Text Clustering

Clustering

• Partition unlabeled examples into disjoint subsets of clusters, such that:
  – Examples within a cluster are very similar
  – Examples in different clusters are very different
• Discover new categories in an unsupervised manner (no sample category labels provided).

Clustering Example
Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (dendrogram) from a set of unlabeled examples.
- Recursive application of a standard clustering algorithm can produce a hierarchical clustering.

Aglomerative vs. Divisive Clustering

- Agglomerative (bottom-up) methods start with each example in its own cluster and iteratively combine them to form larger and larger clusters.
- Divisive (partitional, top-down) separate all examples immediately into clusters.

Hierarchical Agglomerative Clustering (HAC)

- Assumes a similarity function for determining the similarity of two instances.
- Starts with all instances in a separate cluster and then repeatedly joins the two clusters that are most similar until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.
HAC Algorithm

Start with all instances in their own cluster.  
Until there is only one cluster:  
Among the current clusters, determine the two clusters, \( c_i \) and \( c_j \), that are most similar.  
Replace \( c_i \) and \( c_j \) with a single cluster \( c_i \cup c_j \).

Cluster Similarity

- Assume a similarity function that determines the similarity of two instances: \( \text{sim}(x, y) \).  
  - Cosine similarity of document vectors.
- How to compute similarity of two clusters each possibly containing multiple instances?  
  - Single Link: Similarity of two most similar members.  
  - Complete Link: Similarity of two least similar members.  
  - Group Average: Average similarity between members.

Non-Hierarchical Clustering

- Typically must provide the number of desired clusters, \( k \).  
- Randomly choose \( k \) instances as seeds, one per cluster.  
- Form initial clusters based on these seeds.  
- Iterate, repeatedly reallocating instances to different clusters to improve the overall clustering.  
- Stop when clustering converges or after a fixed number of iterations.
K-Means

- Assumes instances are real-valued vectors.
- Clusters based on *centroids, center of gravity*, or mean of points in a cluster, \( c \):
  \[
  \mu(c) = \frac{1}{|c|} \sum_{x \in c} x
  \]
- Reassignment of instances to clusters is based on distance to the current cluster centroids.

Distance Metrics

- **Euclidian distance** (L_2 norm):
  \[
  L_2(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}
  \]
- **L_1 norm**:
  \[
  L_1(\vec{x}, \vec{y}) = \sum_{i=1}^{n} |x_i - y_i|
  \]
- **Cosine Similarity** (transform to a distance by subtracting from 1):
  \[
  1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}
  \]

K-Means Algorithm

Let \( d \) be the distance measure between instances.
Select \( k \) random instances \( \{s_1, s_2, \ldots, s_k\} \) as seeds.
Until clustering converges or other stopping criterion:

For each instance \( x_i \):
  - Assign \( x_i \) to the cluster \( c_j \) such that \( d(x_i, s_j) \) is minimal.
  - *(Update the seeds to the centroid of each cluster)*
For each cluster \( c_j \):
  \[
  s_j = \mu(c_j)
  \]
**K Means Example**

(K=2)

- Pick seeds
- Reassign clusters
- Compute centroids
- Reassign clusters
- Compute centroids
- Reassign clusters
- Converged!

---

**Information Extraction**

- Identify specific pieces of information (data) in a unstructured or semi-structured textual document.
- Transform unstructured information in a corpus of documents or web pages into a structured database.
- Applied to different types of text:
  - Newspaper articles
  - Web pages
  - Scientific articles
  - Newsgroup messages
  - Classified ads
  - Medical notes
MUC

- DARPA funded significant efforts in IE in the early to mid 1990’s.
- Message Understanding Conference (MUC) was an annual event/competition where results were presented.
- Focused on extracting information from news articles:
  - Terrorist events
  - Industrial joint ventures
  - Company management changes
- Information extraction of particular interest to the intelligence community (CIA, NSA).

Other Applications

- Job postings
- Job resumes
- Seminar announcements
- Company information from the web
- Apartment rental ads
- Molecular biology information from MEDLINE

Sample Job Posting

Subject: US-TN-SOFTWARE PROGRAMMER
Date: 17 Nov 1996 17:37:29 GMT
Organization: Reference.Com Posting Service
Message-ID: <56nigp$mrs@bilbo.reference.com>

SOFTWARE PROGRAMMER

Position available for Software Programmer experienced in generating software for PC-Based Voice Mail systems. Experienced in C Programming. Must be familiar with communicating with and controlling voice cards; preferable Dialogic, however, experience with others such as Rhetorix and Natural Microsystems is okay. Prefer 5 years or more experience with PC Based Voice Mail, but will consider as little as 2 years. Need to find a Senior level person who can come on board and pick up code with very little training. Present Operating System is DOS. May go to OS-2 or UNIX in future.

Please reply to:
Kim Anderson
AdNET
(901) 458-2888 fax
kimander@memphisonline.com
Web Extraction

- Many web pages are generated automatically from an underlying database.
- Therefore, the HTML structure of pages is fairly specific and regular (semi-structured).
- However, output is intended for human consumption, not machine interpretation.
- An IE system for such generated pages allows the website to be viewed as a structured database.
- An extractor for a semi-structured web site is sometimes referred to as a wrapper.
- Process of extracting from such pages is sometimes referred to as screen scraping.

Learning for IE

- Writing accurate patterns for each slot for each domain (e.g. each web site) requires laborious software engineering.
- Alternative is to use machine learning:
  - Build a training set of documents paired with human-produced filled extraction templates.
  - Learn extraction patterns or a neural network to identify the fillers of each slot using an appropriate machine learning algorithm.

Evaluating IE Accuracy

- Always evaluate performance on independent, manually-annotated test data not used during system development.
- Measure for each test document:
  - Total number of correct extractions in the solution template: $N$
  - Total number of slot/value pairs extracted by the system: $E$
  - Number of extracted slot/value pairs that are correct (i.e. in the solution template): $C$
- Compute average value of metrics adapted from IR:
  - Recall = $C/N$
  - Precision = $C/E$
  - F-Measure = Harmonic mean of recall and precision
Semantic Parsing for Question Answering

Semantic Parsing

- **Semantic Parsing**: Transforming natural language (NL) sentences into completely formal **logical forms** or meaning representations (MRs).
- Sample application domains where MRs are directly executable by another computer system to perform some task.
  - Database/knowledge-graph queries
  - Robot command language

Geoquery: A Database Query Application

- Query application for U.S. geography database containing about 800 facts [Zelle & Mooney, 1996]

Which rivers run through the states bordering Texas?

`answer(traverse(next_to(stateid('texas'))))`
Most early work on computational semantics is based on predicate logic.

What is the smallest state by area?

\[ \text{answer}(x_1, \text{smallest}(x_2, (\text{state}(x_1), \text{area}(x_1, x_2)))) \]

\( x_1 \) is a logical variable that denotes “the smallest state by area.”

More recent work uses deep neural nets to directly map “language to code” and generate SQL queries or other programs.

Formal Query Language

Learning Semantic Parsers

Manually programming robust semantic parsers is difficult due to the complexity of the task.

Semantic parsers can be learned automatically from sentences paired with their logical form.

Compositional Semantics

Approach to semantic analysis based on building up an MR compositionally based on the syntactic structure of a sentence.

Build MR recursively bottom-up from the parse tree.

\[
\begin{align*}
\text{BuildMR}(\text{parse-tree}) &= \text{if} \text{parse-tree is a terminal node (word)} \text{then return an atomic lexical meaning for the word.} \\
&\quad \text{else for each child, subtree}_i \text{ of parse-tree create its MR by calling BuildMR(subtree}_i) \text{ return an MR by properly combining the resulting MRs for its children into an MR for the overall parse-tree.}
\end{align*}
\]
Composing MRs from Parse Trees

What is the capital of Ohio?

S  answer(capital(loc_2(stateid('ohio'))))

NP  answer()  VP  capital(loc_2(stateid('ohio')))

What  answer()  is  the  capital  of  Ohio

NP  answer()  VP  capital(loc_2(stateid('ohio')))

Experimental Corpora

- GeoQuery [Zelle & Mooney, 1996]
  - 250 queries for the given U.S. geography database
  - 6.87 words on average in NL sentences
  - 5.32 tokens on average in formal expressions
  - Also translated into Spanish, Turkish, & Japanese.

Experimental Methodology

- Evaluated using standard 10-fold cross validation
- Correctness
  - CLang: output exactly matches the correct representation
  - Geoquery: the resulting query retrieves the same answer as the correct representation
- Metrics
  - Precision = \( \frac{|Correct\ Completed\ Parses|}{|Completed\ Parses|} \)
  - Recall = \( \frac{|Correct\ Completed\ Parses|}{|Sentences|} \)