Search and Data Mining: Techniques

Text Mining

Anya Yarygina

Boris Novikov

Introduction

- Generally used to denote any system that analyzes large quantities of natural language text and detects lexical or linguistic usage patterns in an attempt to extract probably useful (although only probably correct) information
- Terms disambiguation
  - Text mining
  - Data mining
  - Natural language processing

Outline

- Introduction
- Mining plain text
  - Extracting information for human consumption
  - Assessing document similarity
  - Extracting structured information
- Techniques
  - Collocations
  - Statistical inference
  - Word sense disambiguation
  - Part-of-speech tagging
- Tools

Reference

- Cristopher D. Manning, Hinrich Schutze. Foundations of statistical natural language processing. The MIT Press, 1999

Miming Plain Text

- Extracting information for human consumption
  - Text summarization
  - Document retrieval
  - Information retrieval
- Assessing document similarity
  - Text categorization
  - Document clustering
  - Language identification
  - Ascribing authorship
  - Identifying key-phrases
- Extracting structured information
  - Entity extraction
  - Information extraction

Entity Extraction

- Named entities
  - Names of people, places, organizations, products
- E-mail addresses, URLs
- Dates, numbers, sums of money
- Acronyms and their definition
- Multiword terms
- Dictionary-based approach
- Capitalization and punctuation pattern
- Regular expression
- Explicit grammars
- Heuristics
- Machine learning

Information Extraction

- Events with attributes
  - Entity extraction
  - Relationship extraction
  - Co-reference ambiguity
- Syntactic parsing of the text
- Small finite-state grammars
- Machine learning

Looking at Text

- Low-level formatting issues
  - Junk formatting/content
  - Uppercase and lowercase
- Tokenization: what is a word?
  - Graphic word
  - Whitespace
  - Problems
    - Periods
    - Single apostrophes
    - Hyphenation
    - Homographs
- Morphology
  - Stemming
  - Lemmatization
- Sentences

Collocations

- Most frequently occurring bi-grams
  - Part-of-speech filter
- Word collocation window
  - Based on mean and variance of the offsets
  - Filter out flat peaks
- Hypothesis testing
  - $P(w_1 w_2) = P(w_1)P(w_2)$
  - t test $t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}$
  - Pearson's chi-squared test $X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$
  - Likelihood ratios
- Mutual information

$$I(x', y') = \log_2 \frac{P(x'y')}{P(x')P(y')}$$

Statistical Inference: n-gram models

- $P(w_n|w_1,\ldots w_{n-1})$
- Markov assumption:
  Only the prior local context affects the word
- Statistical estimators
  $P(w_n|w_1,\ldots w_{n-1}) = P(w_1,\ldots w_n)/P(w_1,\ldots w_{n-1})$
- Combining estimators
  - Simple linear interpolation

$$P_{\text{li}}(w_n|w_{n-2},w_{n-1}) = \lambda_1 P_1(w_n) + \lambda_2 P_2(w_n|w_{n-1}) + \lambda_3 P_3(w_n|w_{n-1},w_{n-2})$$

  - Katz back-off

$$P_{\text{bo}}(w_i|w_{i-n+1}\cdots w_{i-1}) = \begin{cases} (1-d_{w_{i-n+1}\cdots w_{i-1}})\frac{C(w_{i-n+1}\cdots w_i)}{C(w_{i-n+1}\cdots w_{i-1})} \\ \qquad \text{if } C(w_{i-n+1}\cdots w_i) > k \\ \alpha_{w_{i-n+1}\cdots w_{i-1}} P_{\text{bo}}(w_i|w_{i-n+2}\cdots w_{i-1}) \\ \qquad \text{otherwise} \end{cases}$$

  - General linear interpolation

$$P_{\text{li}}(w|h) = \sum_{i=1}^{k} \lambda_i(h) P_i(w|h)$$

  - Witten-Bell smoothing

$$P_{\text{WB}}(w_i|w_{(i-n+1)(i-1)}) = \lambda_{w_{(i-n+1)(i-1)}} P_{\text{MLE}}(w_i|w_{(i-n+1)(i-1)}) + (1-\lambda_{w_{(i-n+1)(i-1)}}) P_{\text{WB}}(w_i|w_{(i-n+2)(i-1)})$$

$$(1-\lambda_{w_{(i-n+1)(i-1)}}) = \frac{|\{w_i : C(w_{i-n+1}\cdots w_i) > 0\}|}{|\{w_i : C(w_{i-n+1}\cdots w_i) > 0\}| + \sum_{w_i} C(w_{i-n+1}\cdots w_i)}$$

- Maximum likelihood estimate
  - $P(w_1,\ldots,w_n) = \frac{C(w_1,\ldots w_n)}{N}$
  - $P(w_1,\ldots,w_n) = \frac{C(w_1,\ldots w_n)}{C(w_1,\ldots w_{n-1})}$
- Laplace law
  - $P(w_1,\ldots,w_n) = \frac{C(w_1,\ldots w_n)+1}{N+B}$
- Lidstone law
  - $P(w_1,\ldots,w_n) = \frac{C(w_1,\ldots w_n)+\lambda}{N+B\lambda}$
- Held out estimation
  - $T_r = \sum_{\{w_1,\ldots,w_n : C_1(w_1,\ldots,w_n)=r\}} C_2(w_1,\ldots,w_n)$
  - $P(w_1,\ldots,w_n) = \frac{T_r}{TN_r}$, where $C(w_1,\ldots w_n) = r$
  - $P(w_1,\ldots,w_n) = \frac{T_r^{01}}{NN_r^0}$
- Deleted estimation
  - $P(w_1,\ldots,w_n) = \frac{T_r^{10}+T_r^{01}}{N(N_r^0+N_r^0)}$

Word Sense Disambiguation

- Supervised disambiguation
  - Bayesian classification
  - Information-theoretic approach
- Unsupervised disambiguation
  - EM algorithm for learning a word sense clustering
  - Constraint-based
  - Resource-based
    - Dictionary-based
    - Thesaurus-based

Word Sense Disambiguation

- Bayesian classification
  - use information from words in the context window to help in the disambiguation decision
  - Bayes decision rule

$$P(s_k|c) = \frac{P(c|s_k)}{P(c)} P(s_k)$$

  - Naïve Bayes assumption

$$P(c|s_k) = \prod_{v_j in\ c} P(v_j|s_k)$$

  - Maximum-likelihood estimation

$$P(v_j|s_k) = \frac{C(v_j, s_k)}{\sum_t C(v_t, s_k)}$$

$$P(s_k) = \frac{C(s_k)}{C(w)}$$

- Information-theoretic approach
  - find a single contextual feature that reliably indicates which sense of the ambiguous word is being used
  - Mutual information

$$I(X;Y) = \Sigma_{x \in X} \Sigma_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

  - Flip-flop algorithm

find random partion $P = \{P_1, P_2\}$ of $t_1, ..., t_m$
while (improving $I(P;Q)$) do
  find partion $Q = \{Q_1, Q_2\}$ of $x_1, ..., x_n$
    that maximizes $I(P;Q)$
  find partion $\{P_1, P_2\}$ of $t_1, ..., t_m$
    that maximizes $I(P;Q)$
end

Word Sense Disambiguation

- Disambiguation based on sense definitions

$$\text{score}(s_k) = \text{overlap}(\ D_k, \cup_{v_j} \text{ in }_c E_{v_j}\ )$$

- Thesaurus-based disambiguation
  - Walker approach

$$\text{score}(s_k) = \Sigma_{w_j} \text{ in }_c \delta(t(s_k), w_j)$$

  - Yarowsky approach

$$\text{score}(s_k) = \log \text{P}(t(s_k)) + \Sigma_{w_j} \text{ in }_c \log P(w_j \mid t(s_k))$$

Word Sense Disambiguation

- One sense per discourse:

The sense of a target word is highly consistent within any given document.

- One sense per collocation:

Nearby words provide strong and consistent clues to the sense of a target word, conditional on relative distance, order, and syntactic relationship.

Word Sense Disambiguation

- EM algorithm for learning a word sense clustering
  - Parameters of the model
    $P(w_j \mid s_k),\ 1 \le j \le J, 1 \le k \le K$
    $P(s_k),\ 1 \le k \le K.$
  - Log-likelihood of the corpus C

    $\Sigma_{i=1}^{I} \boldsymbol{log}\ \Sigma_{k=1}^{K} P(c_i \mid s_k) P(s_k)$
  - Naïve Bayes assumption

    $P(c_i \mid s_k) = \Pi_{w_j \ \mathbf{in}\ c_i} P(w_j \mid s_k)$

- Initialize parameters of the model randomly
- E-step

$$h_{ik} = \frac{P(c_i|s_k)}{\Sigma_{k=1}^{K} P(c_i|s_k)}$$

- M-step

$$P(w_j \mid s_k) = \frac{\Sigma_{\{c_i \mid w_j \ \mathbf{in}\ c_i\}} h_{ik}}{Z_k}$$

$$P(s_k) = \frac{\Sigma_{i=1}^{I} h_{ik}}{\Sigma_{k=1}^{K} \Sigma_{i=1}^{I} h_{ik}} = \frac{\Sigma_{i=1}^{I} h_{ik}}{I}$$

$$Z_k = \Sigma_{k=1}^{K} \Sigma_{\{c_i \mid w_j \ \mathbf{in}\ c_i\}} h_{ik}$$

Part-of-Speech Tagging

- Markov model taggers
  - Probabilistic model
  - Viterbi algorithm
- Transformation-based learning of tags

Part-of-Speech Tagging

Markov model taggers

- Probabilistic model
  - Limited horizon
    $P(X_{i+1} = t^j | X_1, \dots, X_i) = P(X_{i+1} = t^j | X_i)$
  - Time invariant
    $P(X_{i+1} = t^j | X_i) = P(X_2 = t^j | X_2)$
- Optimal tags for a sentence
  - Bayes rule
    $\arg\max P(t_{1,n} | w_{1,n}) = \arg\max P(w_{1,n} | t_{1,n}) P(t_{1,n})$
  - Words are independent on each other
  - Word only depends on its tag

$$\arg\max P(w_{1,n} | t_{1,n}) P(t_{1,n}) = \arg\max \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

- Probability estimations

$$P(t^k | t^j) = \frac{C(t^j, t^k)}{C(t^j)}$$

$$P(w^l | t^j) = \frac{C(w^l : t^j)}{C(t^j)}$$

Viterbi algorithm

- Functions
  - $\delta_i(j)$ probability of being in state j (tag j) at word i
  - $\varphi_{i+1}(j)$ the most likely state (tag) at word i given that we are in state j at word i+1
- Initialization
- Induction

$$\delta_{i+1}(t^j) = \max[\delta_i(t^k) P(t^j | t^k) P(w_{i+1} | t^j)]$$

$$\varphi_{i+1}(t^j) = \arg\max[\delta_i(t^k) P(t^j | t^k) P(w_{i+1} | t^j)]$$

- Termination and path-readout

$$X_{n+1} = \arg\max \delta_{n+1}(t^j)$$

$$X_{i+1} = \varphi_{i+1}(X_{i+1})$$

Part-of-Speech Tagging

- Transformation-based learning of tags
  - Replace tag t_1 with t_2
  - Extract rules while tagging error decreases

Tools

- OpenNLP
  - Sentence detector
  - Tokenizer
  - Name finder
  - Document categorizer
  - Part of speech tagger
  - Chunker
  - Parser
  - Co-reference resolution
- GATE
  - Tokenizer
  - Gazetteer
  - Sentence splitter
  - Part of speech tagger
  - Named entities transducer
  - Co-reference tagger
- cTAKES
  - Sentence boundary detector
  - Rule-based and context dependent tokenizer
  - Normalizer
  - Part-of-speech tagger
  - Phrasal chunker
  - Dictionary lookup annotator
  - Context annotator
  - Negation detector
  - Dependency parser
- UIMA
  - Component interfaces in an analytics pipeline
  - Set of design patterns
  - Data representations
    - in-memory representation of annotations for high-performance analytics
    - XML representation of annotations for integration with remote web services
  - Development roles allowing tools to be used by users with diverse skills