Chapter 22

Natural Language Processing
Why should agents do NLP?

• Knowledge acquisition from spoken and written language artifacts (e.g. on the web)
  – This chapter
  – *Natural* language is messy!

• Communicate with humans
  – Chapter 23
Outline

• Language Models
  – Predict the probability distribution of language expressions

• Information-Seeking Tasks
  – Text Classification
  – Information Retrieval
  – Information Extraction
Language Models

- Formal languages (e.g. Python, Logic)
  - Grammar (generative)
  - Semantics

- Natural languages (e.g. English)
  - Grammaticality is less clear
    - *To be not invited is sad*
  - Ambiguity at many levels (syntax, semantics, ...)
    - I saw the man with the telescope
    - He saw her duck
  - Suggests modeling via probability distributions
    - What is the probability that a random sentence would be a string of words?
    - What is the probability distribution over possible meanings for a sentence?
N-Gram Models

• N-Gram
  – a sequence (of some unit – characters, words, etc.) of length n
  – Unigram, Bigram and Trigrams for n = 1, 2, and 3

• N-Gram Model
  – probability distribution of n-unit sequences
  – Markov chain of order n -1
    • the probability of a unit depends only on some of the immediately preceding units
N-gram character models

- $P(c_{1:n})$ is the probability of a sequence of $N$ characters $c_1$ through $c_N$
  - Typically corpus-based (uses a body of text)
  - $P(\text{“the”}) = .03$
  - $P(\text{“zgq”}) = .000000000002$

- Application: language identification
  - Corpus: $P(\text{Text} | \text{Language})$ (trigrams)
  - Language Identification – use Bayes Rule!

- Application: named–entity recognition
  - “ex “ -> drug name
  - Can handle unseen words!
Smoothing

• What do we do about zero (or low) counts in a training corpus?
  – Sequences with count zero are assigned a small non-zero probability (support generalization)
  – Need to adjust other counts downward, so probability still sums to 1
• Add one smoothing \( \frac{1}{(n+2)} \)
• Backoff (e.g. if no trigram, use bigram)
• Many others in NLP course
• Just like ML, is it better to improve smoothing methods, or to get more data???
Evaluation

• Just like ML, cross-validation with train/validate/test data
• Just like ML, many metrics
  – extrinsic – e.g. language identification
  – intrinsic - perplexity
N-gram *word* models

- Much larger “vocabulary” of units
- Since units are open, out of vocabulary becomes a problem
- “Word” needs to be defined precisely
- Common in speech recognition
Text Classification

• Our spam filter from probability chapters (now think language modeling), can also be recast as supervised learning
  – Input: text
  – Output: one of a set of predefined classes
  – Features: NLP-based (e.g. word and character n-grams)
    • Bag of words: unigrams
    • Feature selection
Information Retrieval

• Corpus of “documents”
• Queries in a language
• Result set (relevant documents)
• Presentation of result set

• Applications: Libraries, Search engines
IR Scoring Functions

• An alternative to boolean models (relevant or not), that assigns a numeric score
  – Useful for ranking in presentation

• BM25 function – linear weighted combination of score for each term in the query
  – TF (term frequency)
  – IDF (inverse document frequency of the term)
  – Document length
IR System Evaluation

<table>
<thead>
<tr>
<th></th>
<th>In result set</th>
<th>Not in result set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>Not relevant</td>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>

• Precision
  – The proportion of documents in the result set that are indeed relevant (3/4)

• Recall
  – The proportion of relevant documents that are in the result set (3/5)
  – Hard for www

• Also useful for evaluating supervised ML
IR Refinements

• Beyond words, via NLP
  – Stemming (couch = couches)
  – Semantics (couch = sofa)
  – Usually helps recall at expense of precision

• Google’s PageRank and HITS – web oriented

• Question Answering – “towards” NLP (local research)
  – Web IR for open domain
  – Fall 2010 AI Magazine
  – E.g., CYC, IBM’s jeopardy program
  – Again, tradeoff between deeper algorithms (here NLP) versus just more data
Information Extraction

• “Skimming” a text and looking for occurrences of a particular class of object and relationships among objects
Finite-State Automata

• FSAs for attribute-based extraction
  – price

• Cascaded FSTs for relational extraction
  – Multiple attributes and their relations

• Good for restricted, formulaic domains (WSJ merger reports)
Probabilistic (not rule-based) Models

• HMMs (chapter 15) for noisy and/or varied texts
  – generative (but don’t need)

• CRFs
  – discriminitive
Corpus-Based Ontology Extraction

• Acquiring a KB, in contrast to finding the speaker in a talk announcement

• IS-A hierarchy constructed from high precision query templates
  – NounPhrase such as NounPhrase
  – Forces such as gravity and *

• Automated template construction
• Both sensitive to noise propagation
Machine Reading

• Rather than bootstrapping, towards no human input of any kind
  – NELL: Never-Ending Language Learning
  – http://rtw.ml.cmu.edu/rtw/
    • Read the Web" is a research project that attempts to create a computer system that learns over time to read the web. Since January 2010, our computer system called NELL (Never-Ending Language Learner) has been running continuously, attempting to perform two tasks each day:
      • First, it attempts to "read," or extract facts from text found in hundreds of millions of web pages (e.g., playsInstrument(George_Harrison, guitar)).
      • Second, it attempts to improve its reading competence, so that tomorrow it can extract more facts from the web, more accurately.