Information Retrieval

Machine learning for IR

Qianni Zhang

Roadmap of this lecture

- Traditional machine learning
  - Classification
  - Cluster
- Deep learning for IR
  - Extended reading

Classification

Text classification examples

- Assign categories to web pages
  - e.g. sports: football, news:world:asia, finance, etc.
- Find the genre of a given web page
  - e.g. research page, news article, review page, etc.
- Categories may be binary
  - spam, non-spam
  - interesting-to-me, not-interesting-to-me
  - appropriate-for-kids, not-appropriate-for-kids
  - etc...

Classification

Applications

- Word sense disambiguation
  - e.g. bank: financial institution, or river bank?
  - we can view word occurrence contexts as documents, and word senses as categories
  - we have a number of documents put in the correct categories, and try to find the correct word sense for a new incoming word occurrence context
- Hierarchical categorisation of web pages
  - classify pages under a hierarchical catalogue (e.g. Yahoo)
  - the hypertextual nature of web pages is useful (one can take into advantage the links between pages)
  - the hierarchical structure of the categories is also useful

Classification

Applications

- Document organisation
  - e.g. a newspaper that wants to put news stories into categories such as Sports, Finance, World, etc.
- Text filtering
  - classify a stream of incoming documents depending on their relevance to the information consumer
  - typically a binary case (relevant not relevant)
  - common to have a profile for the information consumer
    - the profile can be updated depending on the consumers implicit or explicit relevance assessments on the provided information (adaptive filtering)

Classification

Standing queries

- The path from IR to text classification:
  - You have an information need to monitor
  - You want to rerun an appropriate query periodically to find new news items on this topic
  - You will be sent new documents that are found
    - I.e., it's not ranking but classification (relevant vs. not relevant)
- Such queries are called **standing queries**
  - Long used by "information professionals"
  - A modern mass instantiation is **Google Alerts**
- Standing queries are (hand-written) text classifiers

Classification

Spam filtering: Another text classification task

From: "" <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down
Stop paying rent TODAY !
There is no need to spend hundreds or even thousands for similar courses
I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.
Change your life NOW !
==========
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
==========

Classification

Categorization/Classification

- Given:
  - A representation of a document *d*
    - Issue: how to represent text documents.
    - Usually some type of high-dimensional space – bag of words
  - A fixed set of classes:

    $C = \{c_1, c_2, \ldots, c_J\}$
- Determine:
  - The category of *d*: $\gamma(d) \in C$, where $\gamma(d)$ is a classification function
  - We want to build classification functions ("classifiers").

Classification

Supervised learning

- Naive Bayes (simple, common)
- k-Nearest Neighbors (simple, effective)
- Support-vector machines (reasonably powerful)
- … plus many other methods
- No free lunch: requires hand-classified training data
- But data can be built up (and refined) by amateurs

- Many commercial systems use a mixture of methods

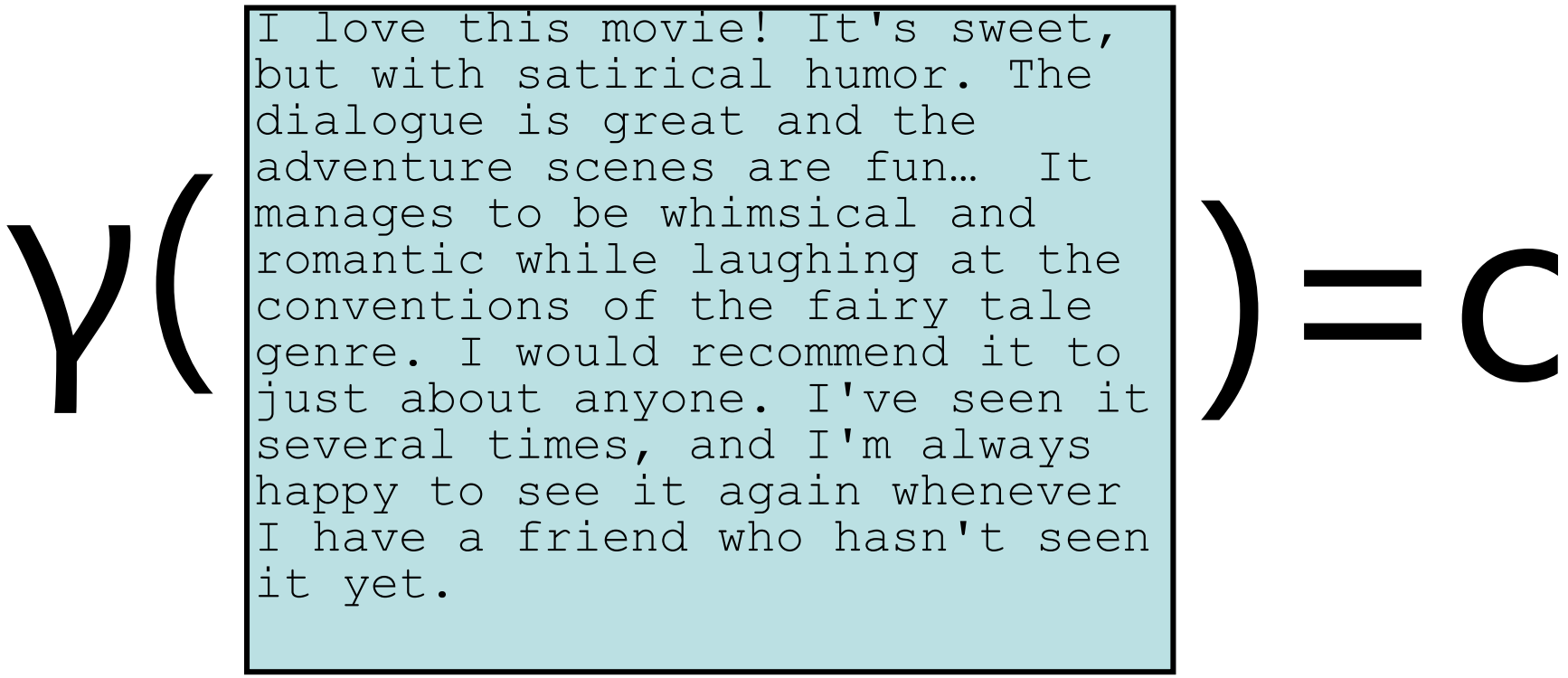
Classification

Main Steps in Classification

- Document indexing - dimensionality reduction
- Choose a classifier
- Train and test the classifier
- Evaluate the effectiveness of the classifier

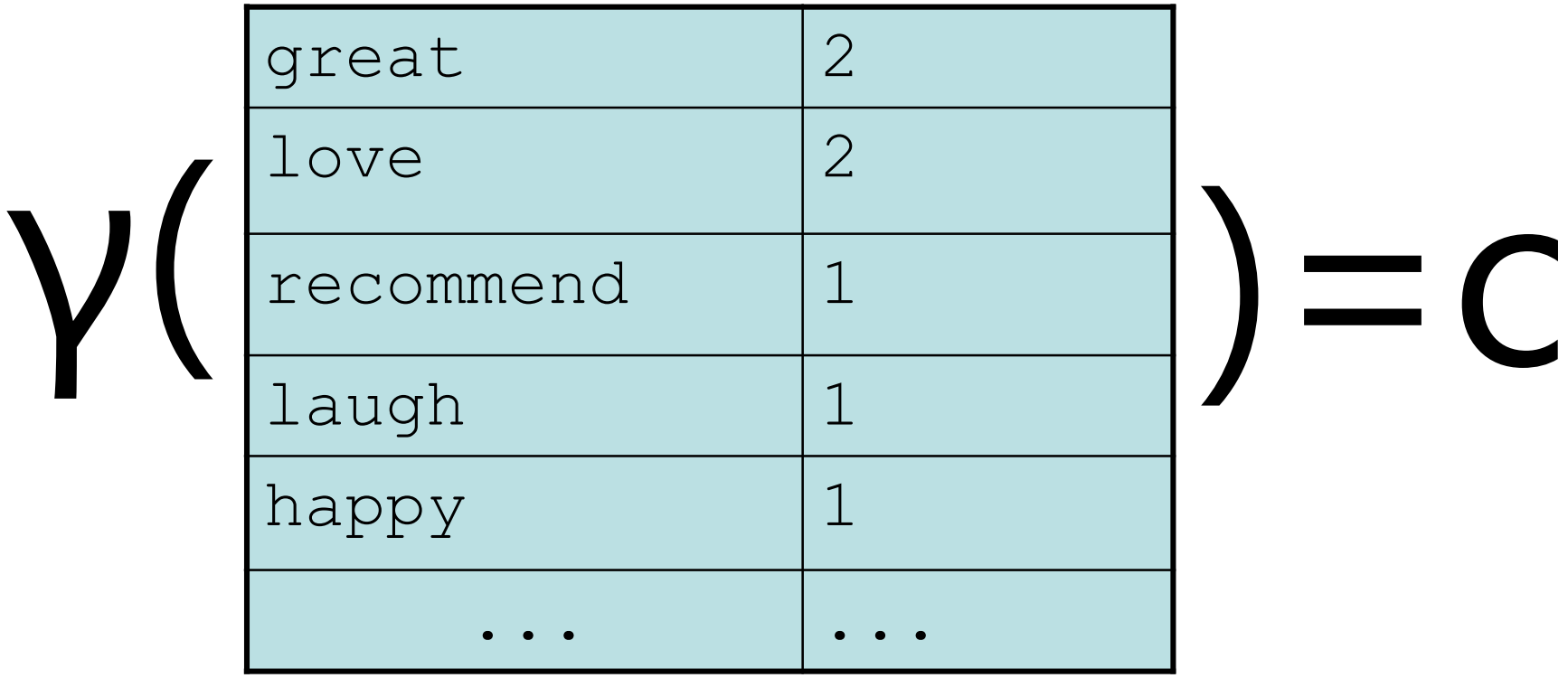
Classification

Features: The bag of words representation



Classification

Features: The bag of words representation



Classification

Features

- Supervised learning classifiers can use any sort of feature
  - URL, email address, punctuation, capitalization, dictionaries, network features
- In the bag of words view of documents
  - We use **only** word features
  - we use **all** of the words in the text (not a subset)

Classification

Feature Selection: Why?

- Dimensionality reduction
- Text collections have a large number of features
  - 10,000 – 1,000,000 unique words … and more
- Selection may make a particular classifier feasible
  - Some classifiers cannot deal with 1,000,000 features
- Reduces training time
  - Training time for some methods is quadratic or worse in the number of features
- Makes runtime models smaller and faster
- Can improve generalization (performance)
  - Eliminates noise features
  - Avoids overfitting

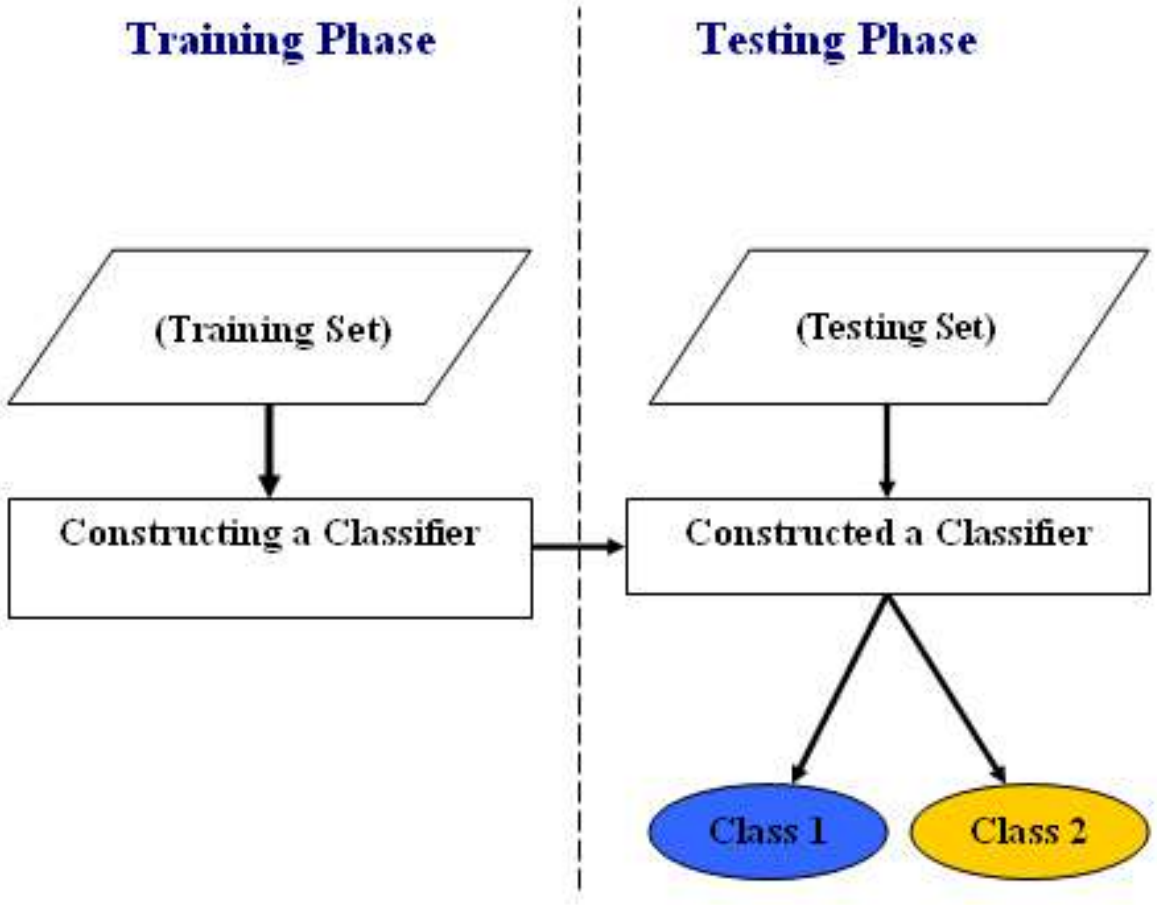
Classification

Feature Selection: Method

- For a given class *c,* we compute a utility measure *A(t, c)* for each term of the vocabulary and select the *k* terms that have the highest values of *A(t, c).* All other terms are discarded and not used in classification
- Frequency: the simplest feature selection method:
  - Just use the commonest terms
  - No particular foundation
  - But it make sense why this works
    - They're the words that can be well-estimated and are most often available as evidence
  - In practice, this is often 90% as good as better methods

Classification

Training and Testing Data



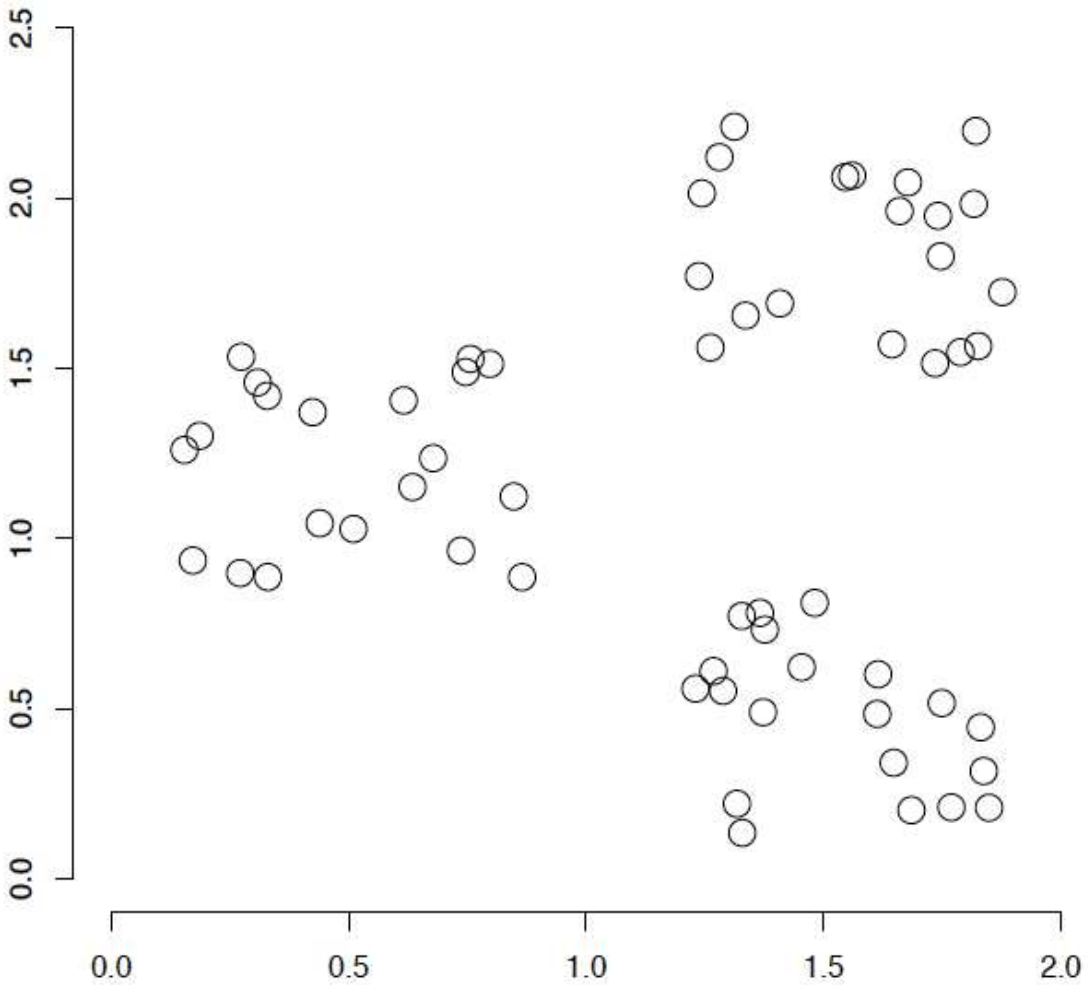
Classification

After a good classifier is designed and trained:

- Performance verified on a validation set
- Embed it in the search engine for your purpose
- Let the search engine run
- Ablation study of search performance, with or without the classification step

Clustering

A data set with clear cluster structure



How would you design an algorithm for finding the three clusters in this case?

Clustering

Clustering vs. Classification

- Classification is supervised and requires a set of labeled training instances for each group (class)
  - Learning with a teacher
- Clustering is unsupervised and learns without a teacher to provide the labeling information of the training data set
  - Unsupervised learning = learning from raw data, as opposed to supervised data where a classification of examples is given
  - A common and important task that finds many applications in IR and other places
  - Also called automatic or unsupervised classification

Clustering

Clustering Algorithms

- Two types of structures produced by clustering algorithms
  - Flat or non-hierarchical clustering
  - Hierarchical clustering
- Flat algorithms
  - Simply consisting of a certain number of clusters and the relation between clusters is often undetermined
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - *K* means clustering
    - (Model based clustering)
  - Measurement: construction error minimization or probabilistic optimization

Clustering

Clustering Algorithms

- Two types of structures produced by clustering algorithms
  - Flat or non-hierarchical clustering
  - Hierarchical clustering
- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive
  - A hierarchy with usual interpretation that each node stands for a sub-cluster of its mother's node
    - The leaves of the tree are the single objects
    - Each node represents the cluster that contains all the objects of its descendants
  - Measurement: similarities of instances

Clustering

Hard vs. soft clustering

- Another important distinction between clustering algorithms is whether they perform soft or hard assignment
- Hard clustering: Each object (or document in the context of IR) is assigned to one and only one cluster
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes
  - You can only do that with a soft clustering approach.

Clustering

Applications of clustering in IR

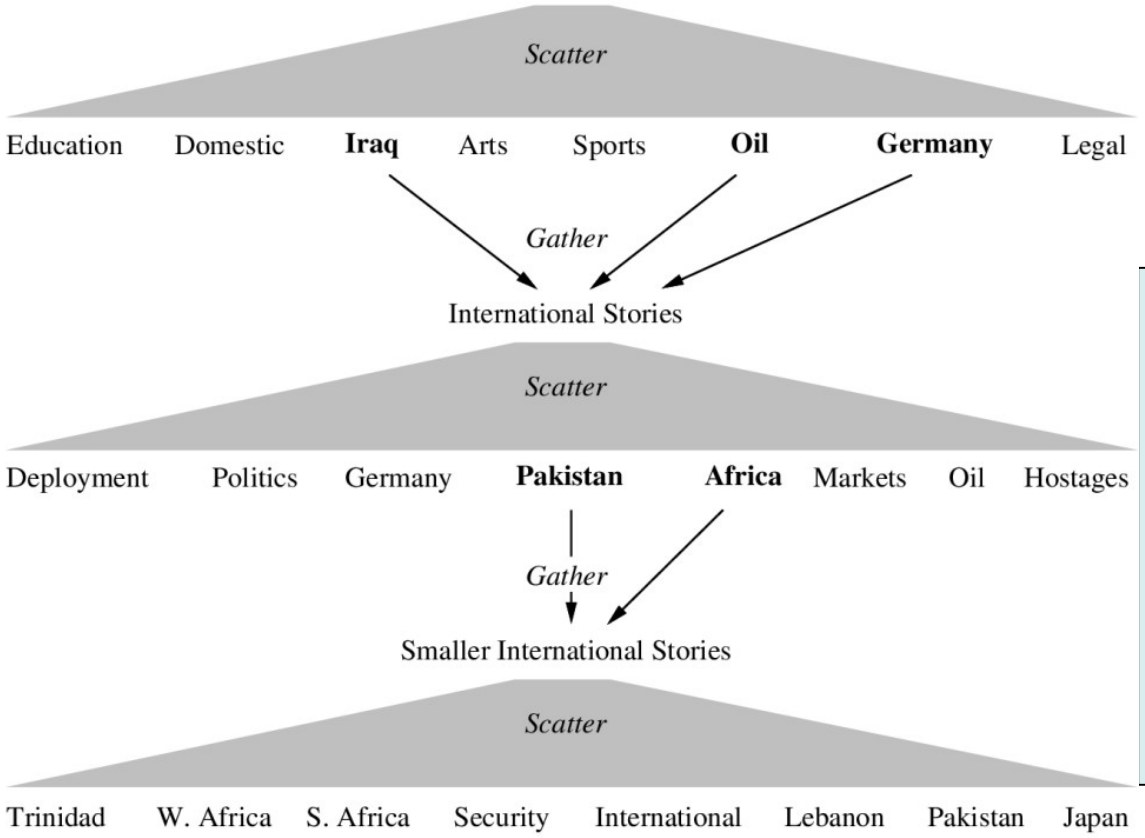
- Cluster Hypothesis (for IR): Documents in the same cluster behave similarly with respect to relevance to information needs
- Possible applications of Clustering in IR
  - Whole corpus analysis/navigation
    - Better user interface: search without typing
  - For improving recall in search applications
    - Better search results (like pseudo RF)
  - For better navigation of search results
    - Effective "user recall" will be higher
  - For speeding up vector space retrieval
    - Cluster-based retrieval gives faster search

These possible applications differ in
•The collection of documents to be clustered
•The aspect of the IR system to be improved

Clustering

Applications of clustering in IR

1. Whole corpus analysis/navigation
   - Better user interface (users prefer browsing over searching since they are unsure about which search terms to use)



The Scatter-Gather user interface. A collection of New York Times news stories is clustered ("scattered") into eight clusters (top row). The user manually *gathers* three of these into a smaller collection *International Stories* and performs another scattering operation. This process repeats until a small cluster with relevant documents is found (e.g., *Trinidad*)

Clustering

Applications of clustering in IR

2. Improve recall in search applications
   - Achieve better search results by
     - Alleviating the term-mismatch (synonym) problem facing the vector space model
       - First, identify an initial set of documents that match the query (i.e., contain some of the query words)
     - Then, add other documents from the same clusters even if they have low similarity to the query
       - Hope: The query "*car*" will also return docs containing *automobile*
       - Because clustering grouped together docs containing *car* with those containing *automobile*.
   - Estimating the collection model of the language modeling (LM) retrieval approach more accurately

Clustering

Applications of clustering in IR

3. Better navigation of search results

- Result set clustering
- Effective “user recall” will be higher

Clustering

Applications of clustering in IR

4. Speed up the search process

- For retrieval models using exhaustive matching (computing the similarity of the query to every document) without efficient inverted index supports

- Solution: cluster-based retrieval
  - First find the clusters that are closest to the query and then only consider documents from these clusters
    - Within this much smaller set, we can compute similarities exhaustively and rank documents in the usual way

Clustering

Issues for clustering

- Representation for clustering
  - Document representation
    - Vector space? Normalization?
      - Centroids aren't length normalized
  - Need a notion of similarity/distance

- How many clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid "trivial" clusters - too large or small
      - If a cluster's too large, then for navigation purposes you've wasted an extra user click without whittling down the set of documents much.

Clustering

Notion of similarity/distance

- Ideal: semantic similarity.
- Practical: term-statistical similarity
  - We will use cosine similarity.
  - Docs as vectors.
  - For many algorithms, easier to think in terms of a *distance* (rather than similarity) between docs.
  - We will mostly speak of Euclidean distance
    - But real implementations use cosine similarity

Flat Clustering

Partitioning Algorithms

- Partitioning method: Construct a partition of *n* documents into a set of *K* clusters
- Given: a set of documents and the number *K*
- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Globally optimal
    - Intractable for many objective functions
    - Exhaustively enumerate all partitions
  - Effective heuristic methods: *K*-means and *K*-medoids algorithms

Flat Clustering

Partitioning Algorithms

- Start out with a partition based on randomly selected seeds (one seed per cluster) and then refine the initial partition
  - In a multi-pass manner (recursion/iterations)
- **Problems associated with non-hierarchical clustering**
  - When to stop ?
  - What is the right number of clusters (cluster cardinality) ?
- **Classic algorithms**
  - The *K-means algorithm*
  - The EM algorithm

Hierarchical Clustering

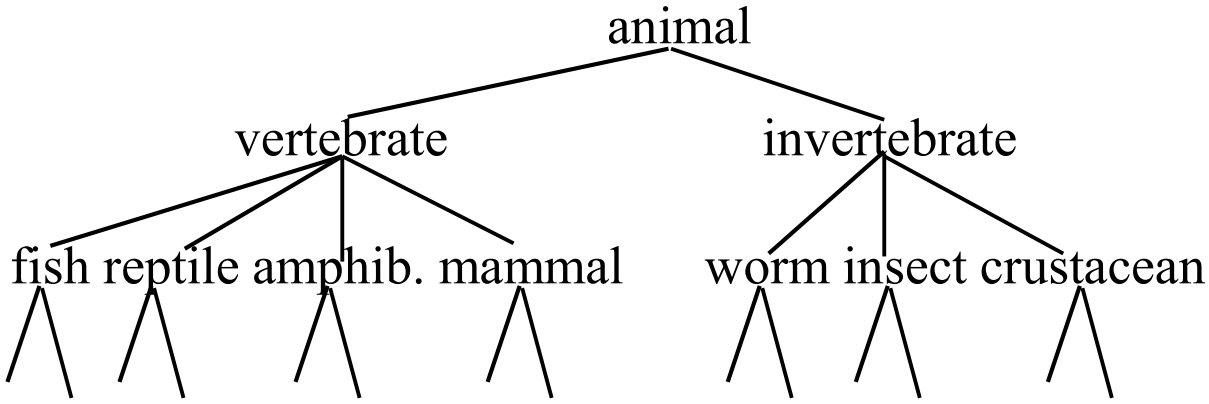
Bottom-up or top-down

- Bottom-up (agglomerative)
  - Start with individual objects and try to group the most similar ones
  - The procedure terminates when one cluster containing all objects has been formed
- Top-down (divisive)
  - Start with all objects in a group and divide them into groups so as to maximize within-group similarity

Hierarchical Clustering

Dendrogram

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents.

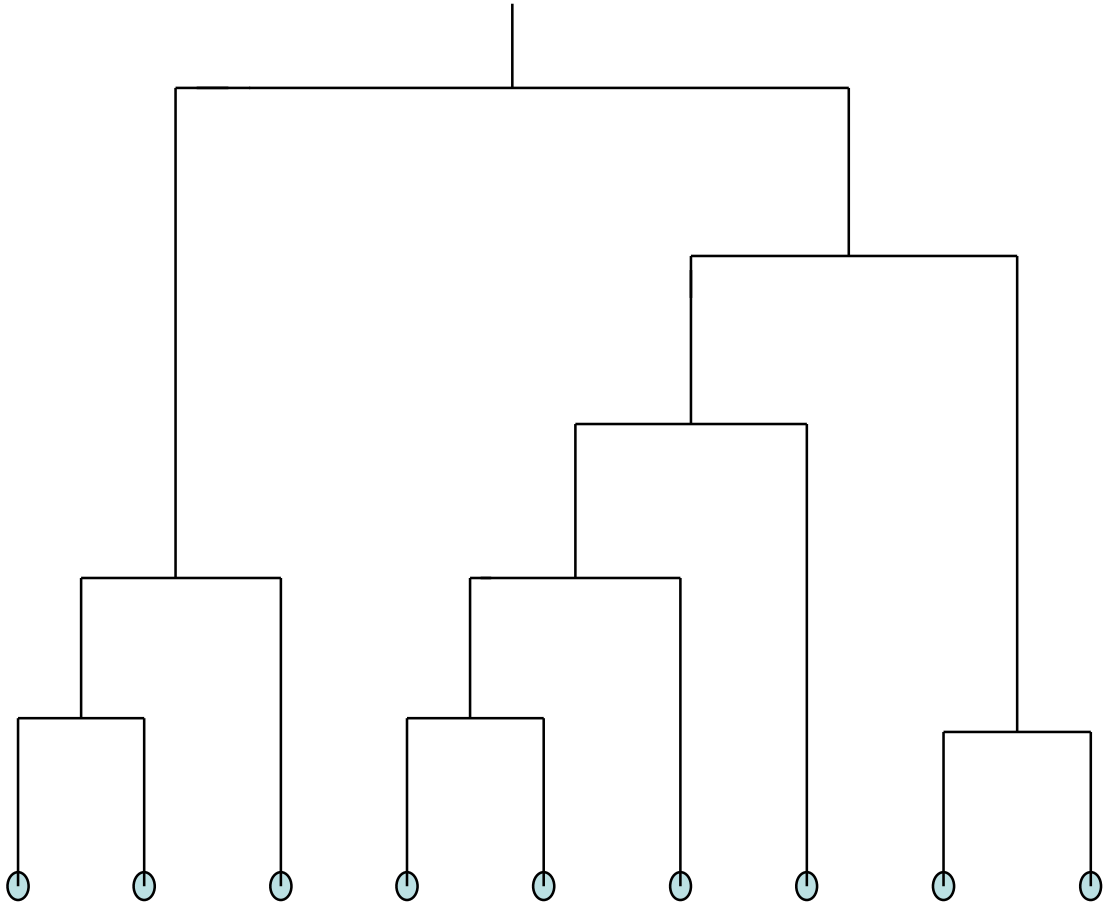


- One approach: recursive application of a partitional clustering algorithm.

Hierarchical Clustering

Dendrogram

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.



Hierarchical Agglomerative Clustering (HAC)

A bottom-up approach

- Assume a similarity measure for determining the similarity of two objects
- Start with all objects in a separate cluster (a singleton) and then repeatedly joins the two clusters that have the most similarity until there is only one cluster survived
- The history of merging/clustering forms a binary tree or hierarchy

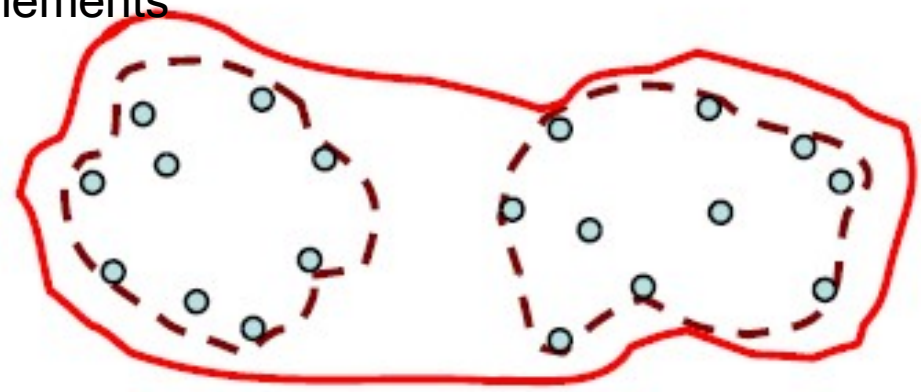
Divisive Clustering

- A top-down approach
- Start with all objects in a single cluster
- At each iteration, select the least **coherent cluster and split it**
- Continue the iterations until a predefined criterion (e.g., the cluster number) is achieved
- The history of clustering forms a binary tree or hierarchy

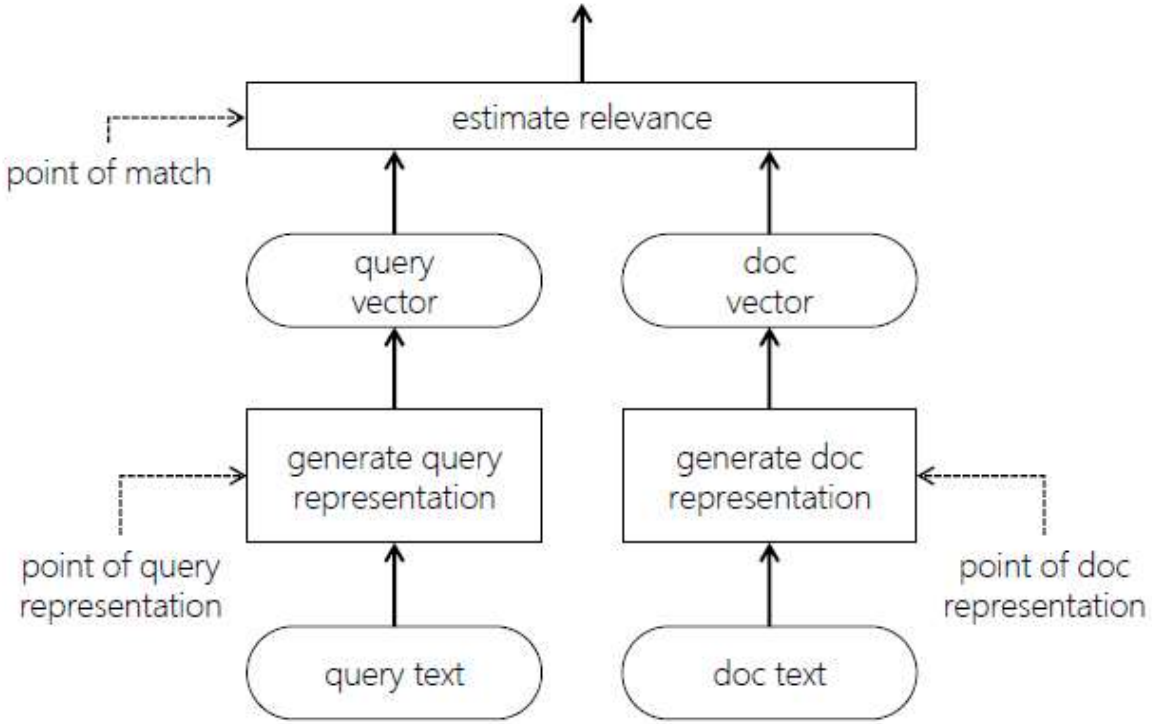
Measures of Cluster Similarity

Find the closest or farthest pair of clusters

- **Single-link**
  - Similarity of the *most* cosine-similar (single-link) samples
- **Complete-link**
  - Similarity of the "furthest" points, the *least* cosine-similar samples
- **Centroid**
  - Clusters whose centroids (centers of gravity) are the most cosine-similar samples
- **Average-link**
  - Average cosine between pairs of elements

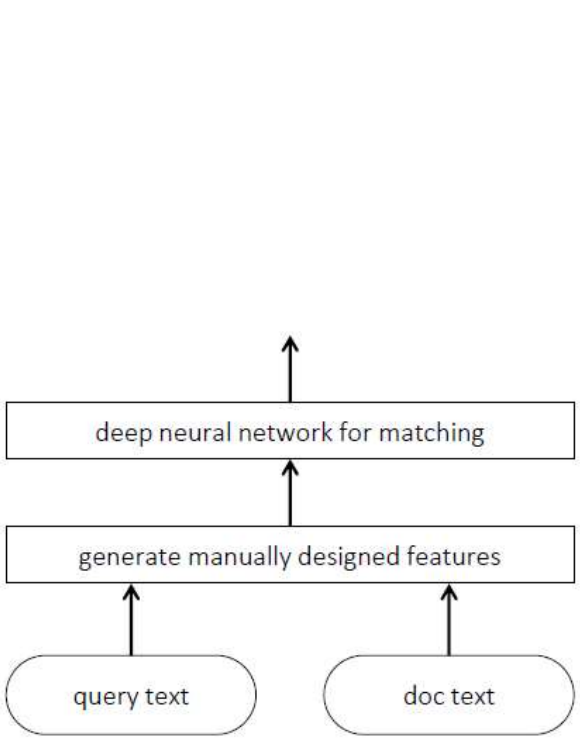


Deep learning methods for IR

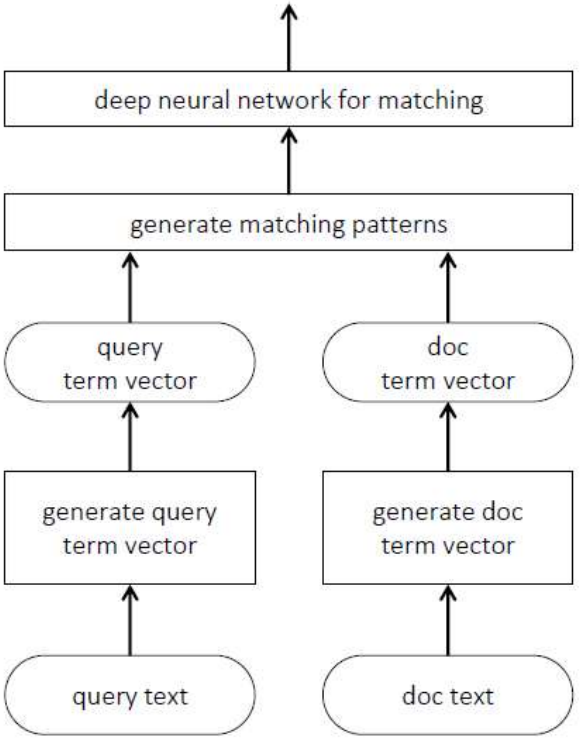


Stages in IR that a neural approach may impact, one or more [1]

Deep learning methods for IR



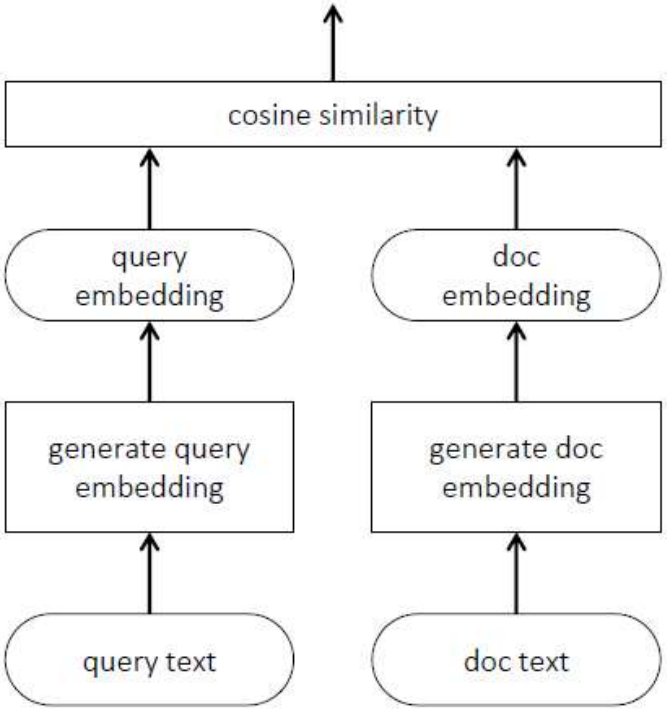
(a) Learning to rank using manually designed features (e.g., Liu [121])



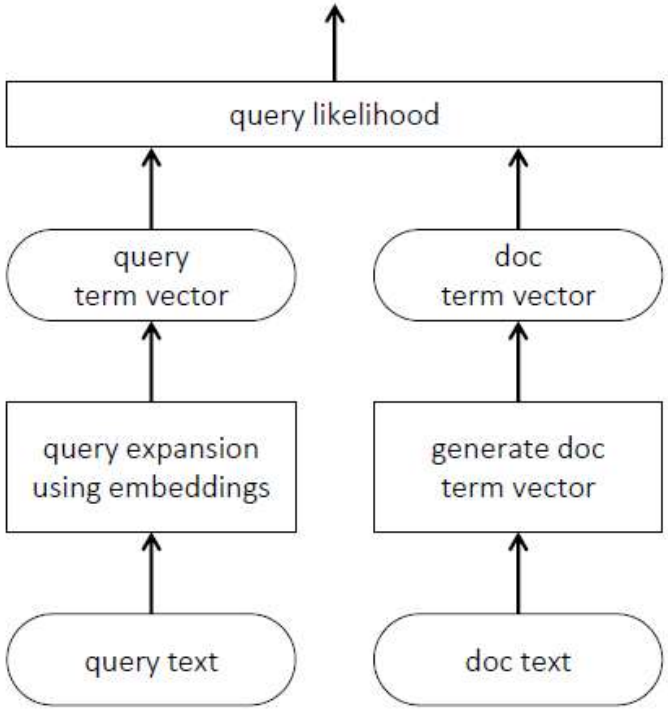
(b) Estimating relevance from patterns of exact matches (e.g., [71, 141])

Examples of different neural approaches to IR [1].

Deep learning methods for IR



(c) Learning query and document representations for matching (e.g., [88, 143])

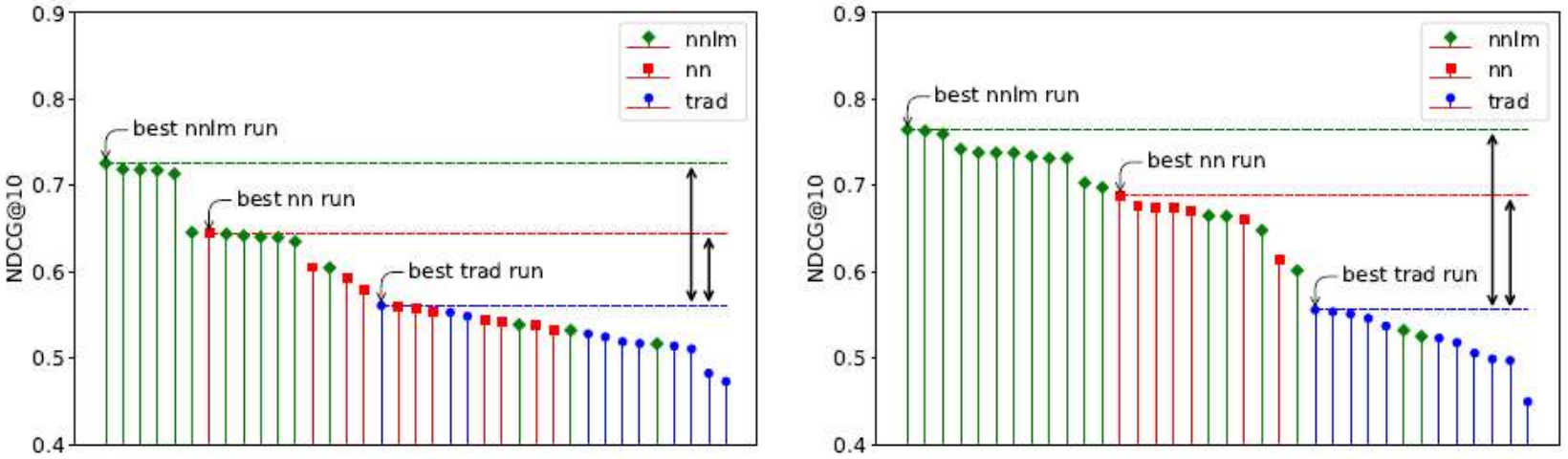


(d) Query expansion using neural embeddings (e.g., [51, 170])

Examples of different neural approaches to IR [1].

Data and challenges

- Trec deep learning track 2019 & 2020



(a) Document retrieval task

(b) Passage retrieval task

Figure 1: NDCG@10 results, broken down by run type. Runs of type "nnlm", meaning they use language models such as BERT, performed best on both tasks. Other neural network models "nn" and non-neural models "trad" had relatively lower performance this year. More iterations of evaluation and analysis would be needed to determine if this is a general result, but it is a strong start for the argument that deep learning methods may take over from traditional methods in IR applications.

Extended reading

[1] Mitra, B., & Craswell, N. (2017). Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*.

[2] Craswell, N., Mitra, B., Yilmaz, E., Campos, D., & Voorhees, E. M. (2020). Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

[3] Craswell, N., Campos, D., Mitra, B., Yilmaz, E., & Billerbeck, B. (2020, October). ORCAS: 20 Million Clicked Query-Document Pairs for Analyzing Search. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (pp. 2983-2989).

[4] Mitra, B., & Craswell, N. (2018). *An introduction to neural information retrieval*. Now Foundations and Trends.

[5] Kenter, T., Borisov, A., Van Gysel, C., Dehghani, M., de Rijke, M., & Mitra, B. (2017, August). Neural networks for information retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1403-1406).

NN4IR tutorials: http://nn4ir.com/sigir2017/

http://nn4ir.com/