Information Extraction from Informal Texts

Lyle Ungar
University of Pennsylvania

What works, what doesn’t
When are machine learning and NLP useful?
IE from Informal Texts: Two Case Studies

◆ Web scale IE
  ● What I learned at Google
  ● Large scale machine learning for NER
    ■ Many entity types, many document types

◆ IE from product discussion forums
  ● How to extract product discussion data
  ● What to do with it once you have it
    ■ Comparative sentiment analysis

How to use entity lists to avoid needing hand-labeled training data
Web Scale Information Extraction

Alex Kehlenbeck, Casey Whitelaw, Lyle Ungar

Google
Web scale entity recognition is hard

- Many kinds of entities
  - People, movies, books, animals, moon craters, ships, songs, companies, gods, cities, …
  - Actors, politicians, senators, baseball players, rabbis, …
- Much ambiguity
  - “Paris” - a city, a person, a book, …
  - “Cambridge” - a city, a school, a publisher, …
- Many sources and formats
  - Full sentences, lists, tables, phrases
  - Different styles on different web sites

Too complex to write rules for every case,
Too many cases to hand label training data
Entity resolution is a key problem
Standard Named Entity Recognition

- High quality hand-labeled training data
- Small number of entities
  - Person, place, organization
  - Gene/protein
- Extract from a single corpus of grammatical text
  - AP Newswire
  - Medline abstracts
- Use fancy machine learning
  - E.g., CRF

Trek, MUCC, CoNNL, BioCreative
How to get training data?

Key insight: there are lots of entity lists
Generating Training Data

◆ Start with lists
  ● Lists of entities of known type
    ■ Company, animal, ship, moon crater, fictional character, mathematicians, wineries
  ● Pairs of entities of known type
    ■ director, movie
    ■ author, book

◆ Find all mentions of these strings on the web
  ● And some simple types like dates, addresses, etc.

◆ Extract their context
  ● … <director> directed <movie> in <year>…

Casey Whitelaw
artist - <NAME>, american <profession> and comedian
artist . <music> : <NAME> (vocals)
band currently listening <album> by <NAME> see
bird ( <genus> rustica ) <NAME> ( 
city <NAME>, [population (metro area)] ( metro .
country held in <capital>, <NAME>
film <NAME> runs <length>. it
film <NAME> starring: <cast>
film watching <NAME> by <cast> see related
ship : <propulsion> armament: motto: <NAME>
wrestler - <NAME>, <nationality> professional wrestler
Keep contexts that are pure in terms of the entity types they contain
- \(<NAME>\) was born \(<Date>\)
- E.g., \(<NAME>\) always is of type person (if it is known) but won’t be pure in terms of actors

Find those contexts on the web
- And the terms in them
- Each occurrence is now a labeled example

High precision, low recall labeling
- The examples are not perfect
Extending Training Data

Find additional examples by finding links

- To the page with the labeled examples
  - Term in anchor text points to page
  - Labeled terms in the page are all of the same type
  - e.g. “San Francisco”
- From the labeled examples
  - Labeled term in anchor text
  - Find same term in page pointed to
Features Used

- local context
  - words left and right of the term
- the tokens in the mention
- base classifiers
  - E.g. person recognizer
- domain of the URL of the page
- type labels of the page
- item type distribution on page
- item type distribution on domain
- membership of the term on lists
- case signature of term
## School: most significant features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>document category</td>
<td>vehicle licensing</td>
</tr>
<tr>
<td>document category</td>
<td>automotive</td>
</tr>
<tr>
<td>left 0 tokens</td>
<td>“Unified”</td>
</tr>
<tr>
<td>right 2 tokens</td>
<td>“-”</td>
</tr>
<tr>
<td>case signature</td>
<td>Aa Aa Aa</td>
</tr>
</tbody>
</table>
- type-labeler                | album      |
- type-labeler                | movie      |
- type-labeler                | company    |
- type-labeler                | place      |
- person-labeler              | person     |
<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>document category</td>
<td>business</td>
</tr>
<tr>
<td>document category</td>
<td>news</td>
</tr>
<tr>
<td>left 0 tokens</td>
<td>“friend”</td>
</tr>
<tr>
<td>left 1 token</td>
<td>“a”</td>
</tr>
<tr>
<td>case signature</td>
<td>Aa Aa</td>
</tr>
<tr>
<td>case signature</td>
<td>Aa</td>
</tr>
<tr>
<td>document category</td>
<td>computers\electronics</td>
</tr>
</tbody>
</table>
**Movie: most significant features**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>document category</td>
<td>entertainment</td>
</tr>
<tr>
<td>right 2 tokens</td>
<td>“}”</td>
</tr>
<tr>
<td>document category</td>
<td>movies</td>
</tr>
<tr>
<td>type-labeler (partial match)</td>
<td>album</td>
</tr>
<tr>
<td>labels in domain</td>
<td>album</td>
</tr>
<tr>
<td>type-labeler</td>
<td>company</td>
</tr>
<tr>
<td>type-labeler</td>
<td>place</td>
</tr>
<tr>
<td>case signature</td>
<td>Aa</td>
</tr>
<tr>
<td>case signature</td>
<td>Aa a</td>
</tr>
</tbody>
</table>
How to do scalable ML?

◆ Data set is big
  ● Millions of observations (mentions)
  ● 100 entity types (y)
    ■ Partly mutually exclusive, partly subsets
  ● Millions of features (X)

◆ Data set is sparse
  ● Most words are rare
Scalable Machine Learning

◆ Learning in a map-reduce environment
  ● Randomly divide training data over e.g., 1,000 machines
  ● Train a model on each subset of the data
  ● Look at each observation once
    ■ Online learning

◆ How to use many kinds of entities?
  ● Share features across multiple predictions
    ■ Error Correcting Output Coding (ECOC)

◆ Screen for Feature selection
  ● compute a t-statistic for each feature independently
  ● keep those above a threshold
Online Learning - Perceptron

- **Update weights after each observation**
  - If prediction is correct, no change
  - If prediction is wrong, change $w$ by enough ($\eta$) to make the prediction correct

$$\delta \leftarrow y - f(w^T x)$$  
Correct or wrong?

$$\forall j \; w_j \leftarrow w_j + \eta \; \delta x_j$$  
Update weight

- $f$ does thresholding

---

Koby Crammer, Ofer Dekel, Shai Shalev-Shwartz,
Yoram Singer
Perceptron Learns Maximum Margin

- Weights only change if a prediction error was made
  - Only support vectors effect weights
- Can cleverly choose the step size, $\eta$
  - Adjust the weights just far enough to correctly classify the new point
    - This simple optimization can be done closed form
    - “passive-aggressive algorithm”

Koby Crammer, Ofer Dekel, Shai Shalev-Shwartz, Yoram Singer
Error Correcting Output Codes

- Use distributed code for labels
  - Map labels for each observation to a vector of code words
  - Learn to predict code vector
    - Then “decode” it to get a prediction
  - Can take advantage of hierarchical structure of entity labels

- Gives more accurate models than predicting each class separately

<table>
<thead>
<tr>
<th></th>
<th>code1</th>
<th>code2</th>
<th>code3</th>
</tr>
</thead>
<tbody>
<tr>
<td>class1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>class2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>class3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Solving multiclass learning problems via error-correcting output codes

Dietterich and Bakiri
Conclusions - Web Scale IE

◆ **Getting training data is key**
  - Use databases of known entities
    - Learn templates
  - Use many *different* entity types to get negative examples
    - For template purity
    - For entity type prediction

◆ **Online learning works well**
  - More data is better than fancier methods
    - All types of features improved performance
  - Build models for all entity types simultaneously
    - ECOC gives significant boost

◆ **Result:** circa 90% accuracy
  - Varies by entity type

◆ **Next:** cluster all mentions to get specific entities
Information extraction and analysis from product discussion forums

Ronen Feldman, Moshe Fresko, Jacob Goldenberg, Oded Netzer, Lyle Ungar
I have some experience with the Burn, and I race in the T4 Racers. Both of them have a narrow heel and are slightly wider in the forefoot. The most noticeable thing about the T4s (for most people) is the arch. It has never bothered me, but some people are really annoyed by it. You can cut the arch out of the insole if it bothers you. The Burn's arch is not as pronounced.
Honda Accords and Toyota Camrys are nice sedans, but hardly the best car on the road (for many people). It's just that they are very competent in their price range. So, a love fest of the best selling may not tell you what is "best". That depends very much on what is important to you. A car could have a quirk, that you would just love, but not be popular to many people. Thus, the best car for you might not sell many. If you are looking for resale value, then it might be a factor.
Sentiment Analysis

- Extract from text how people feel about different products

- Sentiment analysis can be tricky
  - Honda Accords and Toyota Camrys are nice sedans
  - Honda Accords and Toyota Camrys are nice sedans, but hardly the best car on the road
Sentiment Analysis is Hot

◆ Lots of companies
  ● BuzzMetrics, Reputica, Umbria, Cymfony, BuzzLogic, SentiMetrix

◆ Lots of research
 钡 BuzzLogic LingPipe

钡 Nielsen BuzzMetrics
钡 umbria
钡 REUTERS
钡 IVolatility.com
钡 reputica
Relative sentiment analysis

- A.k.a product comparisons
  - Extract how people compare products
- Requires determining the products and the dimensions they are compared on
  - The 2052's will be an ounce lighter than the current 2051's
Product Comparison Examples

- The 2052's will be an ounce lighter than the current 2051's
- I have just added the avi-lite to my rotation-this shoe rocks! It seems lighter than my Nike Zoom Elites 2
- Nike AST 9 is softer than both Adrenaline and Inspire
- The 2120's are not soft or firm but if i had to settle on one they lean to a bit softer than 2110-speva midsole
Product Comparison Questions

- What is being compared to what?
  - I’m looking at this running shoe - what is “comparable”?

- What dimensions are they compared on?
  - What shoe attributes are discussed?

- Which product is preferred on this dimension?
Extraction Process

- Label brands and models
- Find snippets
  - small piece of text involving two products
- Extract comparison snippets
- Categorize snippets
  - Are products viewed as similar, different or neither?
Label Brands and Models

❖ Start with a list of brands and models
  ● Easily available on the web
❖ Brand names are mostly standardized
  ● Honda, Toyota, Ford, ...Nike, Asics, ...
  ● But often skipped
❖ Model names are highly variable
  ● "Mizuno Wave Alchemy 4", "Achemy 4", "wave alchemy 4", "alchemy 4", "Mizuno Alchemy 4"
  ● But can be extracted with fairly simple patterns
    ■ Any combination of "Mizuno" and "Wave" and "Alchemy", where each can be abbreviated or misspelled + "V" or "5"

❖ Results
  ● F1 = 92.9%. (precision = 96.7% recall = 89.4%)
Snippet Extraction

- **Preprocessing**
  - the text is tagged with product models and with parts of speech
  - a shallow parser is used to extract simple noun phrases.

- **Pattern extraction**
  - frequently occurring patterns, each containing either one or two product models, are extracted

- **Pattern filtering**
  - the poor patterns are removed.

- **Snippet extraction**
  - The patterns are run over the corpus, and all matching snippets are extracted.

- **Snippet filtering**
  - Poor quality or overlapping snippets are removed.
Extract Comparison Snippets

◆ Patterns

\(<Model> \ast <adjective> \ast than \ast <Model>\)
\(<Model> \ast <adjective> \ast to \ast <Model>\)

◆ Examples

- 1024 is very similar to my 1023
- 1110 which I can get cheaper than the 1120
- 2051 is noticeably heavier than the inspire 2
- 2051 - stability trainer with significant cushioning (more stable than, say, Asics 21 XX and more cushioned than Adrenaline)
- 2051 (actually is closer to neutral version of upcoming 2052)
- 2051 is more comparable to the 2110 than kayano
Categorize Snippets

◆ Are pairs of items called similar, different or neither?
  ● *Sonata* has soft ride similar to *Camry and Accord*
  ● *300 C Touring* looks so much better than the *Magnum*
  ● *Honda Accords* and *Toyota Camrys* [are nice sedans…]

◆ Rule
  ● “similar” or “comparable” → *similar*
  ● “differ” or “more” or “less” or /JJR/ → *different*
  ● else → *other*

◆ Support Vector Machine (SVM)
  ● 200 snippets labeled on each of shoes and cars

◆ Results (micro-averaged F1)
  ● Rule SVM
  ● 0.785 0.66 Train on cars, test on shoes
  ● 0.82 0.76 Train on shoes, test on cars
# Discussion Board Data Size

<table>
<thead>
<tr>
<th></th>
<th>Car</th>
<th>Shoe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages:</td>
<td>868,174</td>
<td>19,410</td>
</tr>
<tr>
<td>Sentences:</td>
<td>5,972,695</td>
<td>65,010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Car</th>
<th>Shoe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Different Types</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brands</td>
<td>28</td>
<td>20</td>
</tr>
<tr>
<td>Models</td>
<td>180</td>
<td>200</td>
</tr>
<tr>
<td>Terms</td>
<td>1,037</td>
<td>188</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mentions (Tokens)</th>
<th>Car</th>
<th>Shoe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>503,895</td>
<td>20,768</td>
</tr>
<tr>
<td>Models</td>
<td>575,110</td>
<td>32,606</td>
</tr>
<tr>
<td>Terms</td>
<td>6,194,507</td>
<td>74,931</td>
</tr>
</tbody>
</table>
Car Brand Co-occurrence Patterns
Car Brand Co-occurrence Patterns
Shoe Comparisons

Pajak graph display??
Shoe Comparisons - Detail

Note centrality of Asics to comparisons
MDS of Car Industry

Axis dimensions are determined automatically from co-occurrence data
Comparative Sentiment Results

◆ The Camry is bigger and has more leg room
  ● Camry, Accord, size, bigger
  ● Camry, Accord, leg_room, better

◆ The Accord is noisier than the Camry
  ● Camry, Accord, noise, better

◆ The Accord is more sportive and drives better.
  ● Accord, Camry, sportive, better
  ● Accord, Camry, drive, better
  ● Accord, Camry, performance, better

◆ The Camry looks better but the Accord has a better interior

◆ Comments on the price are very mixed
Conclusions - Product Discussions

◆ Simple rules work well
  ● for entity extraction
    ■ Start with lists of product brands and models
    ■ Parsing fails due to non-grammatical “sentences”
    ■ Comparisons often span sentence boundaries
  ● for labeling same/different comparison snippets
    ■ Part of speech (POS) tags help

◆ Insight into consumer decision process
  ● What is being compared to what
    ■ Multi Dimensional Scaling (MDS) on comparisons
  ● Relative sentiment analysis
    ■ “Accord is noisier than the Camry”

◆ Good results with 80% labeling accuracy
Conclusions

- Information extraction from informal texts works
  - As long as you can live with 80-90% accuracy

- Use lists of entities
  - As a seed to learn templates for getting training data for NER
  - As the basis for hand-crafted entity (brand) extraction rules

- Machine learning helps if you have lots of training data
  - Simple rules work better if you don’t
  - Template learning is great on large unlabeled corpora

- NLP sometimes helps
  - Not used at all in Google project
  - Part of speech tagging is useful to recognize comparisons
  - Parsing does not help (yet???)

- Relative sentiment analysis
  - shows what is compared to what on what dimensions