Two applications: Information Extraction and Open Domain Question Answering

Stephen.Pulman@comlab.ox.ac.uk

Information Extraction

Go from text (e.g. newspaper reports) to a stereotyped template containing key facts: who, what, when, where, why?

Example: Terrorist actions

> *Salvadoran President-elect Afredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of the crime.*

Incident:	killing
Perpetrator:	FMLN
Confidence:	accused
Human Target:	Roberto Garcia Alvarado

INFORMATION EXTRACTION

Company succession events:

> *Former BBC One and Channel 4 boss Michael Grade has been confirmed as the new BBC chairman. The former BBC executive, 61, takes on a role vacated by Gavyn Davies, who resigned in the wake of criticism in the Hutton Report in January.*

ORG:	BBC
POSITION:	Chairman
IN:	Michael Grade
OUT:	Gavyn Davies

Many other applications: financial, jobs, biology (protein interactions), etc.

Template components

These are usually specific to the application, but some (person, organisation, location, time) are common to many.

Entities: individuals, organisations, roles, locations, times, dates, amounts of money...

Properties: terrorist, illegal, profitable, popular...

Relations: X killed Y, A fired B, C confirmed D...

Times: dates, order of events, durations...

Components of an IE system

1. We can use a tagger and NP/Verb Group chunker to find the entities and relations.

2. NP/VG classifier/named entity recogniser: a way of classifying NPs or VGs according to the type of entity or relation they describe: people, locations, dates, organisational roles, amounts of money, etc. There are various methods for doing this.

3. Reference resolver: find references of pronouns and definite descriptions.

4. Template filling and merging: fillers for template slots may be in different sentences; there may be repetition.

Named Entity Recognition

- **write rules by hand**

IF NP contains ‘Co.’, ‘Ltd’, ‘Pty’, ‘Bvd’ etc.
THEN *company name*

IF NP contains ‘Jan’, ’January’, etc. or of the form 1-31/1-12/19XX;200X, etc.
THEN *date*

IF NP matches ‘chief (executive/finance/operating/technical) officer’
THEN *company role*

- **use a machine learning method** (Naive Bayes, MaxEnt, decision trees, etc.) to classify NPs according to whether or not they have various features.

Possible features: Contains proper name (upper case); contains number + Jan, Feb etc.; contains Co., Ltd., Corp.; contains ‘chief’, ‘officer’, ‘executive’, CEO etc.

Can also use HMM methods similar to the way they are used in chunking: have B, I, O, notation for named entity sequences

Co-training: learning named entity rules

Collins and Singer's algorithm:

Task: learn rules that will decide whether an NP is a PERSON, ORGANIZATION, or LOCATION. Rules can use features *internal* to the NP (e.g. contains(Mr.)), or *external*, e.g. appositiveNP(Company):

```
NP    , APPOSITIVE NP,
Hadson, the energy and defence  company, said...
```

NP contains(Mr) →PERSON
NP appositiveNP(Company) →ORG

Each rule has a 'strength' associated with it. On new data, an NP gets the label (if any) assigned by the rule with the highest strength. Strength for feature *x* and label *y* is defined

$$\text{h(x,y)} \approx \text{P(x} \mid \text{y)} = \frac{\text{Count(x,y)}+\alpha}{\text{Count(x)} + \text{k}\alpha}$$

Count(x,y) is the number of times feature *x* is seen with label *y* in the training data. Count(x) $= \sum_y$ Count(x,y). α is a smoothing parameter (they set it to 0.1), and *k* is the number of labels (here 3).

The basic idea

The idea (taken from Yarowsky) is that you start with some simple hand built internal rules which are highly accurate. Assign these rules a high strength (0.99).

1. Label the training data with the current set of internal rules.

2. Use the labelled data to induce some high strength external rules.

3. Label the training data using the current set of external rules.

4. Use the labelled data to induce high strength internal rules.

5. Until some threshold of coverage or accuracy is reached, go to 1.

In more detail:

1. Set *n*=5 (*n* is the maximum number of rules of each type induced at each iteration.)

2. Initialization: Set the current internal rules equal to the set of seed rules.

3. Label the training set using the current set of internal rules. Examples where no rule applies are left unlabeled.

4. Use the labeled examples to induce a decision list of external rules, (by filling templates). Add these to the current external rules set.

Let Count'(x) be the number of times feature *x* is seen with some known label in the training data. For each label (Person, Organization and Location), take the *n* contextual rules with the highest value of Count'(x) whose **unsmoothed** strength is above some threshold P_{min} (=0.95 here). (If fewer than *n* rules have precision greater than P_{min} , we keep only those rules which exceed the precision threshold.)

5. Label the training set using the current set of external rules. Examples where no rule applies are left unlabeled.

6. On this new labelled set, induce up to n*k internal rules, by filling templates. Add them to the current internal rules set

7. If n < 2500, let n = n+5 and go to 3. Otherwise, label the training data with all the rules, induce any new internal and external rules and return the whole set of rules.

Example

Start with internal rules:

contains(Mr) → PERSON
contains(Ltd.) → ORGANISATION
contains(U.K.) → LOCATION

Label new examples:

[Mr Smith]/PER said that his mother...

We gave the contract to [Frog Ltd.]/ORG, a small start-up, ...

They are based in [the U.K.]/LOC and export...

Induce new external rules:

subjectOf(say) → PERSON
appositiveNP(start-up) → ORGANIZATION
objectOf(base-in) → LOCATION

Template merging

Hadson Corp. ... report a third quarter net loss of $17 million to $19 million

The Oklahoma City energy and defense concern...

ORG:	Hadson Corp.
LOC:	Oklahoma City
TYPE:	Energy and defense
PROFIT-OR-LOSS:	$17-19m
WHEN:	third quarter

How well do we do?

Recall, Precision, F-measure

Recall = number of items found/ number of items to be found

Precision = number of correct items found/ number of items found

F measure is a combination of these two:

$$F_\alpha = (1+\alpha) \cdot (\text{precision} \cdot \text{recall})/(\alpha \cdot \text{precision} + \text{recall}).$$

α generally = 1.

Information	Approx Percentile Reliability
Named Entities	90
Properties	80
Relations	70
Complete templates	60

Question Answering

```
* What is the deepest lake in the US?
* find free software to increase internet speed
* How large is Missouri’s population?
* What is the current population in Bombay, India?
* What do river otters eat?
* What percentage of the population is left handed?
* who is the President of Turkey?
* How many miles is it from London, England to Plymouth, England?
* What is the airport code for Los Angeles International?
* what is the sniper’s name?
* How many gallons will my tank hold?
* who is the president of Microsoft?
* How many legs does a caterpillar have?
* What is the anorexia nervosa?
* How can I best care for my dog’s health?
* How big is the planet Venus?
```

Some history

1961 Baseball:
Who did the Red Sox lose to on July 5th?
On how many days in July did eight teams play?

1973 Lunar:
What is the average concentration of aluminum in alkali rocks?
How many Brescias contain Olivine?

1983 Team:
Show each continent's highest peak?
What northern countries contain peaks higher than Fuji?

Sentence → Tree → Logical Form → SQL or similar

Limited domains and coverage; typically rather fragile processing.

Information retrieval vs. question answering

In the 90s, the success of Alta Vista, Google etc. gave rise to ‘open domain’ question answering, where answers = short text fragments containing the answer, rather than a list of whole documents.

Typical architecture:

```
         Context
Question    -->      Question Analysis --> Document Selection -->
         User Model

Document Analysis --> Answer Extraction  --> Response Generation
```

Question Analysis

Aim to determine the expected answer type, extract keywords, determine what relations must hold of candidate answers:

Who .... → Person
Where ... → Place
When ... → Time
How many ... → Number

But many are not so easy:

Which ...
What ... Name the...
List the ...
How ...

Some approaches also extract the relevant nominals: e.g. What **football team** sacked its manager last week?

Many groups develop a more fine-grained taxonomy of answer types: e.g. person_in_show_business, capital_city.

Document Selection and Answer Passage Location

Usually done in two stages:

* IR methods to locate documents containing correct key words

* then IE type processing to check that the right entities and relations are mentioned in them.

If a parser is used, the document collections may have been analysed off-line to speed things up.

* Relevant paragraph(s) selected rather than whole document

Answer Extraction

Various types of processing may be performed on the candidate answers:

* parsing and translation into a meaning representation

* classification using vector space techniques

* is answer of correct type, as determined by question analysis

* check via inference whether answer is correct

Inference

Some systems use logical inference. Linguistic and other knowledge is used to make links between question and answer. This gives better results than e.g. keyword overlap:

Q106.2 (1998 Baseball World Series) What is the name of the losing team?

Candidate: San Diego Padres team that lost to the Yankees in the 1998 World Series.
Original exact: Yankees
Axiom derivational morphology rule triggers:
lose_VB(e1,x1,x2) → losing_JJ(x1)
Final exact: San Diego Padres

Q91.4 (Cliffs Notes) What company now owns Cliffs Notes?
Candidate: Cliffs Notes was bought by IDG Books Worldwide.
Axiom: buy_VB(e1,x1,x2) → own_VB(e2,x1,x2)
Final exact: IDG Books Worldwide

Lexical Chains

WordNet glosses can contain information useful for understanding conceptual relatedness, causal explanations, etc.

```
S1: Jim was hungry.
S2: He opened the refrigerator.
Q: What did Jim want?

WN: hungry: feeling a need or desire to eat food
WN: refrigerator: a kitchen appliance in which food can be stored at
low temperature.
WN: want: feel or have a desire

A: Jim wanted to eat/ Jim wanted food.

Q: What day and month did John Lennon die?
Answer passage: Similarly, former Beatle John Lennon
was slain Dec 8th, 1980, by a deranged fan outside his
New York apartment...
```

X was slain → Y slayed X → Y killed X → X died

TREC QA 8

How many calories are there in a Big Mac?
What two US biochemists won the Nobel Prize in medicine in 1992?
Who was the first American in space?
Who is the voice of Miss Piggy?
Where is the Taj Mahal?
etc.

Mean reciprocal rank: The MRR of each individual query is the reciprocal of the rank at which the first correct response was returned, or 0 if none of the first N responses contained a correct answer.

The best 5 systems in the TREC 8 evaluation scored MMR of 0.317 - 0.66 for a 50 byte long answer, and 0.471 - 0.646 for a 250 byte long answer. Later systems have improved a little on these figures.

References

http://trec.nist.gov/pubs/

L. Hirschman and R. Gaizauskas, 2001, Natural language question answering: the view from here, Nat. Lang. Eng., 7:4 pp275–300.

M. Collins and Y. Singer, 1999, Unsupervised models for named entity classification, In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. (on ACL anthology: `http://aclweb.org/anthology-new/`)

Both of these links contain many further references:

`http://gate.ac.uk/sale/ell2/ie/main.pdf`

`http://www.ai.sri.com/~appelt/ie-tutorial/`

Demo: `http://services.gate.ac.uk/annie/`

Long QA bibliography at:

http://answerbus.coli.uni-sb.de/bibliography/index.shtml

Demos:

http://www.trueknowledge.com/

http://demos.inf.ed.ac.uk:8080/qualim/

http://brahms.isi.edu:8080/textmap/

http://www.answerbus.com