Overview

1. Information retrieval (IR)
2. Basic representation schemes
3. Binary representation
   - Finding a similar document to a query
4. Representation tf-idf
   - Cosine similarity
5. Relation of IR systems to k-NN
6. Some problems and remedies
   - 6.1 Query completion / expansion
   - 6.2 Ranking / clustering of retrieved documents
7. Evaluation of IR systems
1. Information Retrieval

Information Retrieval ~ Document Retrieval
Retrieval of documents in response to a “query document”
(as a special case, the query document can consist of a few keywords)

- Document Collection
- Query Document
- Document Matcher
- Retrieved Documents

2. Basic Representation Schemes

Representation of documents using features representing:

- **Words**
  - Often the term *word-level tokens* is used instead.
  - Tokens can be annotated (e.g. with labels representing noun, verb etc.).
  - **Bag-of-words** representation exploits words, but the order is ignored.
  - **Word stem** represents a group of related words stripped of a suffix
- **Terms**
  - may represent single words or multiword units, such as “White House”
Representations schemes

Here we will consider:
- binary representation, indicating the presence (or absence) of words,
- representation using tf-idf

In each case the document is represented using a vector of numbers. Therefore this representation is often referred to as a vector-space model.

3. Binary Representation

Binary indicates presence / absence of words (terms).
Example with 7 documents, each represented using 4 terms:

<table>
<thead>
<tr>
<th>Document</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Document Retrieval: Finding a similar document to a query

Binary representation can be used for document retrieval:
- a query (query document) is represented in the same manner as other documents
- most similar documents are identified using some proximity measure (PM), such as \((1 - \text{Euclidian distance}) / \# \text{ terms}\)

<table>
<thead>
<tr>
<th>Document</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2/4</td>
</tr>
<tr>
<td>D2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1/4</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0/4</td>
</tr>
<tr>
<td>D4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/4</td>
</tr>
<tr>
<td>D6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1/4</td>
</tr>
<tr>
<td>D7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2/4</td>
</tr>
<tr>
<td>Query</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>3/4</td>
</tr>
</tbody>
</table>

One of the two most similar documents to the Query

4. Representation tf-idf

The method of tf-idf is based on the following assumptions:

- The term appearing often in the document may be more important for identification than the term appearing rarely.
  This aspect is captured by \(tf, \text{ term frequency in a document}\).

- If a term appears in many documents, it will be probably irrelevant.
  This aspect is captured by \(idf, \text{ inverse document frequency}\).
**Representation tf-idf**

Calculation of tf-idf of a term $t_j$ in document $d_i$:

$$w_{\text{tf-idf}}(t_j, d_i) = \text{tf} (t_j, d_i) \times \text{idf} (t_j)$$

where

- $\text{tf} (t_j, d_i)$ is *term frequency in a document*.
- $\text{idf} (t_j)$ is *inverse document frequency*.

Inverse document frequency is calculated:

$$\text{idf} (t_j) = \log_2 \left( \frac{N}{df(t_j)} \right)$$

where

- $N$ is number of documents
- $df(t_j)$ is a number of documents in which term $t_j$ occurred

---

**Cosine Measure of Similarity / Distance**

Cosine measure of similarity / distance is calculated by multiplying the weights of the shared words in two documents (this is similar to logical AND of the two vectors).

$$\text{cosine}(d_i, d_j) = \sum_{t_j} (w_{\text{tf-idf}}(d_i, t_j) \times w_{\text{tf-idf}}(d_j, t_j)) / (\text{norm}(d_i) \times \text{norm}(d_j))$$

$$\text{norm}(d_i) = \sqrt{\sum_{t_j} (w_{\text{tf-idf}}(d_i, t_j)^2)}$$
Example of a calculation of tf-idf

Calculation of tf-idf using an example with 7 documents, each represented using 4 terms.

Calculation of tf:

<table>
<thead>
<tr>
<th>Document</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D7</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Calculation of idf:

<table>
<thead>
<tr>
<th>Document</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>df(tj)</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>(N/df(tj))</td>
<td>2.00</td>
<td>4.00</td>
<td>2.67</td>
<td>2.00</td>
</tr>
<tr>
<td>(\log_2(N/df(tj)))</td>
<td>1.00</td>
<td>2.00</td>
<td>1.42</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Joining the two parts to obtain tf-idf

Resulting $w_{\text{tf-idf}}$ values:

<table>
<thead>
<tr>
<th>Document</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>Cos</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>0</td>
<td>1.42</td>
<td>0</td>
<td>0.67</td>
</tr>
<tr>
<td>D2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.50</td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.50</td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>0</td>
<td>1.42</td>
<td>0</td>
<td>0.71</td>
</tr>
<tr>
<td>D6</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.22</td>
</tr>
<tr>
<td>D7</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>Q1</td>
<td>1</td>
<td>0</td>
<td>1.42</td>
<td>1</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Normalization constants

The following table show the calculation of constants used in normalization of cosine similarity:

<table>
<thead>
<tr>
<th>Document</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>sum</th>
<th>sqrt</th>
<th>norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1.73</td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2.00</td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1.42</td>
<td></td>
</tr>
<tr>
<td>D6</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>2.24</td>
<td></td>
</tr>
<tr>
<td>D7</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>2.24</td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2.00</td>
<td></td>
</tr>
</tbody>
</table>
5. Relation of IR systems to k-NN

We note that a typical IR system, as described, is basically a k-NN method. After representing the documents as vectors, the system tries to identify the nearest match (or matches) to a query.

6. Some problems and remedies

6.1 Feature reduction
6.2 Query completion / expansion
6.3 Ranking / clustering retrieved documents
6.1 Feature Reduction

Removal of high frequency words - stop words,
Ex. removal of “about”, “cannot” etc.
Advantages: simplifies the search process; reduces the size of text

Suffix stripping and stemming (substitution of a word by a stem),
Ex. substitution of “moved” by “mov”
Unfortunately this can lead to errors.
In general, it is necessary to take into account the context.

Normalization of reduced words
Ex. Substitution of “mov” by “move” (infinitive)
This is usually done with the help of lexicon.

6.2 Query completions / expansion

Problem:
If the query document is very short, i.e. just a few words,
the system may retrieve a large number of documents.
Some of them will be irrelevant to the what the user intended.

Solution (general idea):
Complete / expand the query by adding more text.
Specific methods that can be used:
- Ask the user to expand the query;
- Add synonyms to the text by exploiting e.g. Wordnet
- Use others sources (e.g. Wikipedia) to expand the query;
- Exploit an idea similar to active learning:
  Ask the user to identify the documents of interest in the retrieved set
  and use these documents (or some words) to expand the query.
### 6.3 Ranking / Clustering Retrieved Documents

**Problem:**
The system will in general retrieve a large number of documents. Some of them will be irrelevant to what the user intended. Query expansion helps, but does not resolve the problem.

**Solution:**
- Provide a ranking of the retrieved documents:
  - Assess how well each satisfies the given query using an appropriate distance/probabilistic measure.
  - Order all documents using this measure and return top N items.
- Cluster the retrieved documents into groups and label them.
  - Rank the documents in each group.
  - The user may be interested in one of these groups only.

### 6. Evaluation of IR systems

Effectiveness is usually measured in terms of:

- **Precision**
  \[
  \text{Precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}}
  \]

- **Recall**
  \[
  \text{Recall} = \frac{\text{number of relevant items retrieved}}{\text{total number of relevant items in the collection}}
  \]

Recall / Precision Curves
Evaluation of IR systems

Precision $P_i = \frac{TP_i}{TP_i + FP_i}$

Recall $R_i = \frac{TP_i}{TP_i + FN_i}$

F measure, (strictly speaking F1), combines the two measures, giving similar weights to each (so called harmonic mean):

$$F1 = \frac{2 \cdot P_i \cdot R_i}{P_i + R_i}$$

In general we could use:

$$F_{\beta} = \frac{(\beta^2 + 1) \cdot P_i \cdot R_i}{(\beta^2 \cdot P_i) + R_i}$$

( in $F_{\beta}$ parameter $\beta = 1$)

Evaluation of IR systems

Precision and recall can be combined in different ways for different classes.

We distinguish:
- Micro-averaged F1
- Macro-averaged F1

$$\text{MacroF1} = \frac{2 \cdot (P_i + P_j) / 2 \cdot (R_i + R_j) / 2}{(P_i + P_j) / 2 + (R_i + R_j) / 2}$$
Evaluation of IR systems

TREC .. Text Retrieval Conference Competition
There were TREC 1-5 competitions (e.g. TREC 2 was in 1995)