Learning from Text: Non-linguistic knowledge from linguistic patterns

S G Pulman

- need non-linguistic knowledge for many applications: e.g. Information Extraction (govt. announcements): ‘Washington has warned Iran over nuclear power’. You need to know that capital cities are often used as a description of governments (London, Berlin, Paris etc.)

- Question answering: ‘Who is the President of England?’ - to give helpful answer, need to know that England doesn't have a President, but that President and Prime Minister are the same type of thing.

- ... and for good quality NLP: disambiguation, reference resolution, etc.
**Hearst: acquiring hyponymy relations**

The linguists’ term for inclusion relations:

If ‘A is a kind of B’ is true, then A is a ‘hyponym’ of B (and B is a ‘hypernym’ of B).

E.g. a tabby is a kind of cat; a cat is a kind of animal

This kind of knowledge is often included in what have come to be called in the semantic Web world as ‘ontologies’ (a strictly inaccurate use of the term).

Main idea: some syntactic patterns unambiguously signal hyponymy relations:

- The bow lute, such as the Bambara ndang, is plucked and has an individual curved neck for each string.

→ “Bambara ndang” is a hyponym of “bow lute”

(even if you don’t know what either of them is.)
Is this true? Google results for “such as” yesterday:

How can I insert special characters, such as dingbats and accented letters, in my document?

MIME Encapsulation of Aggregate Documents, such as HTML (MHTML).

Discounted brand name watches such as Nautica, Seiko, Tissot ...

Certain Foie Gras Linked To Diseases Such As Alzheimer's And Arthritis, Animal Study Suggests

Suits to clear name, such as Clemens’, are rare, risky.

Economic Issues, Such as Health Care Costs, Top Concerns for Democratic, Republican Voters
Linguistic Patterns must:

- occur frequently and in many text genres
  (so that we find lots of examples)

- be strong indicators of the relation of interest
  (so that we minimise false positives)

- be able to be found without deep linguistic analysis
  (so we can process at the current state of the art)
Hearst’s patterns:

such NP as {NP,}* {(and | or)} NP

...works by such authors as Herrick, Goldsmith, and Shakespeare.

“Herrick/Goldsmith/Shakespeare” hyponym of “author”

NP as {NP,}* {,} {(and | or)} other NP

Bruises, wounds, broken bones or other injuries ...
temples, treasuries, and other important civic buildings.

“Bruises/wounds/broken bones” hyponym of “injury”
temple/treasury” hyponym of “civic building”

NP, including {NP,}* {and | or} NP

All common-law countries, including Canada and England...

“Canada/England” hyponym of “common-law countries”

NP, especially {NP,}* {and | or} NP

... most European countries, including France, England and Spain.

“France/England/Spain” hyponym of “European country”
Aside: learning patterns automatically

Hearst outlines an unimplemented method for learning such patterns. In fact, the method used by Collins and Singer for learning Named Entity classification rules would work well here.

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

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

Count($x, y$) is the number of times feature $x$ is seen with label $y$ in the training data. Count($x$) = $\sum_y$ Count($x, y$). $\alpha$ 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 $\text{Count}'(x)$ 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 $\text{Count}'(x)$ whose **unsmoothed** strength is above some threshold $P_{\text{min}}$ ($=0.95$ here). (If fewer than $n$ rules have precision greater than $P_{\text{min}}$, 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 $n \times k$ 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
Apply to Hearst patterns:

0. Initialise with hand-written patterns.
1. Use the current patterns to find examples of hyponyms.
2. Search for other occurrences of this hyponyms.
3. Induce new patterns from these occurrences, add them to the list.
4. While still improving, go to 1.

Example:

... most European countries, including France, England and Spain.

“France/England/Spain” hyponym of “European country”

Google search for “England France Europe” gives on first page:

Soccer Schools and Soccer Academy for Boys and Girls in Europe (England, France, Spain and Italy)

Craft Guilds in North-Western Europe (England, France, Low Countries).

- suggesting possible pattern: NP ‘(’{NP,}+(and) NP ‘)’
Extending to part-whole relations

Hearst reported that she had tried to also discover part-whole relations (‘mereonyms’) like ‘finger/hand’ or ‘bathroom/house’ but with little success.

Berland and Charniak had more success: as before, use clear cases to find examples:

...the basement of the building
... the basement in question is in a four-storey apartment building...
...the basement of the apartment building..
From the building’s basement...
...the basement of a building..
...the basements of buildings...

Induce patterns from the examples: (NN=singular noun, NNS=plural noun)

a. <whole>+NN/NNS 's/POS <part>+NN/NNS 'building's basement'
b. <part>+NN/NNS of {the\|a} (JJ|NN)* <whole>+NN/NNS 'basement of a building'
c. <part>+NN in of {the\|a} (JJ|NN)* <whole>+NN 'basement in a building'
d. <part>+NNS of <whole>+NNS 'basements of buildings'
e. <part>+NNS in <whole>+NNS 'basements in buildings'
Results:

Patterns c-e are too general, so only a and b produced useful results. Also, need to filter the results so that we only retain pairs where \( P(\text{whole} \mid \text{part}) \) is high and also where \( \text{count(whole+part)} \) is high. They do this by comparing \( P(\text{whole} \mid \text{part}) \) and \( P(\text{whole}) \), where these are estimated by frequency of occurrence in the patterns found.

Results for ‘car’= whole:

airbag, brake, bumper, dashboard, driver, fender, headlight, hood (bonnet in British English), ignition, occupant, pipe, radiator, seat, shifter (gearstick Br. Eng?), speedometer, tailpipe, trunk (= boot Br. Eng), vent, wheel, windshield

In tests, human subjects rated 70% of the top 20 words found for a given ‘whole’ as accurate.
VerbOcean: semantic relations between verbs

Chklovski and Pantel use a similar technique again to discover relations between verbs. The relations they target are:

Semantic similarity: maximise::enhance, produce::create, reduce::restrict.

Strength (where one verb describes a more extreme case of the other): taint::poison, permit::authorize, surprise::startle, startle::shock

Antonymy (‘oppositeness’ of meaning): buy::sell, lend::borrow, live::die, differ::equal, fail::succeed, open::close, assemble::dismantle

Enablement (you have to do X to do Y): fight::win, try::succeed

Happens-before: marry::divorce, arrest::prosecute, schedule::reschedule, tie::untie
Method

1. find pairs of verbs that occur together frequently: in particular, using a dependency parsed corpus, verbs that link similar nouns (e.g. person buy thing from person, person sell thing to person)

2. apply patterns to instances to determine set of possible relations

3. rank the results using a mutual-information-like measure

They use Google to find instances from the web.
## Patterns

<table>
<thead>
<tr>
<th>Relation</th>
<th>Patterns</th>
<th>number of hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Narrow similarity</td>
<td>X(ed) i.e. Y(ed)</td>
<td>219k</td>
</tr>
<tr>
<td>Broad similarity</td>
<td>Xed and Yed, to X and Y</td>
<td>154m</td>
</tr>
<tr>
<td>Strength</td>
<td>X(ed) (and) even Y(ed)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Y(ed) or at least X(ed)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>not only/just Xed but Yed</td>
<td>1m total</td>
</tr>
<tr>
<td>Enablement</td>
<td>Xed .. by Ying the/or..</td>
<td>2.3m total</td>
</tr>
<tr>
<td></td>
<td>to X .. by Ying the/or...</td>
<td></td>
</tr>
<tr>
<td>Antonymy</td>
<td>either X(s/ed/ing) or Y(s/ed/ing)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>whether to X or Y</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Xed ... but Yed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to X ... but Y</td>
<td>18m total</td>
</tr>
<tr>
<td>Happens-before Adverb</td>
<td>to X/Xed (...) and Adverb Y/Yed</td>
<td>8.2m</td>
</tr>
<tr>
<td></td>
<td>= then/later/subsequently/eventually</td>
<td></td>
</tr>
</tbody>
</table>
Testing for the strength of association

\[ S_{pattern}(V1,V2) = \frac{P(V1,\text{pattern},V2)}{P(\text{pattern}) \times P(V1) \times P(V2)} \]

We need to distinguish between symmetric relations (like antonymy) from asymmetric ones (happens-before) because in the symmetric case the verbs can appear in either order. Chklovski and Pantel approximate probabilities by using the number of Google hits, and the occurrence of the patterns in a smaller 500m tagged corpus.

\[ P(V1,\text{pattern},V2) \approx \frac{\text{hits}(V1,\text{pattern},V2)}{N}, \text{ where } N \text{ is the number of words Google (then) indexed (} \approx 7.2 \times 10^{11}) \].

\[ P(\text{pattern}) \approx \frac{\text{hits(\text{pattern}) in tagged corpus}}{\text{number of words in corpus}} \]

This estimate is doubled for the symmetric relations.

\[ P(V) \approx \frac{\text{hits(\text{V})} \times 8.5}{N} \]

- where 8.5 is a correction factor to estimate the number of morphological variants of the verb forms.
Results

Found nearly 30k pairs of verbs. Humans judged a sample of 100.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Tags correct</th>
<th>Preferred tag correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>similarity</td>
<td>63.4</td>
<td>40.2</td>
</tr>
<tr>
<td>strength</td>
<td>75.0</td>
<td>75.0</td>
</tr>
<tr>
<td>antonymy</td>
<td>50.0</td>
<td>43.8</td>
</tr>
<tr>
<td>enablement</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>happens-before</td>
<td>67.6</td>
<td>55.9</td>
</tr>
<tr>
<td>no relation</td>
<td>72.9</td>
<td>72.9</td>
</tr>
</tbody>
</table>

Average: 67.6 55.9

Note that the relations are not disjoint: e.g. strength overlaps with similarity, and enablement overlaps with happens-before.
References

All at http://www.aclweb.org/anthology

Marti A. Hearst, 1992, Automatic Acquisition of Hyponyms from Large Text Corpora, COLING, pp539-545.

