NATURAL LANGUAGE PROCESSING

CHAPTER 22

Natural Language Processing

- Intelligent Agents
  - Knowledge Acquisition
- Knowledge Acquisition Tasks
  - Text Classification
  - Information Retrieval
  - Information Extraction
- Language Models

Language Models

- Language Models
  - Predicts the probability distribution of language expressions
  - Formal languages: Grammar, Semantics
  - Natural Languages
    - Ambiguous
    - No definite set of sentences
    - Probability distribution over sentences
    - Models are approximations

Language Models

- N-gram Character Models
  - Text / Sentence : characters, digits, symbols..
  - Probability dist. over sequence of characters
  - N-gram
    - A sequence of characters of length N
  - N-gram Model
    - Model of prob. Dist. Of n-characters
    - Defined as Markov chain of order n-1
    - Depends only on the immediately preceding characters

Language Models

- N-gram Character Models
  - Tri-gram Model (Markov chain order-2)
    - Is given by: $P(c_i \mid c_{1:i-1}) = P(c_i \mid c_{i-2:i-1})$
    - Applying Markov Chain rule and Markov assumption

$$P(c_{1:N}) = \prod_{i=1}^{N} P(c_i \mid c_{1:i-1}) = \prod_{i=1}^{N} P(c_i \mid c_{i-2:i-1})$$

    - Corpus

Language Models

- N-gram Models for Language Identification
  - Build a Tri-gram Model with each language
    - Represented by: $P(c_i \mid c_{i-2:i-1}, \ell)$
    - Results in a model of P(Text|Language)
    - To find Most probable language apply Bayes and Markov assumption

$$\begin{aligned} \ell^* &= \operatorname*{argmax}_{\ell} P(\ell \mid c_{1:N}) \\ &= \operatorname*{argmax}_{\ell} P(\ell) P(c_{1:N} \mid \ell) \\ &= \operatorname*{argmax}_{\ell} P(\ell) \prod_{i=1}^{N} P(c_i \mid c_{i-2:i-1}, \ell) \end{aligned}$$

- Applications: Genre classification, named entity recognition, Spelling correction

Language Models

- Smoothing N-gram Models
  - Provides only estimate of true prob. Distribution
  - Smoothing
    - Process of adjusting the probability of low frequency counts
    - Assign a small non-zero probability
    - Laplace Smoothing: P(X=True) →1/(n+2)
  - Back-off Model
    - Back-off n-1 grams
    - Linear Interpolation Smoothing

$$P(c_i|c_{i-2:i-1}) = \lambda_3 P(c_i|c_{i-2:i-1}) + \lambda_2 P(c_i|c_{i-1}) + \lambda_1 P(c_i)$$

Language Models

- Model Evaluation
  - Training corpus and validation corpus
  - Perplexity
    - Is a measure of probability of sequence

$$Perplexity(c_{1:N}) = P(c_{1:N})^{-\frac{1}{N}}$$

    - Reciprocal of probability normalized by sequence length
- N-gram Word Models
  - Larger Vocabulary
  - How to deal with Out-of-Vocabulary words?

Text Classification

- Text Classification (Categorization)
  - Language Identification, genre classification, Sentiment analysis, Spam Detection
  - Spam Detection: A Supervised Learning Approach
    - Spam and Ham
    - Both Word model and Character model

Spam: Wholesale Fashion Watches -57% today. Designer watches for cheap ...
Spam: You can buy ViagraFr$1.85 All Medications at unbeatable prices! ...
Spam: WE CAN TREAT ANYTHING YOU SUFFER FROM JUST TRUST US ...
Spam: Sta.rt earn*ing the salary yo,u d-eserve by o'btaining the prope,r crede'ntials!

Ham: The practical significance of hypertree width in identifying more ...
Ham: Abstract: We will motivate the problem of social identity clustering: ...
Ham: Good to see you my friend. Hey Peter, It was good to hear from you. ...
Ham: PDS implies convexity of the resulting optimization problem (Kernel Ridge ...

Text Classification

- Text Classification Approach
  - Define one n-gram language model for Spam, P(Message|Spam) and Ham P(Message | Ham) each
  - Classify a new message

$$\underset{c \in \{spam, ham\}}{\operatorname{argmax}} P(c \mid message) = \underset{c \in \{spam, ham\}}{\operatorname{argmax}} P(message \mid c)\, P(c)$$

  - Bag of words
  - Feature Selection
  - Classification using data compression
    - LZW compression algorithms
    - Better compression is the predicted class

Information Retrieval

- Information Retrieval
  - Finding relevant document based on user request
  - IR system has
    - Corpus of documents
      - Page, Multiple pages, paragraph
    - Queries posed
      - Format in which query is given
    - Result Set
      - Subset produced by IR based on query
    - Presentation of the result set
      - Order in which it is displayed
- Boolean Keyword model

Information Retrieval

- IR Scoring Fucntions
  - Document+Query → Numeric Score
  - Linear weighted combination of scores
  - Factors affecting Weight:
    - Term Frequency (TF)
    - Inverse document frequency(IDF)
    - Length of the document
  - BM25 Function
    - Has index of N documents with look-up TF(qi, dj) and Document frequency count DF(qi)

$$BM25(d_j, q_{1:N}) = \sum_{i=1}^{N} IDF(q_i) \cdot \frac{TF(q_i, d_j) \cdot (k+1)}{TF(q_i, d_j) + k \cdot (1 - b + b \cdot \frac{|d_j|}{L})}$$

Information Retrieval

- IR System Evaluation
  - Precision
    - Proportion of documents in the result set that are relevant
  - Recall
    - Proportion of all the relevant documents in the collection that are in the result set

Information Retrieval

- IR refinements
  - Case folding
  - Stemming
  - Synonyms

Information Retrieval

- Page Rank Algorithm
  - TF score problem
  - Pages with in-link is ranked higher
  - In-link defines the quality of the linked-to page
  - Count of in-links : scope for spammer
  - PR algo: Weight links from high quality sites
  - PR is given by:

$$PR(p) = \frac{1-d}{N} + d\sum_{i} \frac{PR(in_i)}{C(in_i)}$$

  - Random Surfer Model

Information Retrieval

- HITS (Hyperlink Induced Topic Search) Algorithm
  - Query dependent, link analysis algorithm
  - Intersection of HIT lists of query
    - Set of pages relevant to the query
    - Pages link-to or link-from the original set
  - Authority: degree of other pages pointing to it
  - Hub : degree it points to authoritative pages
  - Normalize the score recursively for convergence

Information Retrieval

- HITS (Hyperlink Induced Topic Search) Algorithm

function HITS(*query*) **returns** *pages* with hub and authority numbers

```
pages ← EXPAND-PAGES(RELEVANT-PAGES(query))
for each p in pages do
    p.AUTHORITY ← 1
    p.HUB ← 1
repeat until convergence do
    for each p in pages do
        p.AUTHORITY ← Σ_i INLINK_i(p).HUB
        p.HUB ← Σ_i OUTLINK_i(p).AUTHORITY
    NORMALIZE(pages)
return pages
```

Figure 22.1 The HITS algorithm for computing hubs and authorities with respect to a query. RELEVANT-PAGES fetches the pages that match the query, and EXPAND-PAGES adds in every page that links to or is linked from one of the relevant pages. NORMALIZE divides each page's score by the sum of the squares of all pages' scores (separately for both the authority and hubs scores).

Information Extraction

- Information Extraction
  - Process of acquiring knowledge
    - Skimming a text
    - Look for occurrence of class and their related objects
  - Accuracy
    - High : Domain specific
    - Less: Generalised domains

Information Extraction

- Finite State Automata for Information Extraction
  - Attribute based extraction
    - Entire text : Object
    - Extract its attributes
  - Eg. text: “IBM Thinkbook 970. Our price $199”
  - Attribute List: {Manufacturer: IBM, Model: Thinkbook 970, Price:$199)
  - Identify a pattern (template) for each attribute to be extracted (Regex)
    - Template Structure: Prefix regex, target regex and postfix regex
  - Attribute matching with text
    - Exactly one match, No match, Multiple matches: On priority

Information Extraction

- Finite State Automata for Information Extraction
  - Relational Extraction Systems
    - Multiple objects and their relationship
    - Built as a series of small, efficient FSAs (Cascaded FS Transducers)
  - Eg:FASTUS: handles news stories and extract relations

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.

$$e \in JointVentures \wedge Product(e, \text{“}golf\ clubs\text{”}) \wedge Date(e, \text{“}Friday\text{”}) \wedge Member(e, \text{“}Bridgestone\ Sports\ Co\text{”}) \wedge Member(e, \text{“}a\ local\ concern\text{”}) \wedge Member(e, \text{“}a\ Japanese\ trading\ house\text{”})\ .$$

- Fc
  - Transduce to different format
  - Forward to next automata

Information Extraction

- Finite State Automata for Information Extraction
  - Stages of FASTUS
    - Tokenization
    - Complex word handling
      - Lexical entries and FS grammar rules
    - Basic group handling
      - Noun group, verb groups, preposition and conjunction
    - Complex phrase handling
      - FS rules that are processed quickly and produce unambiguous o/ps
      - Deals with domain specific events
    - Structure merging
      - Merges multiple instances to same lexical entries in a specific domain

Information Extraction

- Probabilistic Models Information Extraction
  - Hidden Markov model for information extraction
    - Hidden states: Prefix, Target, Postfix of the attribute template
    - Observations : Words of the text
  - Two HMMs

Text:	There	will	be	a	seminar	by	Dr.	Andrew	McCallum	on	Friday
Speaker:	-	-	-	-	PRE	PRE	TARGET	TARGET	TARGET	POST	-
Date:	-	-	-	-	-	-	-	-	-	PRE	TARGET

  - Pros
    - Noise tolerance & Template not required
  - Most likely path: Apply each attribute HMM separately / combine all attributes into a single HMM

Information Extraction

- Probabilistic Models Information Extraction

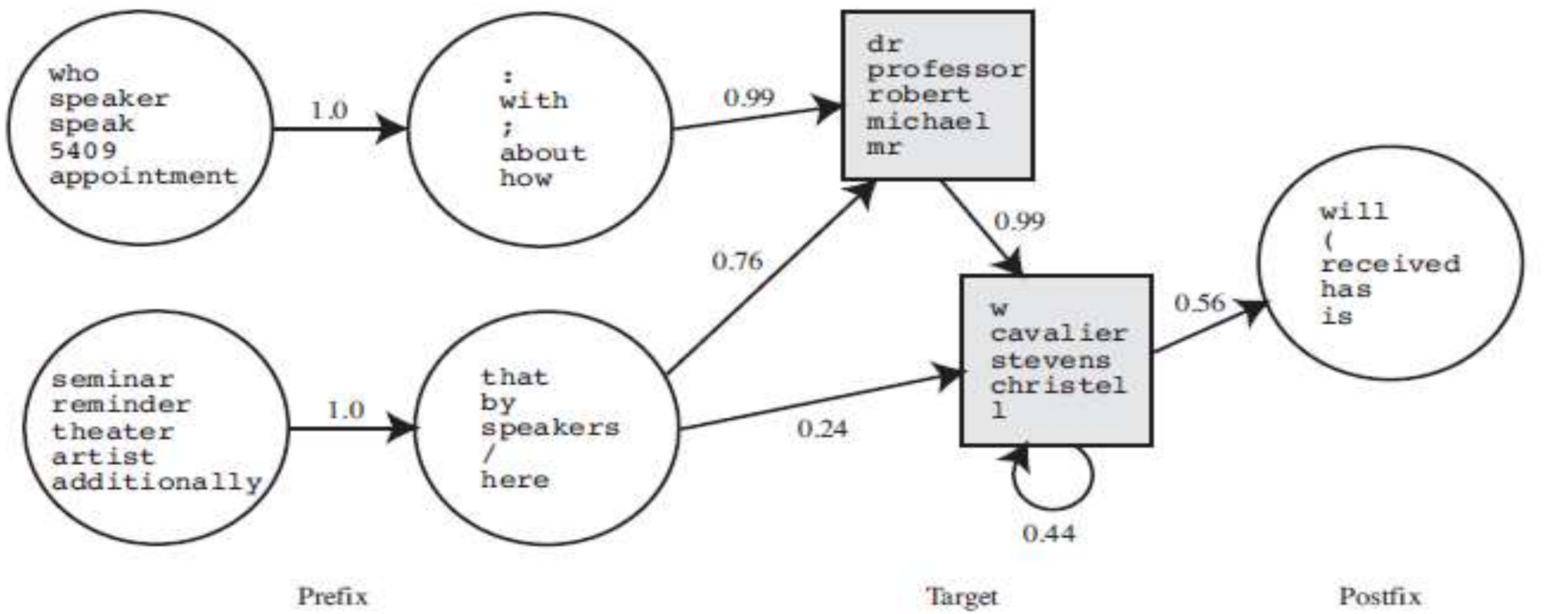

Figure 22.2 Hidden Markov model for the **speaker** of a talk announcement. The two square states are the target (note the second target state has a self-loop, so the target can match a string of any length), the four circles to the left are the prefix, and the one on the right is the postfix. For each state, only a few of the high-probability words are shown. From Freitag and McCallum (2000).

Information Extraction

- Conditional random Fields for Information Extraction
  - HMM: generative model
  - Need a discriminative model
    - Models the CP of the hidden attributes given the observations
    - Given text e1:N, the conditional model finds the hidden state sequence X1:N that maximizes P(X1:N | e1:N)
    - Linear chain CRF : models temporal sequence and defines a CPD
    - Feature functions : key comp

$$F(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{e}, i) = \sum_k \lambda_k f_k(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{e}, i)$$

$$\mathbf{P}(\mathbf{x}_{1:N} | \mathbf{e}_{1:N}) = \alpha\, e^{\left[\sum_{i=1}^{N} F(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{e}, i)\right]}$$

$$f_1(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{e}, i) = \begin{cases} 1 & \text{if } \mathbf{x}_i = \text{SPEAKER and } \mathbf{e}_i = \text{ANDREW} \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{e}, i) = \begin{cases} 1 & \text{if } \mathbf{x}_i = \text{SPEAKER and } \mathbf{e}_{i+1} = \text{SAID} \\ 0 & \text{otherwise} \end{cases}$$

Information Extraction

- Ontology Extraction from Large Corpora
  - Building a large KB from a corpus
  - Differs from other approaches
    - Open ended, Precision, Aggregated results
  - Generalized template focussing on high precision and low recall

$$NP \text{ such as } NP\ (, NP)^{*}\ (,)?\ ((\text{and} \mid \text{or})\ NP)?$$

Information Extraction

- Automated Template Construction
  - Learn templates from few examples and apply it recursively

("Isaac Asimov", "The Robots of Dawn")
("David Brin", "Startide Rising")
("James Gleick", "Chaos—Making a New Science")
("Charles Dickens", "Great Expectations")
("William Shakespeare", "The Comedy of Errors")

(Author, Title, Order, Prefix, Middle, Postfix, URL)

  - Pros & Cons
    - Given good set of templates, system can collect good set of examples and vice-versa
    - If incorrect template is provided , error will propagate

Information Extraction

- Machine Reading
  - No human intervention
  - Eg:TEXTRUNNER

Type	Template	Example	Frequency
Verb	NP_1 *Verb* NP_2	X established Y	38%
Noun–Prep	NP_1 *NP Prep* NP_2	X settlement with Y	23%
Verb–Prep	NP_1 *Verb Prep* NP_2	X moved to Y	16%
Infinitive	NP_1 **to** *Verb* NP_2	X plans to acquire Y	9%
Modifier	NP_1 *Verb* NP_2 *Noun*	X is Y winner	5%
Noun-Coordinate	NP_1 (, | **and** | **-** | :) NP_2 *NP*	X-Y deal	2%
Verb-Coordinate	NP_1 (,| **and**) NP_2 *Verb*	X, Y merge	1%
Appositive	NP_1 *NP* (:| ,)? NP_2	X hometown : Y	1%

Figure 22.3 Eight general templates that cover about 95% of the ways that relations are expressed in English.

THANK YOU