MAI4CAREU

Master programmes in Artificial Intelligence 4 Careers in Europe

University of Ruse

Information Retrieval

Yordan Kalmukov

May 2023

Co-financed by the European Union
Connecting Europe Facility

This Master is run under the context of Action No 2020-EU-IA-0087, co-financed by the EU CEF Telecom under GA nr. INEA/CEF/ICT/A2020/2267423

Information Retrieval (IR)

1. Introduction

A decade or two ago, most of the data were structured or semi-structured and stored mainly in relational databases. However, this is not the case at the moment. Especially after the mass distribution of content management systems and social networks, which make users the main content generators. Millions of statuses and images are posted on social media every day. Multiple documents are added to various Internet sites, e-libraries, learning systems, popular online journals, etc. But to be useful, all of this content needs to be accessible and, most importantly, easily discoverable. Therefore, any document management system, or even more generally any type of resources, must provide the necessary functional search capabilities, both by additional external descriptors (keywords and text annotations) and by content.

Usually when people talk about documents, they mean textual documents, but this term is actually much broader and can include any type of objects, including images, audio, video, physical objects, even people. Images, audio and video files are relatively easy searchable by their multimedia content. For example, images can be searched by color content and its spatial distribution. In addition, all non-text objects (e.g. people, cars, properties, etc.) can be easily described with textual annotations or keywords selected from predefined structures, and then efficiently searched for by the provided descripting characteristics.

Content-based search is an important problem with practical significance, which applies in various subject domains, for example for:

- Search for similar documents, even in unstructured and unindexed collections. Any document management system should provide opportunities for fast and efficient document search, as well as to sort the search results according to the degree of similarity with the request.
- Building search engines on the Internet. Search engines are an indispensable means of accessing information on the vast web space, currently consisting of more than two billion websites.
- Grouping documents into clusters based on their common characteristics. For example, automated grouping of users by interests; grouping music or books by genres and thematic areas; etc.
- Pairing objects. Solving the assignment problem for optimal allocation of resources. Allows to optimally assigning resources to consumers; employees to tasks; financial resources to projects; to identify and assign appropriate evaluators and reviewers; to match suitable men and women on dating websites; and many others.
- Building recommender systems. Recommender systems are very popular on the Internet nowadays. For example, if a user is reading a certain publication, this type of system will automatically recommend him/her other similar articles, about the same or a similar topic. On this principle, the e-shops discover and recommend similar or related products, and video sharing platforms - similar or related videos. Of course, we cannot skip the controversial algorithms for content recommendation used by the social networks. They are often (correctly) criticized, for offering "more of the same" and filtering any alternatives and other opinions.

The search for similar documents is a fundamental task. Without a solution of it, the vast amount of data and documents published on the Internet would be undiscoverable and therefore useless. A search is usually performed by a query submitted by a user, and the result is returned as a set of documents related to it. However, it is important to determine not only which documents are related to the query, but also how closely they are related to it, with search results sorted in descending order by degree of similarity to the query.

2. Evaluation of search results

Search is performed by a user-submitted query, and the returned results are usually ranked, i.e. sort by degree of similarity with the query. In order to evaluate the accuracy of the IR system, and respectively the methods and algorithms it uses, it is necessary to assess how adequate the returned results are. However, who can judge whether a given result is adequate or not? Of course, people (users). The main goal of searching is to model the subjective human perception of similarity and semantic relatedness between documents, and to discover and retrieve only those documents that users believe are truly related to the search query.

Before reviewing the most commonly used evaluation metrics, some basic classification concepts should be clarified:

True positives (**TP**) – the number of returned documents that the search system correctly classified as adequate.

False positives (**FP**) – the number of returned documents that the system thinks are adequate, but in reality they are not. These are false positives, i.e. the system has misclassified them.

True negatives (**TN**) – the number of documents that the system correctly classifies as inadequate. I.e. both according to the system and in reality, the documents have nothing to do with the search query.

False negatives (**FN**) – the number of documents that the system thinks are inadequate but are actually adequate and related to the search query. Here, the system classified the documents in incorrectly as well.

In general, it is considered unrealistic for a search system to return all documents relevant to the request and all returned documents to be adequate. Because searching involves content analysis that can be ambiguous and misleading, it is virtually impossible for the returned results not to contain some amount of error and noise. Figure 1 visually presents the relationships between the main four classification concepts. Almost all accuracy evaluation indicators are actually based on them.

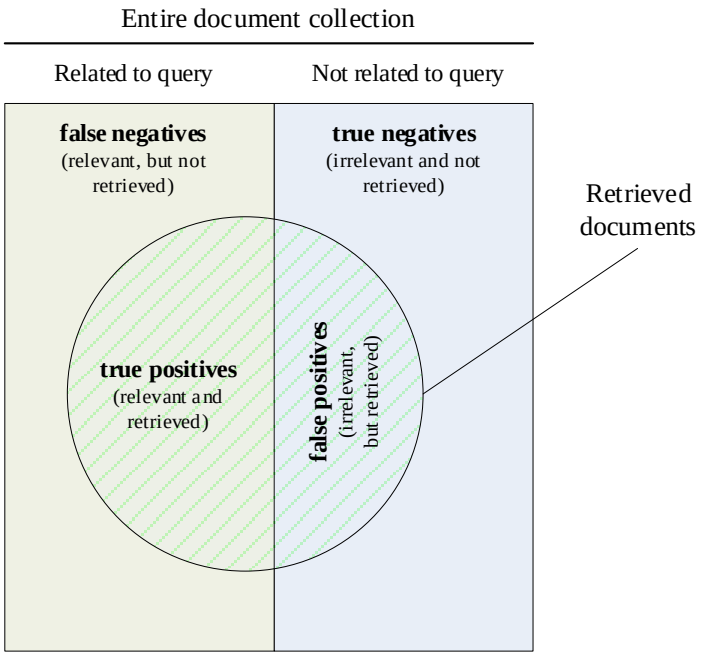

Figure 1. Relationship between true positives (relevant and correctly retrieved); false positives (irrelevant, but still retrieved); false negatives (relevant but missed/not retrieved); and true negatives (irrelevant and not retrieved).

Precision

Shows what proportion of retrieved results are adequate/relevant.

$$precision = \frac{number\ of\ relevant\ retrieved}{number\ of\ all\ retrieved} = \frac{TP}{TP + FP} \quad (1)$$

For example, if the search returns 20 results and 15 out of them are related to the query while the remaining 5 are not, then the precision is 15/20, i.e. 0.75.

Precision is very easy to assess - it is simply assessing how much of the returned results are adequate/relevant.

Recall

It shows how much of the relevant (i.e. related to the query) documents are retrieved. This measure is also called sensitivity.

$$recall = \frac{number\ of\ relevant\ retrieved}{number\ of\ all\ relevant} = \frac{TP}{TP + FN} \quad (2)$$

Returning to the above example, if there are 50 relevant documents in the database, then the recall is 15/50, i.e. 0.3.

Unlike precision, recall is much more difficult to evaluate. If the database (the collection of documents) is static and it is known in advance how many documents it contains related to the specific query - then it is easy. But usually the database is dynamic and frequently changing. In practice, it is almost impossible to know at any given moment how many documents it contains related to a particular user query. Of course, there are exceptions - for pre-prepared datasets that contains queries and rated results.

Precision and recall are usually related, and precision is often measured at predetermined levels of recall.

Accuracy

Traditionally, accuracy is defined as the ratio of all correctly classified documents to the number of all documents in the database.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Accuracy is often criticized and accused of being a misleading measure. Why? Let, for example, there are 100 documents in the database. 5 of them are related to the given search query, ie. are adequate, and 95 are not. Then, if the system returns no results, the accuracy will be 0.95 (or 95%). Formally speaking, this is true, but it is strange and unexpected for the user to get 95% accuracy after the system has not found and not returned any related documents, although there are such. Also, since accuracy counts TN in the numerator, its calculated values will always be very, and unduly, high. The reason is that with a huge collection of heterogeneous documents, the number of documents unrelated to a given query will always be many times higher than the number of related documents.

Another problem with accuracy that applies to all other measures (without precision) is that it is very difficult to determine the values of TN and FN in a real-world situation, unless dealing with pre-prepared datasets having test queries and pre-rated results.

Top-N accuracy

In machine learning, especially in object classification, Top-N accuracy is often used. The classifier returns not a single guess, what the object is, but several ones, sorted in descending order of probability. According to Top-1 accuracy (the usual definition of accuracy), the classifier has correctly determined the object's type if the guess with the highest probability matches the actual object. Top-N accuracy allows for a more tolerant and less stringent interpretation of accuracy. For example, with Top-5, an object is considered correctly classified if it is present among the first 5 guesses of the classifier, even if the guess with the highest probability is a completely different object than the actual one. In this case, at Top-1 accuracy, the classification is incorrect because the most likely guess is not the actual object at all. But according to Top-5 accuracy, the classification is correct because the actual object is present among the top 5 guesses, even though it is not the most likely. In a number of situations where it is not necessary to specify the type of an object with great precision, such a loose interpretation of precision is justified, even encouraged. For example, in recommender systems that look for other similar objects – other songs in the same or related genres on video sharing platforms; other similar, but not quite the same, products in online stores, in online bookstores, etc. Through such more tolerant interpretation of accuracy, the user may discover other interesting offers besides what he was looking for.

3. Evaluation of results ordering

The evaluation measures discussed above only assess how adequate the retrieved results are, but not whether they are ordered correctly or not. The order/ranking depends on the similarity measure that is used to calculate the degree of similarity between the query and each of the documents.

To assess whether the order of the results is correct, one should determine the degree of correlation between the automatic order and the reference order provided by real people. In this case, the absolute value of the similarity factors between the query and each of the returned results does not matter. What matters is the relationships between the similarities since they determine the order. Getting users to specify an accurate absolute value (eg. 81% or 64.5%, etc.) for the similarities between the query and the results is extremely difficult. One tends to give higher values, another lower. But, if they are told "Sort the results by degree of similarity", then the probability that different users will sort them in the same way is significantly higher.

To determine the degree of correlation between the automatic order and the user-supplied reference order, the Pearson's linear correlation coefficient is usually used, "comparing" positionally the relationships between all pairs of similarities. The relationship between two results i and j can be "greater than", "less than" or "equal to". When i > j, it means that i is situated in the result list before j. If the similarity factors between the query and the two results i and j are the same, then the relationship between i and j is "equal" and the results should be displayed with the same weight (position), even if they have to be visually arranged below each other. In order to calculate the Pearson correlation coefficient (4), these relationships must be coded with numbers. For example: 1 for less than; 2 for equal; and 3 for greater than.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \quad (4)$$

where:

r_{xy} – Pearson's correlation coefficient. Indicates the degree of linear correlation between the set of relations X and the set of relations Y.

x_i – the code of the i-th relation from the X set, the i-th value of X.

y_i – the code of the i-th relation from the set Y.

$\overline{x}$ – the arithmetic mean value of all elements of X - the so-called sample mean of X.

$\overline{y}$ – the sample mean of Y.

n – the number of elements (relations) in the sets X or Y.

The relationship between all pairs of results means to consider the relationships between each result and every other result, not just between adjacent results. For example with 4 results: 1-st with 2-nd, 1-st with 3-rd, 1-st with 4-th, 2-nd with 3-rd, 2-nd with 4-th, and 3-rd with 4-th. In general, relationships only between neighboring results (for example: 1-st with 2-nd, 2-nd with 3-rd and 3-rd with 4-th) could be used as well, but taking into account "each with each" gives a significantly more accurate evaluation of order, especially in cases where the displacement is by more than one position. Considering only the relationships of neighboring results leads to two problems that can be easily illustrated by examples. Let the following relationship between the similarities of the neighboring results 2, 3 and 4 with the query be given:

sim(result_2, query) > sim(result _3, query) < sim(result _4, query)

It is clear that the degree of similarity of results 2 and 4 with the query is higher than that of result 3. That is, results 2 and 4 should be ordered before result 3. But, which one of the two results, 2 or 4, should be first and which should be second in the order? If we consider only relationships between neighboring results, then this question cannot be answered at all. It cannot, because it is not known what the relationship between results 2 and 4 is. However, if “each-to-each” relationships are considered, then one will also know what the correct relationship between results 2 and 4 is. This will allow much more accurate comparison of the automatic results order with the reference one provided by users.

Another problem with considering relationships only between neighboring results is that it does not allow distinction between swapping two neighboring results (which is considered a small error) from swapping two significantly more distant results (which is a significantly larger error). When accounting relationships between all pairs of results, such a distinction is possible. Let, for example, the system returns 200 similar documents. This means 39,800 relationships between individual results. If two neighboring results were swapped, say 5-th and 6-th, the error would be 2/39800. However, if the 5-th and 80-th are swapped, which is considered a significantly more serious error, then the relationships with all 75 intermediate results will be different in the automatic order than in the reference ranking. So the error will be (2*75)/39800 ie. 75 times higher, which is actually correct. It is correct because swapping two neighboring results is not much of a problem, but moving adequate ones to the end of the list and inadequate ones to the beginning of the list is a problem.

4. Document representation models

In order to calculate the degree of similarity between the query and the documents, it is necessary to find some type of semantic representation of both. The chosen representation model should be able to recreate the semantics, or the meaning, of the document and at the same time allow its automated processing using mathematical means. The goal is, ultimately, by using one or more mathematical formulas, to calculate a single number that indicates how closely one document is related to another document. This indicator is called a similarity coefficient or semantic similarity between the documents. Its calculation is a direct consequence of the chosen document presentation model.

Document presentation models can be divided into 3 large groups:

- ***Set-theoretic models***. Documents are represented by a set of a finite number of features - most often words. This set can be unordered, but feature vectors are usually used. Semantic similarity between documents is calculated using metrics known from the theory of sets.
- ***Algebraic models***. Documents are represented by vectors, matrices or tuples containing real numbers. These numbers generally represent the *term weights* in the document. The weights consider both local and global features — i.e. how important a given word is to the description of the document, but also what its meaning is within the whole collection of documents. Semantic similarity between documents is usually calculated as the cosine of the angle between the vectors, but can also be calculated in other ways.
- ***Probabilistic models***. The searching and retrieval process is viewed as probabilistic inference – estimating the probability that a given document is related with the user's query. Here again, the documents are represented by vectors or matrices, but the numbers in them represent the probability that the corresponding words occur in the documents. In this sense, probabilistic and algebraic models are very similar. The only difference between them is the way the elements of the vectors are calculated. But ultimately, probabilities are also determined based on statistical characteristics such as frequency of occurrence, etc.

Figure 2 presents a classification of the models, together with some of their specific representatives. The majority of them will be examined in details, impalmented and experimentally examined.

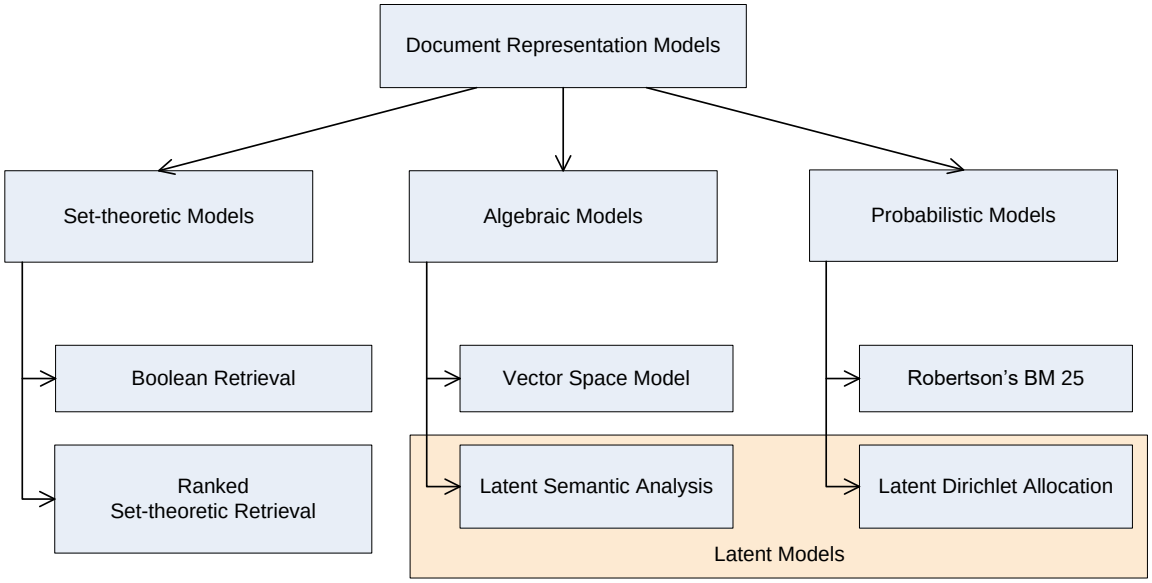

Figure 2. Classification of the document representation models

It is noticeable in the figure that the Latent Semantic Analysis and the Latent Dirichlet Allocation are grouped together, within the group named "Latent Models". In the other models, all words (or features) describing the semantics of the document are treated as independent. For example, the words "hat" and "capella" will be processed as completely separate and unrelated, even though in reality they have a synonymous meaning. This is not the case with latent models. They make it possible to discover relationships, not only between documents, but also between individual words. Thus, words that have a similar meaning are grouped into more general latent topics. Topics are called latent or hidden because they are not explicitly stated in the documents, but are inferred from words having a similar meaning. For example, the words "rocket", "shuttle", "space", "probe", "planet", etc. form a space theme. A document is considered to be space-related if it contains any of these words (one or more). Thus, two documents can be classified as semantically related, even if they do not contain the same words, but only interconnected ones. In this way, latent models successfully cope with synonyms and partly with polysemy – the different meanings of the same word.

5. Methods of describing documents

Methods of describing documents are directly related to document representation models. In general, methods of describing documents could be divided into:

- ***Explicit***. Users are required *to provide explicitly* additional information to describe the documents. These methods are mandatory, in case where it is not possible to preform an automatic content analysis and extract appropriate descriptors/features. But they can be used in many other cases as well. The description can be done by pre-defined *set of keywords* or *text annotation*.
- ***Implicit***. Users do not need to do anything. The description of the resources/documents is obtained by performing a content analysis. These methods are quite suitable for text,

images, maybe audio and video content, but not for describing people or physical objects, for example.

Some resources/documents, for example textual documents, images, audio and video files, can be described both explicitly and implicitly. For others, however, the explicit description is the only possible. For example, if it is necessary to describe the interests of people for whom there is no information and there is nowhere to get it automatically. Then each person must explicitly provide this information about himself/herself. He/she can do so by selecting his/her interests from a predefined list of keywords; or from a taxonomy of keywords; or enter the relevant keywords himself/herself; or to describe it in free text. All these options are possible, but the chosen method for describing the objects/documents will largely determine the model for their presentation, as well as the semantic similarity measure between them.