate and organize the extracted information in large knowledge bases.
  - develop methods that can continually harvest new knowledge, both to acquire new facts that emerge and to enable new types of relations to be identified.

On-Demand Information Extraction

[Sekine 2006] proposed **"on-demand" information extraction**, where a user would provide a query for a desired relation and the system would automatically learn paraphrases and build a table of extracted information.



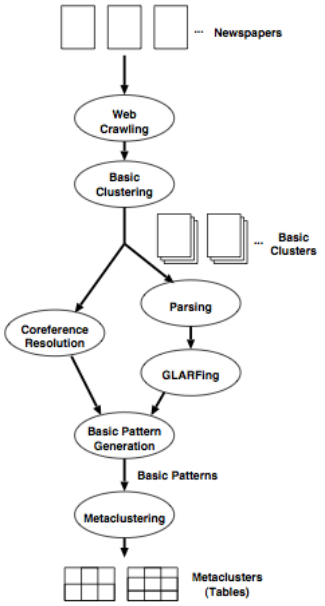
Preemptive Information Extraction

- [Shinyama and Sekine, 2006] explored the idea of **preemptive information extraction** and proposed:

  "a technique called Unrestricted Relation Discovery that discovers all possible relations from texts and presents them as tables."

- Their system used clustering, pattern learning, and meta-clustering to build a set of tables filled with information extracted for different relations, without training data.

- This work was among the earliest Open IE research, and a preliminary system was built. But the effort did not continue on as large of a scale as similar efforts undertaken by other research groups.



Output Results

Source articles	28,009
Basic clusters	5,543
Basic patterns (token)	643,767
Basic patterns (type)	7,990
Metaclusters	302
Metaclusters (rows ≥ 3)	101

Nominations Table

Article	1:confirm	2:be-confirmed
2005-09-21	Senate	Roberts
2005-10-03	Supreme Court	Miers
2005-10-20	Senate	Bush
2005-10-26	Senate	Sauerbrey
2005-10-31	Senate	Mr. Alito
2005-11-04	Senate	Alito
2005-11-17	Fed	Bernanke

Hurricanes Table

Hurricane	Date (Affected Place)	Articles
Philippe	Sep 17-20 (?)	6
* Rita	Sep 17-26 (Louisiana, Texas, etc.)	566
* Stan	Oct 1-5 (Mexico, Nicaragua, etc.)	83
* Tammy	Oct 5-? (Georgia, Alabama)	18
Vince	Oct 8-11 (Portugal, Spain)	12
* Wilma	Oct 15-25 (Cuba, Honduras, etc.)	368
Alpha	Oct 22-24 (Haiti, Dominican Rep.)	80
* Beta	Oct 26-31 (Nicaragua, Honduras)	55
* Gamma	Nov 13-20 (Belize, etc.)	36

Never-Ending Language Learning (NELL)

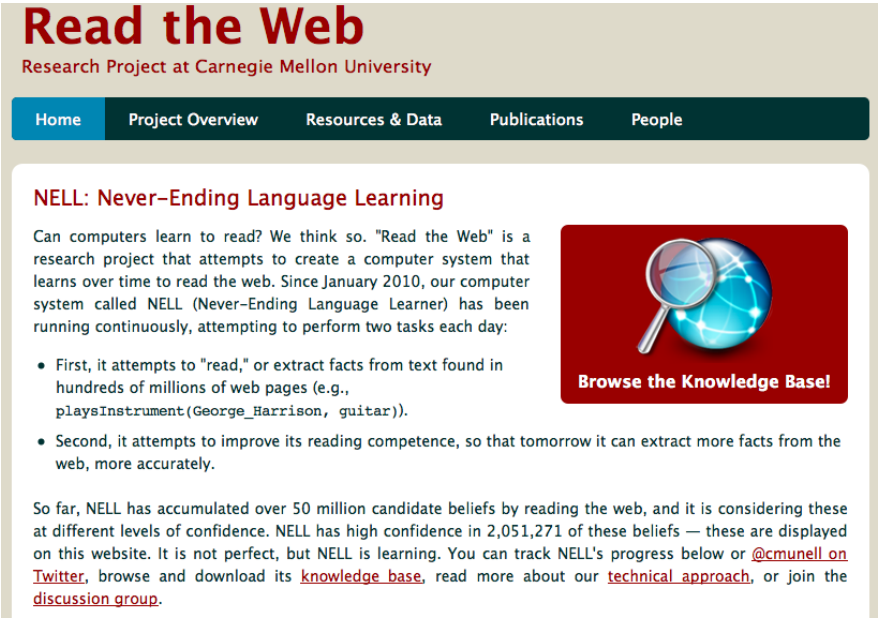
- The NELL effort at Carnegie Mellon University uses semi-supervised learning methods to automatically extract large amounts of knowledge from the Web.

- The NELL project aims to continually acquire knowledge and improve its performance. From [Carlson et al., AAAI 2010]:

  "By a "never- ending language learner" we mean a computer system that runs 24 hours per day, 7 days per week, forever, performing two tasks each day:

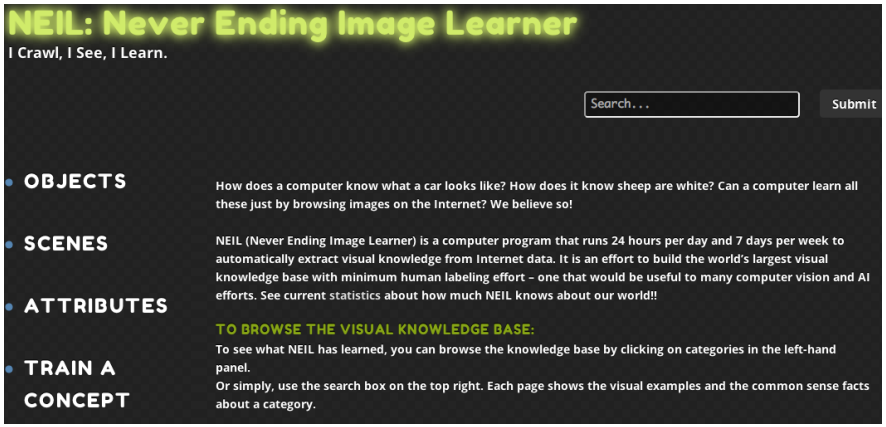
  *1. **Reading task**: extract information from web text to further populate a growing knowledge base of structured facts and knowledge.*

  *2. **Learning task**: learn to read better each day than the day before, as evidenced by its ability to go back to yesterday's text sources and extract more information more accurately."*



Never-Ending Image Learner (NEIL)

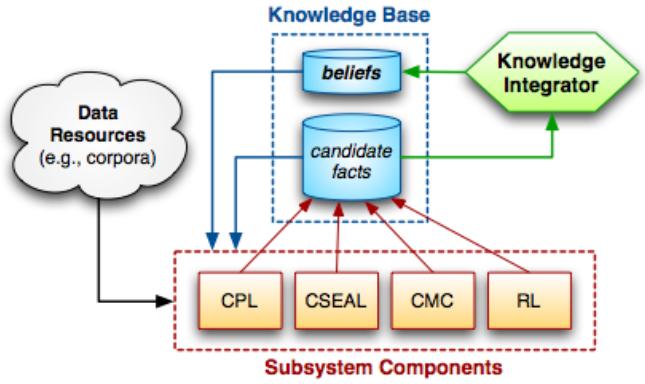
Interesting Aside: NELL also inspired a follow-on effort at CMU called NEIL to continually extract visual knowledge!



Never-Ending Language Learning (NELL)

- NELL consists of an ensemble of extraction methods that can learn:
  - *semantic categories*, such as cities, companies, and teams
  - *relations*, such as HasOfficesIn(Organization, Location)
- NELL includes natural language pattern learners, extractors for semi-structured text (e.g., tables and lists), morphological similarity learners, probabilistic inference rule learning, etc.
- [Carlson et al., 2010] presented a method that trains extractors category and relation extractors using small amounts of labeled data, and applies them to the Web. "Coupling constraints" are defined across extractors to improve accuracy.

NELL Architecture



Examples of Different Types of Learning

CPL: Semantic Class Learning with Contextual Patterns

Predicate	Pattern
emotion	hearts full of X
beverage	cup of aromatic X
newspaper	op-ed page of X
teamPlaysInLeague	X ranks second in Y
bookAuthor	Y classic X

CSEAL: Web Page Wrapper Induction

Predicate	Web URL	Extraction Template
academicField	http://scholendow.ais.msu.edu/student/ScholSearch.Asp	` [X] -`
athlete	http://www.quotes-search.com/d_occupation.aspx?o=+athlete	`<a href='d_author.aspx?a=[X]'>-`
bird	http://www.michaelforsberg.com/stock.html	`<option>[X]</option>`
bookAuthor	http://lifebehindthecurve.com/	`</li> <li>[X] by [Y] &#8211;`

RL: Horn Clauses induced by Rule Learner

Probability	Consequent	Antecedents
0.95	athletePlaysSport(X, basketball)	⇐ athleteInLeague(X, NBA)
0.91	teamPlaysInLeague(X, NHL)	⇐ teamWonTrophy(X, Stanley Cup)
0.90	athleteInLeague(X, Y)	⇐ athletePlaysForTeam(X, Z), teamPlaysInLeague(Z, Y)
0.88	cityInState(X, Y)	⇐ cityCapitalOfState(X, Y), cityInCountry(X, USA)
† 0.62	newspaperInCity(X, New York)	⇐ companyEconomicSector(X, media), generalizations(X, blog)

Examples of Learned Knowledge

Predicate	Instance	Source(s)
ethnicGroup	Cubans	CSEAL
arthropod	spruce beetles	CPL, CSEAL
female	Kate Mara	CPL, CMC
sport	BMX bicycling	CSEAL, CMC
profession	legal assistants	CPL
magazine	Thrasher	CPL
bird	Buff-throated Warbler	CSEAL
river	Fording River	CPL, CMC
mediaType	chemistry books	CPL, CMC
cityInState	(troy, Michigan)	CSEAL
musicArtistGenre	(Nirvana, Grunge)	CPL
tvStationInCity	(WLS-TV, Chicago)	CPL, CSEAL
sportUsesEquip	(soccer, balls)	CPL
athleteInLeague	(Dan Fouts, NFL)	RL
starredIn	(Will Smith, Seven Pounds)	CPL
productType	(Acrobat Reader, FILE)	CPL
athletePlaysSport	(scott shields, baseball)	RL
cityInCountry	(Dublin Airport, Ireland)	CPL

Table 1: Example beliefs promoted by NELL.

KnowItAll

- A research group at the University of Washington began an OpenIE research project called KnowItAll, which has produced a steady stream of research results related to open information extraction.
- The emphasis of this project has been massive Web-scale IE, with an emphasis on speed and extracting large volumes of information.
- Consequently, many of the methods use very shallow pattern matching and few NLP tools.
- The original KnowItAll system used Hearst's hyponym patterns to identify relation instances in an iterative learning framework.

KnowItAll Rule Examples [Etzioni et al., 2004]

NP1 {“,”} “such as” NPList2
NP1 {“,”}“and other” NP2
NP1 {“,”}“including” NPList2
NP1 “is a” NP2
NP1 “is the” NP2 “of” NP3
“the” NP1 “of” NP2 “is” NP3

```
Extraction Rule:

  NP1 "such as" NPList2
  & head(NP1)="countries"
  & properNoun(head(each(NPList2)))
  =>
  instanceOf(Country,head(each(NPList2)))
  keywords: "countries such as"
```

```
Extraction Rule for a Binary Relation:

  NP1 "plays for" NP2
  & properNoun(head(NP1))
  & head(NP2)="Seattle Mariners"
  =>
  instanceOf(Athlete,head(NP1))
  & instanceOf(SportsTeam,head(NP2))
  & playsFor(head(NP1),head(NP2))
  keywords: "plays for", "Seattle Mariners"
```

Examples of KnowItAll Research Efforts

- **ReVerb**: identifies and extracts unspecified binary relations.
- **RESOLVER**: a probabilistic relational model for determining whether two relation expressions are “synonymous” (paraphrases).
- **TextRunner** generates labeled examples using heuristics and trains a classifier for unrestricted relation extraction. RESOLVER is incorporated to identify synonymous relation phrases.
- **SHERLOCK**: learns first-order Horn Clauses as inference rules. For example:

  Contains(Food, Chemical) :- IsMadeFrom(Food, Ingredient) ∧ Contains (Ingredient, Chemical);

Common Syntactic Patterns

- 500 randomly sampled sentences were reviewed to manually identify the types of constructions the captured a relation expression.
- 95% of the identified patterns could be grouped into 8 lexico-syntactic categories.
- While these patterns are not sufficient to identify a relation, these results suggest that most relation expressions can be captured by this small set of patterns.

Common Lexico-syntactic Patterns

95% of the 500 sampled sentences have relation expressions matching one of these patterns.

Relative Frequency	Category	Simplified Lexico-Syntactic Pattern
37.8	Verb	E_1 Verb E_2 *X established Y*
22.8	Noun+Prep	E_1 NP Prep E_2 *X settlement with Y*
16.0	Verb+Prep	E_1 Verb Prep E_2 *X moved to Y*
9.4	Infinitive	E_1 to Verb E_2 *X plans to acquire Y*
5.2	Modifier	E_1 Verb E_2 Noun *X is Y winner*
1.8	$Coordinate_n$	E_1 (and|,|-|:) E_2 NP *X-Y deal*
1.0	$Coordinate_v$	E_1 (and|,) E_2 Verb *X , Y merge*
0.8	Appositive	E_1 NP (:|,)? E_2 *X hometown : Y*

Seed Labeled Data

Relation-independent heuristics are applied to the Penn Treebank to obtain labeled relation instances.

For example:

Class: + **Heuristic**: Subject,Verb,Object (SVO) Triple

Example: "*<Einstein> received <the Nobel Prize>*"

Class: - **Heuristic**: ADVP crossing

Example: "*He studied <Einstein's work> when visiting <Germany>.*"

O-CRF

- Labeled instances for training are generated heuristically.
- A sequential tagging model (CRF) is trained to label tokens that express a binary relation using IOB tags.
- A noun phrase chunker is applied and all NPs pairs within a certain distance from each other are candidates for a relation instance.
- The feature set includes POS tags, regular expressions to detect things like capitalization and punctuation, context words, and conjunctions of features for adjacent positions in a context window of size +/- 6 words.
- Context words are only captured for closed class words and not for open class words! Presumably for improved generality.

Relation Extraction as Sequence Labeling

Each relation must be anchored by two noun phrases, which are called "entities" (ENT).

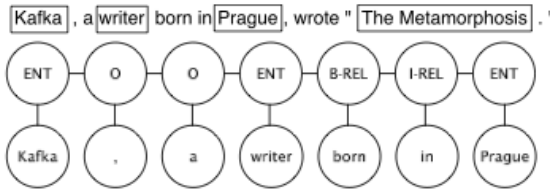


Figure 1: Relation Extraction as Sequence Labeling: A CRF is used to identify the relationship, *born in*, between *Kafka* and *Prague*

O-CRF's Limitations

- Relations can only be identified if they are explicitly mentioned in a text.
- Relations can only be identified through lexical context. Document style features are not considered.
- Relations can only be identified between NPs within the same sentence.
- O-CRF does not cluster/normalize relations.
  - Relation "synonyms" (paraphrases) were identified by a different system called RESOLVER [Yates and Etzioni, 2007].

Relation-Specific Extraction

- For comparison, a traditional relation extraction system was trained with a CRF model, which they called R1-CRF.
- R1-CRF is identical to O-CRF except:
  - R1-CRF was trained from manually labeled positive and negative instances of a specific relation R.
  - R1-CRF used both closed-class and open-class words as features. (O-CRF could only use closed-class words.)
  - No additional steps are needed to identify the relation type, since it is trained to identify only instances of relation R.

Ensembles and Stacking

- **Ensemble methods** are widely used in NLP and often yield better performance than individual systems.
- An **ensemble** is a set of different systems (models) that perform the same task. Ensemble methods consider the output of all the models to make a decision.
- Voting methods decide on a class label based on the set of votes, e.g. by majority vote or the most confident one.
- Stacking methods train a **meta-classifier** that learns how to weight or combine the output values of the individual systems to make better decisions.

A Stacked Relation Extractor (H-CRF)

- A hybrid relation extraction system (H-CRF) is created using a stacking framework.
- The O-CRF and R1-CRF classifiers are the individual components.
- The H-CRF meta-classifier also uses a CRF sequential tagging model for learning.
- The H-CRF's feature set includes:
  - probability estimates for the O-CRF and R-CRF's labels
  - an edit distance measure between the predicted relations
  - a feature indicating whether either model returned No Relation
  - lexical and POS terms between the two candidate NPs

Evaluation of Open Relation Extraction

The first evaluation compares the performance of O-CRF with the TextRunner Open IE system (O-NB). TextRunner had extracted 7.5 million tuples from 9 million Web pages.

Both systems were tested on 500 sentences .

	O-CRF			O-NB		
Category	**P**	**R**	**F1**	**P**	**R**	**F1**
Verb	93.9	65.1	76.9	100	38.6	55.7
Noun+Prep	89.1	36.0	51.3	100	9.7	55.7
Verb+Prep	95.2	50.0	65.6	95.2	25.3	40.0
Infinitive	95.7	46.8	62.9	100	25.5	40.6
Other	0	0	0	0	0	0
All	**88.3**	**45.2**	**59.8**	86.6	23.2	36.6

Comparison for Known Relation Extraction

- The performance of O-CRF and R1-CRF was then compared for specific relations.
- Labeled data was acquired for 4 relations: corporate acquisitions, birthplaces, product inventions, and award winners. The data was divided into training and test sets.
- For each relation, R1-CRF was trained using the labeled training set. Both models were then evaluated on the test set.
- Recall and Precision were measured on the relation tuples that were generated by each system.

Evaluation Results for Known Relations

	O-CRF		R1-CRF		
Relation	**P**	**R**	**P**	**R**	**Train Ex**
Acquisition	75.6	19.5	67.6	69.2	3042
Birthplace	90.6	31.1	92.3	64.4	1853
InventorOf	88.0	17.5	81.3	50.8	682
WonAward	62.5	15.3	73.6	52.8	354
All	**75.0**	18.4	73.9	**58.4**	5930

The two systems achieve comparable levels of precision.
But recall is much higher for R1-CRF!

However … R1-CRF reqiures labeled training data for the relation, while O-CRF was not trained specifically for this relation.

How Much Training Data is Needed?

So they looked at learning curves to determine how much labeled training data was necessary to achieve roughly the same precision.

	O-CRF		R1-CRF		
Relation	**P**	**R**	**P**	**R**	**Train Ex**
Acquisition	75.6	19.5	67.6	69.2	3042*
Birthplace	90.6	31.1	92.3	53.3	600
InventorOf	88.0	17.5	81.3	50.8	682*
WonAward	62.5	15.3	65.4	61.1	50
All	**75.0**	18.4	70.17	**60.7**	>4374

- For the WonAward relation, 50 training examples were needed.
- For the BirthPlace relation, 600 training examples were needed.
- For the Acquisition and InventorOf relations, R1-CRF never achieved comparable precision, even with substantial training data.

Analysis of Results

- R1-CRF benefits a lot from the lexical features.

  Example: "*Yahoo to Acquire Inktomi*"

  Acquire is mistagged as a proper noun, so O-CRF is confused. But R1-CRF still recognizes "*acquire*" as a relation trigger.

- O-CRF also failed to recognize synonyms for the relation.

  R1-CRF identified 16.25 synonyms per relation, on average. With RESOLVER, O-CRF found only 6.5 synonyms per relation.

- Conclusions:
  - Open IE provides good precision without relation training data.
  - But when higher recall is needed and manually labeling data is possible, traditional RE is desirable.

Evaluating the Hybrid Extractor

	R1-CRF			Hybrid		
Relation	**P**	**R**	**F1**	**P**	**R**	**F1**
Acquisition	67.6	69.2	68.4	76.0	67.5	71.5
Birthplace	93.6	64.4	76.3	96.5	62.2	75.6
InventorOf	81.3	50.8	62.5	87.5	52.5	65.6
WonAward	73.6	52.8	61.5	75.0	50.0	60.0
All	73.9	**58.4**	65.2	**79.2**	56.9	**66.2**

- Using both O-CRF and RI-CRF in the stacked ensemble framework produces better precision (79%) than either one alone.
- Recall does not improve, but is nearly as good as the R1-CRF.
- The hybrid approach requires labeled training data for the relation, so the trade-off is manual effort for higher precision.

Conclusions

- Open Information Extraction holds great promise for automatically constructing large and rich knowledge bases.
- These efforts have advanced the state-of-the-art for robustly and efficiently extracting large volumes of diverse knowledge from unstructured, often unwieldly Web text.
- However, there is ample room for improvement in the accuracy, organization, and richness of the learned knowledge.
- Open IE learners tend to learn the most prevalent facts and relations, and are less able to learn less common knowledge or acquire specialized concepts with domain-specific idiosyncracies.