Moreno-Schneider, Julián, Martínez, Paloma, Martínez-Fernández, José L. (2017).Combining heterogeneous sources in an interactive multimedia content retrieval model.Expert Systems with Applications, 69, pp. 201-213.

DOI: 10.1016/j.eswa.2016.10.049

Combining heterogeneous sources in an interactive multimedia content retrieval model

Julián Moreno Schneider[a,*], Paloma Martínez[b], José L. Martínez Fernández[c]

[a] *Deutches Forschungzentrum für Künstliches Intelligenz - DFKI, Alt-Moabit, 91c, 10559, Berlin, Germany*
[b] *Computer Science Department, Universidad Carlos III de Madrid, Avda. Universidad, 30, 28911, Leganés, Madrid, Spain*
[c] *MeaningCloud Llc.*

Abstract

Interactive multimodal information retrieval systems (IMIR) increase the capabilities of traditional search systems, by adding the ability to retrieve information of different types (modes) and from different sources. This article describes a formal model for interactive multimodal information retrieval. This model includes formal and widespread definitions of each component of an IMIR system. A use case that focuses on information retrieval regarding sports validates the model, by developing a prototype that implements a subset of the features of the model. Adaptive techniques applied to the retrieval functionality of IMIR systems have been defined by analysing past interactions using decision trees, neural networks, and clustering techniques. This model includes a strategy for selecting sources and combining the results obtained from every source. After modifying the strategy of the prototype for selecting sources, the system is re-evaluated using classification techniques. This evaluation compares the normalised discounted cumulative gain (NDCG) measure obtained using two different approaches: the multimodal system using a baseline strategy based on predefined rules as a source selection strategy, and the same multimodal system with the functionality adapted by past user interactions. In the adapted system, a final value of 81,54% was obtained for the NDCG.

Keywords: Multimodal Information Retrieval, User adaptation, Retrieval Engines, Rule-based Expert Systems

1. Introduction

Present day society is characterised by a constant technological revolution, where the generation and consumption of information is attaining huge levels. The amount content on the internet, the main container of information, is increasing exponentially. There are plenty of services that offer multimedia content. Among well-known examples, Google (www.google.com) specialises in text content, YouTube (www.youtube.com) provides searches for videos, SoundCloud (soundcloud.com) facilitates music sharing, and Flicker (www.flickr.com) allows users to publish and search for photos. When dealing with multimedia, such systems are mainly based on textual metadata.

When dealing with audio, images, or videos, commercial systems are mainly based on the characterisation of resources in terms of textual metadata, which is later matched against user query expressions. Some examples of metadata for documents are *'author'*, *'date of creation'*, *'title'*, and *'language'*.

Thus, retrieval methods must evolve to become dependent on the device used to query (PC, smartphone, tablet, etc.), what is being queried, and who is querying. Furthermore, advances in the devices available to users are leading to a change in the formats applied in the definition of queries. Google has introduced voice query, where users can interact with the search engine by using a microphone to formulate a query, and previously they have included queries using images to search for other similar images.

The nature of internet access is also changing. The use of smartphones has exceeded the use of traditional computers for browsing the internet, but other devices are also becoming popular, such as smartwatches (9%), smart televisions (34%), games consoles (37%), smart wristbands (7%), and tablets (47%).[1]

The main problem presented by this growing presence of multimedia content is that users need to access larger and larger quantities of information in different formats and sources, and they wish to do so in a faster and easier manner, without having to query several sources.

If we consider a scenario where a journalist (say a sports editor for television) has to prepare news regarding F1, the journalist must cover information from all F1 races, travelling to all F1 Grand Prix locations for live

*Corresponding author
Email addresses: julian.moreno_schneider@dfki.de (Julián Moreno Schneider), pmf@inf.uc3m.es (Paloma Martínez), jmartinez@meaningcloud.com (José L. Martínez Fernández)

[1] Data extracted from http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/ accessed at 16/07/2016

broadcasts. They must document and archive all audiovisual material captured in a race. At the same time, they must develop additional pieces of information related to the last race. Retrieving these pieces of information is a difficult task, which can be simplified by using a multimodal retrieval system.

This retrieval must be simple, quick, and transparent to the user. Web search engines are the most well-known retrieval systems, but these do not allow the mixing of formats in queries. That is, a query composed of a combination of text and image cannot be performed.

When dealing with a technology that can query several retrieval engines, two problems arise for each engine that is considered. Namely, when this engine should be queried, and how its results are processed. Most techniques rely on the mode of the query to select an engine, and a simple mixture to present the final list as a combination. Therefore, they do not really adapt to different environments or queries. Furthermore, new techniques may have to be developed if we would like to work with several multimodal engines.

The main goal of this study is to adapt the functionality of an *interactive multimodal information retrieval (IMIR)* system based on past user behaviour. In particular, the aim is to exploit past interactions of an IMIR system through the use of classification algorithms, in order to avoid the need for expert-defined rules. We attempt to employ semi-supervised machine learning-type decision trees and neural networks. To accomplish this goal, we must fulfil two preliminary tasks. First, we define a multimodal information retrieval model that queries multiple heterogeneous sources, emphasising which sources are queried and how the results are combined. Second, we implement a working IMIR prototype based on this model. This implementation will later be adapted to take into account past user interactions.

The remainder of this article is organised as follows. Section 2 reviews work related to multimodal information retrieval and expert systems. Section 3 describes the defined formal model. The implementation of a basic prototype based on this model is described in Section 4. The functionality adaptation techniques of the IMIR prototype based on past user interactions are presented in Section 5, and evaluated in Section 6. Finally, conclusions and directions for future research are presented in Section 7.

2. Related Work

In this section, the main components of an IMIR system are described, considering the perspectives of the repositories that are queried, the manner in which the user may formulate queries, the underlying information retrieval (IR) models, the combination of retrieval engines required to answer user queries, and finally how the results are merged to obtain a list to be displayed to the user.

2.1. Multimedia Information

Information collections are divided according to the modes of the objects that compose them. A monomodal collection contains items from a single mode, such as the Wikipedia dataset used in (Hong and Si, 2012). By contrast, a multimodal collection contains objects from different modes, such as in the work of Yilmaz et al. (2012), which manages video and text; or the work of Camargo and González (2016), which employs two data sets of images, Flickr4Concepts and MIRFlickr (Huiskes and Lew, 2008).

Furthermore, there is a special case in this division: monomodal collections containing multimedia objects accompanied by metadata, such as the ImageCLEF 2011 Medical Retrieval Task dataset (Kalpathy-Cramer et al., 2011), which encompasses images and metadata. Finally, it is interesting to mention two completely multimodal collections. The work of Jou et al. (2013) employs a multimodal collection composed of 18000 hours of broadcast news, 3.58 million of new articles, and 430 million Twitter messages; and the TREC Federated Web Search (Fed-Web) Track 2013 Forum (Demeester et al., 2013) offers a multimodal collection composed of results obtained from 157 real web search engines, divided into 24 categories (ranging from news, academic articles, and images to jokes and lyrics). The collection contains both the search result snippets ($1,973,591$) and the pages ($1,894,463$) that the search results link to (that is, the HTML of the corresponding web pages).

2.2. Representation of Information Needs: Query

In most cases, queries arise in the textual mode, such as on commercial internet search engines (Yahoo, Bing, Google, etc.). For further details, see the work of Sushmita (2012). Some studies have used multimedia elements as queries, such as image queries (Wong et al., 2005), voice queries (Hauptmann et al., 2002), and short videos (Yang et al., 2012). Some researchers have studied multimodal query representation using specific languages, such as rich unified content description (Daras et al., 2011). These types of languages are interesting, because they offer the capability of representing every multimedia element in a query. In our work, we include a formal representation in the model definition.

2.3. Retrieval Techniques

Considering that retrieval techniques are not the focus of this study, only a brief introduction is provided, in order to provide some context to the reader. For a complete review of IR techniques, see (Baeza-Yates and Ribeiro-Neto, 2011) and (Manning et al., 2008). Retrieval through the matching of a text query and document content (keywords) is the most commonly employed method, such as in (Hong and Si, 2012) or (Görg et al., 2010). One study that investigates image retrieval based on low-level features is (Romberg et al., 2012). There exist other studies that have

used metadata to perform the retrieval of multimedia elements, such as (Hauptmann et al., 2002) for retrieving videos or (Lana-Serrano et al., 2011) and (Benavent et al., 2013) for retrieving images.

Available internet search engines (such as Yahoo or Bing) have also been used as *retrieval engines* in some studies, such as (Sushmita, 2012). Another interesting work is (Torres, 2005), which defines the visual object information retrieval (VOIR) prototype, combining two layers (conceptual and feature-based) to perform retrieval.

Multimedia retrieval based on annotations, relevance feedback, and concepts is similar to a metadata-based search. Documents are retrieved based on the similarity of the document and query annotations. Some methods employing this approach are the Mediamill system (Worring et al., 2007) and the ESCRIRE project or EsCosServer architecture (Medina-Ramírez, 2007).

Other types of multimodal retrieval systems create combined or centralised indexes, containing all modes of documents, such as (Marchand-Maillet et al., 2011).

2.4. Selecting Different Retrieval Engines

The retrieval engine (RE) selection or handling approach is responsible for selecting which REs are triggered by each query, and in which order they are queried, in cases where there is more than one RE suitable for the input query. The basic handling approach is to send the query to all available systems without distinction (Hong and Si, 2012). Another approach is to divide the query into elements according to their modes, and send each element to its corresponding RE (Demner-Fushman et al., 2012). We implement a similar approach to this, where we split the query into its elements, and then for each element we query every RE that accepts that type of element.

Another common approach to multimedia retrieval is to employ multiple REs sequentially. Hu et al. (2011) describe a music retrieval system, which searches for similar audio elements using text and then queries a content-based music retrieval system. The approach proposed in (Vallet et al., 2012) first searches using text in the external sources DBPedia, Flicker, and Google Images, and then uses these images to retrieve video by visual content. In Hauptmann et al. (2002), a video retrieval system is presented that allows voice queries. It transcribes the query, and matches it against the information extracted from videos. Some more complex techniques can be found in (Chernov et al., 2006), which implements a *broker* (handler) to select the systems to be activated, depending on the terms present in the query. A probabilistic approach to selecting REs, depending on the relevant entities that each engine would return, is presented in (Balog et al., 2012). Although our initial approach is based on the elements of the query (text, image, etc.), after applying the adaptation techniques to the system (as described in Section 5), the new strategy is based on the linguistic information of the query.

Federated search systems cover information retrieval from several heterogeneous retrieval engines. As claimed in Demeester et al. (2013), 'Federated search allows the inclusion of hidden web collection results that are not easily accessible by other ways.' The first task of the federated web search track (Demeester et al., 2013) is to evaluate and compare different resource selection strategies for a federated search. In (Pal and Mitra, 2013), search engines are ranked based on a score computed using the frequency of occurrences of query terms in the top eight results offered by each search engine. Finally, in (Bellogin et al., 2013) three different approaches were tested: (1) considering similarities between categories of the query and the results, (2) concatenating of all of the snippets from each resource and indexing them as a single document, and (3) aggregating the two previous scores using a Borda voting mechanism (Dwork et al., 2001).

2.5. How Results are Merged

Whenever multiple REs are queried, each provides a set of results, and these must be processed to obtain a single set, which is then returned to the user. First, it is interesting to introduce systems that perform retrieval through joint indexes, where the fusing of results is performed before the retrieval process by representing the documents in the index feature vector space, such as in (Demner-Fushman et al., 2012) and (Marchand-Maillet et al., 2011).

Considering post-retrieval approaches, a simple method is to organise the results randomly, as in (Chernov et al., 2006). Another common approach is the reordering of results based on their scores, but this implies that scoring criteria must be unified, in order to be homogeneous among different REs. In (Arampatzis et al., 2011), two scores are fused, one from textual retrieval and one from visual retrieval. In (Romberg et al., 2012), two REs are employed (one for images and one for texts), and these are combined using a linear combination.

The most common approach, as followed in the federated web search track, is the rearrangement of results based on scores. The authors of Guan et al. (2013) compute the score of each document as a linear combination of similarities between the query and different fields in a combined index, while in (Pal and Mitra, 2013) the original ranking obtained from the search engine and the search engine score value obtained according to the source selection strategy are combined linearly. The method in (Mourao and Magalhaes, 2013) considers that each list of results from an engine has a score that is equal to the ranking. Subsequently, it looks for results that appear in more than one list, to add the scores of every list.

The approach implemented in our prototype (see Section 4) is based on a combination of weighted scores from each RE. Our approach is similar to that of Pal and Mitra (2013), with one key difference: the ranking of the integrated sources is determined by the order considered in the rules adopted in the handling strategy.

In (Bota et al., 2014), a framework is presented for developing *composite retrieval*. This type of retrieval is based on the querying of heterogeneous web search systems and

the generation of a combined response, which is composed of a set of bundles, where each one encompasses results from a different vertical.

Other than these, there exist more complex approaches, which determine the new order of results using machine learning techniques. For example, in (Hong and Si, 2012) the authors use a central index, containing a summary of every document of the collections. Owing to the fact that there have not been many studies that consider machine learning techniques for the combination of results, we will explore this possibility in this paper to incorporate functionality adaptation.

The most interesting work concerning the fusing of results is the approach presented in (Wu and Crestani, 2015). This paper describes a geometric space where the documents are associated to the query and the RE, so that the space is a hypercube of dimension n (the number of documents). Each RE returns a vector with a value associated to each result, which is zero when the result is not returned by the RE. This approach is similar to the fusion strategy that we formalise in our model.

2.6. User Behaviour

Adaptive information retrieval has attracted the interest of researchers, because current systems work in the same manner for all people at all times. There exist some studies that create or adapt user models in order to classify the behaviour of users (Rekha et al., 2011), while others employ search query histories (Golovchinsky and Diriye, 2011). Some different interactions are interesting to record, depending on the specific purpose. Relevance judgments, known as *relevance feedback* (Salton and Buckley, 1997), are the most commonly employed interactions as an indicator of user behaviour.

Another interesting interaction is the *analysis of clicks*. In this sense, the work of Agichtein et al. (2006) adapted the simple approach of ignoring the original scores of the ranker, and instead simply merged the rank orders.

In our case, we consider both types of interactions. The prototype will register the query history and user feedback. All of this information is later employed for the functionality adaptation.

2.7. Meta Search

To conclude this section, we will describe meta-search systems. A *meta-search engine* queries different search engines (SE), normally web search engines, in order to offer the user a better set of results for a certain query without the user needing to query different systems himself.

A superb description of the field of meta-search can be found in (Manoj and Jacob, 2008). The authors provide a detailed description of the field up until 2008. It is demonstrated that there was significant interest in this kind of web search in the early 2000s, and there were plenty of different approaches considered for search engine selection and the combination of results. The same authors also produced the article (Manoj and Jacob, 2013), where they present a programmable meta-search engine based on T!, a LISP-like programming language. Another recent study is (Manral and Hossain, 2015), which describes a meta-search engine that considers the meta-data of websites (title, keywords, etc.) together with the rankings of the websites in the original search engines, by applying the PageRank algorithm (Brin and Page, 1998).

Furthermore, meta-search systems have also been applied to specific domains, as in the work of Smalheiser et al. (2014), which presents a domain-specific meta-search engine, called Metta, that is useful for querying biomedical literature. It is important to mention that this only queries five different biomedical databases, and only within a specific domain. This resembles the retrieval of information and documents that normally follow the same structure, which makes the task considerably easier than a general purpose meta-search engine such as ours.

There also exist some commercial meta-search services in the domain of travel, such as *Expedia* (www.expedia.es), *Kayak* (www.kayak.es), and *Skyscanner* (www.skyscanner.es), that offer results from different travel providers. If a user performs a search, they obtain results from different engines for flights, hotels, etc. The results are reasonably structured, and they contain the same information.

The meta-search approach is similar to the approach that we develop in this work, where several different engines are queried and the results are combined into a single set. The main difference is that our approach considers heterogeneous search engines, which respond with a wide variety of formats (such as documents, semantic elements, and concrete answers), while most of the search engines queried by meta-search systems return results with common characteristics in the format or domain.

2.8. Discussion

After reviewing the main related work, we now introduced our research in this paper, which is based on various approaches. The management of a multimodal query is based on the approaches of Yang et al. (2002) and Marchand-Maillet et al. (2011). The engine selection (handler) strategy is based on a mixture of the approaches in Renaud and Azzopardi (2012) and Demner-Fushman et al. (2012), which employ the content of the query (terms) to analyse which *RE* to query. Finally, we implement a sequential execution approach, similar to that used in (Hauptmann et al., 2002).

To the best of our knowledge, there exists no previous work that considers multimodal information retrieval (texts, images, videos, and audio) through semantic relations, while also taking into account the user and their behaviour within the system. The existence of this gap justifies the approach taken in this article.

3. Model to Describe MIR Systems

The objective is to generate different IMIR systems using a model that allows for standardisation among all systems that are based on the same model. This standardisation allows the comparison of different systems, and the ability to exchange modules between them.

The six main components of an IMIR system (multimedia information, retrieval engines, query modalities, handler, fusion of results, and interactivity) are explained in the following sections.

Figure 1 illustrates a general architecture, encompassing the elements that are considered in the formal model.

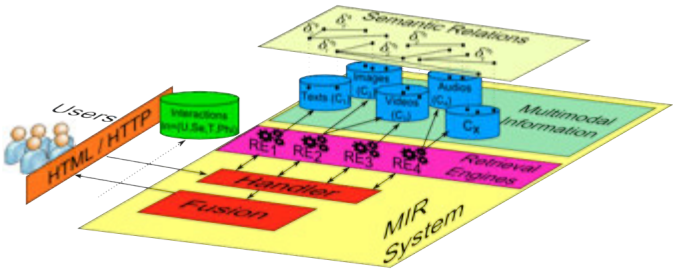

Figure 1: General architecture of the formal model to define MIR systems

In observing Figure 1, it is important to remark that in this approach several *REs* (RE_1, RE_2, RE_3, etc.) can coexist, and that multimedia repositories could be related by means of semantic relations included in an ontology (upper green plane in Figure 1), if an ontology-based RE is included. The handler is the component in charge of deciding which *REs* are queried, and in which order. The fusion module is responsible for combining the results returned by the different *REs* in order to display them to the user using appropriate visualisation techniques.

3.1. Multimodal Information

Multimodal information is sorted into a set of collections (see Equation 1), where N is the number of collections:

$$\mathcal{C} = \{C_1, C_2 \dots C_N\} \tag{1}$$

Each collection (see Equation 2) is composed of a set of documents, where M is the number of documents of the i^{th} collection:

$$C_i = \{D_{i1}, D_{i2} \dots D_{iM}\} \tag{2}$$

Each document (see Equation 3) consists of a set of elements, where P represents the number of elements of document D_{ij}, and each element d_{ijk} is a multimedia element (text, audio, image, or video):

$$D_{ij} = \{d_{ij1}, d_{ij2}, \dots d_{ijP}\} \tag{3}$$

The mode of a collection ($\mathcal{M}(\mathcal{C})$) is defined by the type of documents that compose it. The mode of a document ($\mathcal{M}(D)$) is defined by its elements, being monomodal when all of its elements have the same mode, and multimodal if there are at least two elements that have different modes (see Equation 4):

$$\mathcal{M}(D) = \begin{cases} mono\ \forall i,j\ \mathcal{M}(d_i) = \mathcal{M}(d_j) \\ multi\ \exists i,j\ \mathcal{M}(d_i) \neq \mathcal{M}(d_j) \end{cases} \tag{4}$$

where $1 \leq i, j \leq K$ and $\mathcal{M}(d_i) \in \{txt,\ img,\ vid,\ aud,\ conc,\ trip,\ inst\}$ (as described in detail in Section 4.5).

Besides their content, multimedia elements can be annotated using semantic information. Assuming that two documents containing related information are related in some way, both documents can be interconnected using semantic relations. Considering two documents (D_{ij} and D_{xy}), there are two types of semantic relations that can appear between them:

1. A *multimedia relation* (δ^m) relates two different documents or multimedia elements directly, and is represented as $\delta^m(D_{ij}, D_{xy})$. For example, one multimedia relation is *isKeyframeOf*, which relates an image and a video because the image is an extracted keyframe from that video.

2. A *concept-based relation* or *semantic relation* (δ^s) relates two documents indirectly through a semantic concept. A document is related to a concept, represented as $\delta^s(D_{ij}, o)$, where o is a concept of the knowledge-based system.

Both semantic and multimedia relations are used in ontology based REs (see Section 4.4). This enables relations between elements of the collection to be represented, which subsequently allows semantic and exploratory searches.

3.2. Query Modalities

Our model considers multimodal queries that are defined as a set of elements:

$$Q = \{q_1, q_2, \dots, q_K\} \tag{5}$$

where K is the number of elements in the query, and each element q_k is a multimedia object element (text, audio, image, or video).

The modality of the query ($\mathcal{M}(Q)$) is defined by its elements, as shown in Equation 4.

An example of a multimodal query containing text and an image is displayed as follows ($Q_2 = \{q_{21}, q_{22}\}$, where $\mathcal{M}(q_{21}) = txt$ and $\mathcal{M}(q_{22}) = vid$):

'When did the event in the image take place?' together with the image shown in Figure 2.

Figure 2: Example of an image that is part of a multimodal query

3.3. Retrieval Engines

The relationship between queries and documents in collections will determine the retrieval technique to be used. This is determined by considering query keywords in the document, similarities of low-level features (colour, texture, frequency, or movements), or equivalent semantic elements in queries and documents. There are a wide variety of techniques that can be applied (see Section 2.3).

The objective is to allow the integration of any RE. Because of this, an RE is considered as a process or retrieval approach ($\mathcal{P}$) that accesses some collections ($\mathcal{C}$) with a query (Q), and obtains a set of results ($\mathcal{S}$) from them. REs are used as *'black-boxes'*. Equation 6 presents the triplet that represents an RE:

$$RE = [\mathcal{C}, Q, \mathcal{P}] \tag{6}$$

The RE functionality is defined as:

$$\mathcal{S} = \mathcal{P}(Q) \tag{7}$$

where $\mathcal{P}(\bullet)$ represents the retrieval approach of the engine, and $\mathcal{S}$ represents the result set returned when Q is sent to the engine.

Some examples of retrieval techniques are the *vector space retrieval model* (Salton et al., 1975), *content-based image retrieval (CBIR)* (Smeulders et al., 2000), and *semantic search*, which retrieves a set of documents by matching the semantic concepts of the query with those of the documents (Shah et al., 2002).

3.4. Managing Multiple REs using a Handler

Multimodal information retrieval is not limited to querying a single multimodal source, but rather querying several sources is more appropriate, owing to the current distribution of web content. This is based on the fact that many websites specialise in particular content types: Youtube for videos, Flickr or Instagram (instagram.com) for images, Spotify (www.spotify.com/es) or SoundCloud for audio, and Google or Yahoo (es.yahoo.com) for text, although these work also with other modes. The challenge arises when deciding which of the available sources is to be queried for each query.

Our model names this module as the *handler* ($\mathcal{H}$), and defines it as a triplet (see Equation 8) composed of a set of REs ($\mathcal{E}$), the input query (Q), and the handling strategy (Ξ). The handling strategy is in charge of selecting which REs are queried:

$$\mathcal{H} = [\mathcal{E}, Q, \Xi] \tag{8}$$

The functionality of the handling strategy (Ξ) is to provide a subset ($\mathcal{E}'$) of the available REs ($\mathcal{E}$), depending on the formulated query (see Equation 9):

$$\mathcal{E}' = \Xi(\mathcal{E}, Q) \tag{9}$$

The handling strategy is defined as a set of rules. Equation 10 presents the formal definition of a rule:

$$conditions \rightarrow \mathcal{E}' = \{RE_1, \ldots, RE_Z\} \tag{10}$$

where $1 \leq Z \leq L$, L is the number of available REs, and $\mathcal{E}' = \{RE_1, \ldots, RE_Z\}$ is the ordered list of REs that are queried when *'condition'* is met. A *'condition'* is a set of boolean elements. There are three possible types of handling strategy:

- *Parallel Execution:* The handler decides which REs are triggered, and sends the query (or part of the query) to all of these at the same time. The sets of results obtained from each RE are then sent to the fusion module component.
- *Sequential Execution:* The different REs are queried in an ordered manner. As defined in (Galiano, 2011), there are two types of sequential execution: (1) *filtering*, where an RE only retrieves results from the results previously defined as relevant by other Res; and (2) *feedback*, where information extracted from the most relevant results of an RE is used to modify the query sent to the next RE.
- *Hybrid Execution:* This is a combination of parallel and sequential executions. There is a main pipeline execution, as in sequential execution, but instead of executing one RE at each step, a set of REs are executed in parallel.

3.5. Fusion of Results

The set of results for each RE is defined as a vector, which contains a pair document-score for each document in the target collections. A result ($\mathcal{R}$) is represented by a pair document-score ($< D, \gamma >$). Thus, the i^{th} result returned by RE_j is $\mathcal{R}_{ij} =< D_i, \gamma_{ij} >$.

Considering these requirements, the final set of results can be formally defined as a matrix dot product (see Equation 11).

The results obtained from each RE must be combined, in order to obtain a single set of results. To perform this aggregation, a linear combination (Strang, 2006) is used. This linear combination computes the final score for a document as the weighted sum of the scores that returned by each RE for this document:

$$\mathcal{S}_{final} = \mathcal{A} \cdot \mathcal{V} \tag{11}$$

where:

- M represents the number of retrieval engines.
- N represents the number of results defined in Equation 16.
- $\mathcal{A}$ represents a vector containing the weight coefficients of each RE:

$$\mathcal{A} = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_N \end{bmatrix} \tag{12}$$

- $\mathcal{V}$ represents a matrix containing the scores of the results. Each column corresponds to a concrete document from the collections, and each row corresponds to an RE. The intersection of a row and a column stores the score assigned to the document (D_x) by the RE_y (γ_{xy}):

$$\mathcal{V} = \begin{bmatrix} \mathcal{S}_1 \\ \vdots \\ \mathcal{S}_N \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \cdots & \gamma_{1M} \\ \vdots & \ddots & \vdots \\ \gamma_{N1} & \cdots & \gamma_{NM} \end{bmatrix} \tag{13}$$

- $\mathcal{S}_{final}$ represents the final scores of the results as an ordered list:

$$\mathcal{S}_{final} = \begin{bmatrix} \gamma_1^{final} \\ \gamma_2^{final} \\ \vdots \\ \gamma_M^{final} \end{bmatrix} \tag{14}$$

  where γ_m^{final} represents the score of the m^{th} result after combining every set of results for each RE. The score of a result is generalised as:

$$\gamma_i^{final} = \sum_{j=1}^{M} \alpha_j \cdot \gamma_{ji} \quad (1 \leq i \leq N) \tag{15}$$

  where M is the size of the possible results vector (see Equation 16). This size is the sum of the sizes of all of the collections, excluding repeated documents.

The computation of this matrix product is only possible if the set of results for each RE ($\mathcal{S}_n$) has the same length. Therefore, an RE returns zero values for each element in the collections that has not been retrieved. This length is equal to the number of documents (excluding repetitions) that all of the collections contain. This implies that these vectors have a size defined by Equation 16:

$$size(\mathcal{S}_n) = N = size(\bigcup_{j=1}^{K} \mathcal{C}_j) \tag{16}$$

where:

- K is the number of queried REs.
- $\mathcal{C}_j$ represents the document collections used by RE_j.

3.6. User Interactions

A user is part of the system, and its activities (queries, displayed results, timestamps, etc.) are recorded. The information logged from users can be very different, depending on each application or system. The definition of an interaction in the model must consider the registration of as much different information from the user as possible.

Users perform interactions during sessions. A *session (Se)* is defined by an initial and final timestamp, and consists of a set of interactions, because the user enters the system until they disconnect. Similarly, interactions are organized in terms of the *user (U)* who performed them, the *timestamp (ts)* of the moment when they were performed, their *type* ($\mathcal{T}$), and an *additional information* field (Φ). The final attribute can be different for each interaction type. With these considerations, Equation 17 presents a quintuple representing an *interaction* (In):

$$In = (U, Se, ts, \mathcal{T}, \Phi) \tag{17}$$

An example of an interaction is that *'User34'* has *visualised* a result with *id='news008'* from the source *'qa'* that was at position *'3'* of type *'text'* at the moment *'29-11-2013 09:51:04'*. This interaction is registered as:
`(user34, session288, '29-11-2013 09:51:04', visualisation, 'news008-qa-3-text')`

4. Development of an IMIR Prototype in the Sports Domain

In order to validate the proposed model, a prototype is defined using a subset of model components: multimodal information, multimodal query, multiple REs, a handler, fusion of results, and interaction management.

Some of the components were developed during the collaboration in the Buscamedia project (Martínez et al., 2012). *Buscamedia* (CEN-20091026) was a research project aimed at achieving significant progress in the areas of semantics, audiovisual production, and the distribution of rich media, regardless of consumer networks and terminals, with the aim of creating a single semantic multimedia search engine.

As illustrated in Figure 3, the basic architecture of the prototype follows the basic functionality of an IR system. A user sends a query to the system, and then it is sent

to the rule-based handler, which analyses the query and determines its type (text, audio, or a combination of text and image). The handler is in charge of querying the available REs (depending on the query and its type). As explained in Section 3.3, each RE returns a set of results to the handler, which then sends them to the fusion of results module. This module combines, filters, and reranks the results, obtaining a single set of results. Finally, this single set of results is returned to the user.

Full descriptions of these elements and their concrete definitions using the model are provided below.

4.1. Multimedia Collections

TREC (Text Retrieval Conference) and CLEF (Cross-Language Evaluation Forum) collections were analysed in order to be integrated in this prototype, but they do not integrate multimedia objects with semantic relationships between documents. Owing to the fact that this research has been performed in the Buscamedia research project, a Spanish collection regarding sports that was generated within the project was employed. This collection is known as *'Sports20'*, and is multidomain, covering football, basketball, and formula one sports. It has been supplied by the content provider partner. The data was obtained during October 2010, and it is composed of four subsets of documents in different modes:

- 9245 textual news items (compiled from various newspapers), each consisting of a title, subtitle, and body: $C_1 = \{D_{1,i}\}$, where $1 \leq i \leq 9245$ and $\mathcal{M}(D_{1,i}) = txt$.
- 33 videos (sports newscasts): $C_2 = \{D_{2,j}\}$, where $1 \leq j \leq 33$ and $\mathcal{M}(D_{2,j}) = vid$. These videos contain manually generated transcriptions. These transcriptions of the videos do not contain descriptions of the images and news, they are simply transcriptions of the audio.
- 659 images (key-frames extracted from the videos): $C_3 = \{D_{3,k}\}$, where $1 \leq k \leq 659$ and $\mathcal{M}(D_{3,k}) = img$. Each video has been processed to extract images based on a scene detection algorithm, applied by a partner of the Buscamedia project.
- 1191 semantic concepts (the semi-automatic population of an ontology, which is explained in Section 4.2): $C_4 = \{D_{4,x}, D_{4,y}\}$, where $1 \leq x \leq 1191$, with $\mathcal{M}(D_{4,x}) = conc$, $1 \leq y \leq 1590$, and $\mathcal{M}(D_{4,y}) = inst$.

4.2. Semantic Resources: Ontology

The prototype takes advantage of a semantic search using a multidomain ontology with a double functionality. It semantically relates the documents of the collections, and is an RE (see Section 4.4).

This ontology is a specialized domain-specific ontology of the sports domain, which contains multilingual documents in Spanish, Catalonian, and English. It is composed of 30 smaller ontologies, with a total of 1191 classes, 722 properties that relate objects, and 338 data properties. Furthermore, it is populated with 1590 individuals. The ontology contains 94 multimedia relations, and 1735 semantic relations.

4.3. Query Modalities

In this prototype, the following three query modalities proposed by the users at the Buscamedia project are implemented:

- *Text query:* The query is a text, ranging from a single token to a complete sentence:

$$Q_{text} = \{q_1, \ldots, q_i\} \tag{18}$$

  where q_i are text tokens.

- *Voice query:* The query is an audio file containing a spoken query:

$$Q_{voice} = \{q_1\} \tag{19}$$

  where $\mathcal{M}(q_1) = aud$. Once the spoken query has been transcribed, it is handled as a text query.

- *Textual and image query:* The query is a combination of a text query and an image:

$$Q_{text-image} = \{q_1, \ldots, q_M, q_{M+1}\} \tag{20}$$

  where:

  - $\mathcal{M}(q_i) = txt \ \forall i \in [1, M]$
  - $\mathcal{M}(q_{M+1}) = img$

4.4. Retrieval Engines

The definitions of the REs are based on specifications of the Buscamedia project. Three information retrieval processes and three preprocessing modules have been integrated into the prototype (see Figure 3).

4.4.1. Information Retrieval

Three information retrieval systems that process the query and return information extracted from the collections of documents have been implemented in the prototype:

1. *Question answering search* (QAS) makes a comparison between the query and the documents in the collections, and extracts an answer from the most relevant. It returns a set of results, containing concrete answers and documents supporting them. This engine retrieves information using SOLR-LUCENE (Smiley and Pugh, 2009). In addition, it performs

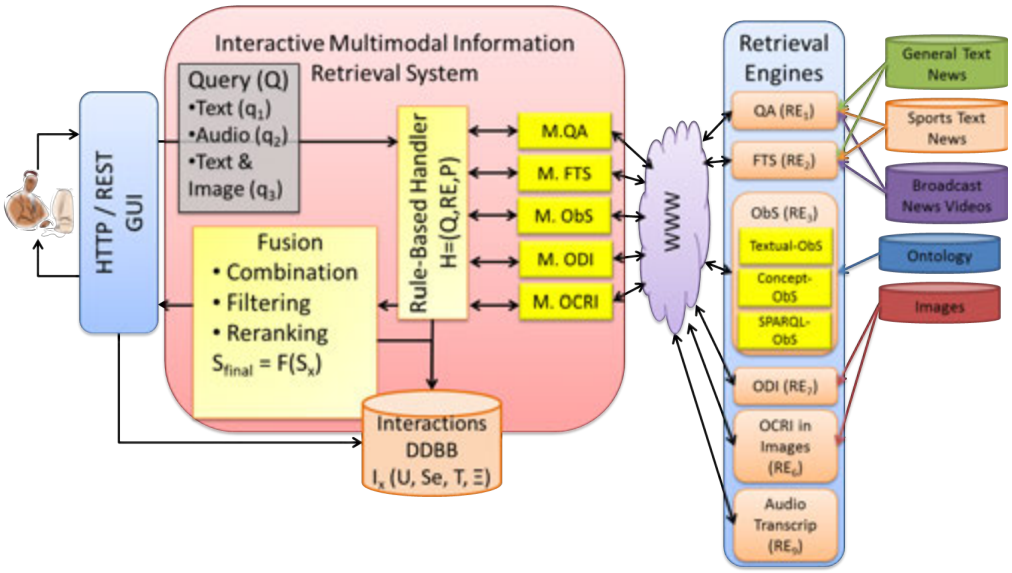

Figure 3: Architecture of the IMIR prototype

morphological tagging, syntactic analysis, named entity recognition, semantic tagging, and classification of the query. The final answers are obtained by a process of answer extraction and re-ranking from the retrieved documents. For the linguistic analysis the proprietary technology MeaningCloud[2] is employed:

$$RE_1 = (C_1, Q_{text}, \mathcal{P}_1) \tag{21}$$

2. *Full text search* (*FTS*) functions as a classic keyword-based text retrieval method. It returns a set of textual documents that contain the keywords of the query. The engine uses BM25F (Pérez-Iglesias et al., 2009), with the same push factors for information retrieval, and Snowball analysis (Porter, 2001) (for removal of stop words, and stemming and removal of special characters and punctuation):

$$RE_2 = (C_1, Q_{text}, \mathcal{P}_2) \tag{22}$$

3. Ontology-based search (*ObS*) offers three different ways of searching inside the ontology:

   (a) *Textual search* (*Textual-ObS*): This uses a text query to retrieve concepts from the ontology, searching over textual metadata properties of the ontology, namely titles and descriptions. It processes the query linguistically using language identification and cleaning, tokenisation, entity extraction using dictionaries, linguistic annotation, and partition judgment.

   Furthermore, named entity recognition is performed using Linked Open Data (LOD) (linkeddata.org) dictionaries. The metadata is added to a Lucene index, which is queried with the query:

$$RE_3 = (C_4, Q_{text}, \mathcal{P}_3) \tag{23}$$

   (b) *Concept search* (*Concept-ObS*): This retrieves all information from the ontology (individuals, classes, etc.) that is related to the concept received as input:

$$RE_4 = (C_4, Q, \mathcal{P}_4) \tag{24}$$

   (c) *SPARQL-based Search* (*SPARQL-ObS*): This allows the use of SPARQL (Prud'hommeaux and Seaborne, 2008) queries against the ontology. This search engine returns a set of results consisting of ontology triplets:

$$RE_5 = (C_4, Q, \mathcal{P}_5) \tag{25}$$

4.4.2. Preprocessing Modules

There are three preprocessing modules implemented in the prototype devoted to analyse and transform queries:

1. *OCR in images (OCRI)* receives an image, and extracts the existing text about it. The result returned is the text present in the image (subtitles, text boxes, text on logos, etc.). Firstly, the preprocessor identifies the areas that possibly contain text. These areas are known as *pills*. The pills are obtained by applying the Homogeneous Texture Descriptor (HTD) (Manjunath et al., 2001). False positives are checked by applying two classifiers based on support vector machine (SVM) techniques (Burges, 1998). If both classifiers deny the pill, then it is not considered as containing text. Once the pills have been identified, a sequence of tokens is generated using a free OCR

[2]https://www.meaningcloud.com/es/ accessed at 16/07/2016

software package called Tesseract (Smith, 2007). For example, if Figure 2 is used as a query, then the result of this preprocessing module should be: *'SALAMANCA, ESTA MAÑANA. Susto monumental en El Helmántico por el desmayo de Miguel García'.*

2. *Object detection in images (ODI)* extracts the existing objects in the query image, and returns a set of concepts represented as terms. It employs the visual attention algorithm proposed in (Itti and Koch, 2000), which detects a set of specific locations over the entire image, and establishes the order in which visual attention is circulated through them. Following this, an algorithm based on SURF (Bay et al., 2008) is applied to select interesting objects. For example, if Figure 2 is used as a query, then the result should be: *'field, football player, referee'.*

3. *Audio transcription (AT)* transcribes the incoming audio file. It returns a textual transcription, together with temporal information. It uses Windows Speech Recognizer (WSR) version 5.1[3] and Dragon Naturally Speaking (DNS) version 12.5.1[4] to perform the audio transcription.

   For a complete description of this RE, we refer the reader to (Schneider et al., 2009), (González et al., 2013), and (Schneider et al., 2014).

4.5. Orchestrating REs (Handler)

A handler is required to decide which REs will be queried for each query. The handler implemented for this prototype is based on rules. Every rule consists of a number of conditions (the left side of the assignment) and a subset ($\mathcal{E}'$) of the available REs ($\mathcal{E}$) (the right side of the assignment).

$$Conditions \rightarrow \mathcal{E}' \tag{26}$$

The rules implemented in the prototype use two attributes in the conditions: the mode ($\mathcal{M}(Q)$) and type ($\Psi(Q)$) of the query.

$\mathcal{M}(Q)$ can take the values *txt* (text query), *aud* (audio file), *vid* (video file), *img* (image file or content), *conc* (semantic concept - textual identifier), *trip* (semantic triplet - rdf format), and *inst* (semantic concept instance identifier). $\Psi(Q)$ can take the values * (every query), *question* (a complete question), *short* (text query with three or fewer tokens), *long* (text query with more than three tokens), *voice* (the query is an audio file), and *multi* (queries combining text and image):

$$\begin{array}{l} \mathcal{M}(Q) = value \\ \quad and \qquad \rightarrow \mathcal{E}' = \{RE_1, \dots RE_Z\} \\ \Psi(Q) = value \end{array} \tag{27}$$

where $\mathcal{E}' = \{RE_1, \dots RE_Z\}$ is the subset consisting of *REs* that are queried.

Two different handlers have been implemented in the prototype.

1. The first handler queries every available RE: $\mathcal{H}_1 = (\mathcal{E}, Q, \Xi_1)$ where $\Xi_1\{* \rightarrow \mathcal{E}' = \{RE_1, \dots, RE_N\}\}$.

2. The second handler is a heuristic rule-based strategy, supported by predefined rules. The following rules have been defined by experts:

   (a) Only text as query ($\mathcal{M}(Q) = txt$): Three rules are defined, depending on the query type, which are *question* (a complete question), *short* (three or fewer tokens), and *long* (more than three tokens). Analysing the first queries processed by our baseline system, we found that most of the queries with three or fewer tokens triggered documents or websites, while longer queries triggered more specific information. We are simply counting the tokens, and not deleting the stopwords. This is because most of the queries do not contain stopwords if they are short, and stopwords are important for longer queries, such as in questions:

   $$\begin{array}{c} \Psi(Q) = question \rightarrow \{QAS, FTS\} \\ \Psi(Q) = long \rightarrow \{FTS\} \\ \Psi(Q) = short \rightarrow \{FTS, ObS\} \end{array} \tag{28}$$

   (b) Voice query ($\mathcal{M}(Q) = aud$): The query file is transcribed using an automatic transcription service, and then the resulting text is treated as a regular text query:

   $$\Psi(Q) = voice \rightarrow \{AT\} \tag{29}$$

   (c) Multi query ($\Psi(Q) = multi$): The query is divided into two parts, text and image. The text is treated as an independent text query, while the image is analysed using the image preprocessing modules (OCRI and ODI), obtaining text that is later also managed as a text query.

4.6. Heterogeneous Results Management: Fusion of Results

The use of multiple REs requires the implementation of a module for fusing the results (as explained in Section 3.5). This module receives all of the results obtained from each RE, and obtains a single homogeneous set of results.

Only one fusion module is implemented, and this is based on a *Round Robin strategy* (Silberschatz et al., 2008). The formal definition of the Round Robin strategy is displayed in Equation 30, which determines the final position of the j^{th} result of the i^{th} RE ($D_{i,j}$):

$$rank(D_{i,j}) = (N_E \cdot j + i) - N_E \tag{30}$$

where N_E is the number of sets of results that are combined.

[3] http://msdn.microsoft.com/en-us/library/ms723627(v=vs.85).aspx
[4] http://www.nuance.com/dragon/index.htm

4.7. Graphical User Interface

We require a graphical user interface (GUI), to make our method usable by users. Figure 4 displays a screenshot of the querying interface with the available search modalities. There are four clearly defined parts, marked by numbers. (1) represents the textual query box; (2) marks the textual and image query box; (3) indicates the voice query box; and (4) denotes the lateral navigation menu, which allows navigation through the different graphical interfaces.

Figure 4: Screenshot of query boxes implemented in the prototype

With regard to the system output, Figure 5 presents a list of results for the textual query *'Barcelona'*. This list contains results of two types: semantic concepts and news documents. The available or related multimodal content (videos, images, texts, or audio) appears at the bottom of each result (marked with red box at the bottom of the figure). The relevance feedback is performed by the three coloured smilies (green, orange, and red) on the upper right corner, each one symbolising a relevance value.

Figure 5: Screenshot of the prototype showing results list for textual query *'Barcelona'*

5. Adapting IR Functionality based on User Interactions

The main objective of this study is to adapt the functionality of the handler and the results combination module. In order to achieve this, user interactions are used to improve the retrieval performance.

The rules used by the handler of the basic prototype were manually defined using the query properties. The modification of the functionality is based on the analysis of the past interactions of users, which are analysed and processed in order to generate new rules that represent user behaviour.

Decision trees (Cintra et al., 2013), multilayer perceptron (Gutiérrez et al., 2010), and simple K-means (Kanungo et al., 2002) algorithms have been studied in connection with behaviour pattern classification, because of their well-known efficiency.

The input of the algorithm is the user query and past interactions. Meanwhile, the output of the algorithm is an ordered set of $RESs$.

The notation of the interactions is described in the formal model. The different types of interactions ($\mathcal{T}$ in Equation 17) and their associated information (Φ in Equation 17) are shown in Table 1.

Type $\mathcal{T}$	Associated Information Φ	Action
REG	-	Registration
LOG	IN, OUT	Log in/out
PRESS	$\mathcal{W}$	Pressing GUI element (button, link, etc.)
SEARCH	Q	Search a query
VIEW	LIST, CLUSTER, GROUP, DOCUMENT	Changing visualisation
DOC	$R_{i,j}$	Document visualisation
RELEV	GOOD, BAD; NEUTRAL and $R_{i,j}$	Relevance judgment of a document

Table 1: Interaction types and associated information to be used in adapting IR functionality

Rule-generation models require a training set to classify future queries. The training set to be provided to the rule-generation module is composed of two types of information: the information used to specify the characteristics of the query (the query features) and the labelled class associated to the features of this query.

The query features consist of its linguistic characteristics:

1. *Mode (m)*: The mode of the query, with value *'t' (text)*, *'a' (audio)*, or *'ti' (text and image)*.
2. *Type (t)*: The type of the query, with value *'Question'*, *'Short'*, *'Long'*, or *'Concept'*.

3. *Length (l)*: Number of tokens of the query. We simply count the tokens, and do not delete the stopwords. As explained before, this is because most queries do not contain stopwords if they are short, and stopwords are important for longer queries, such as in questions.

4. *Named entities in the query (e)*: In this approach, we decided to use information regarding named entities in queries. These named entities can be PERSON, LOCATION, ORGANISATION, etc. These entities are extracted with the commercial tool MeaningCloud.

5. *Number of named entities (n_e)* present in the query, analysed with the MeaningCloud tool.

6. *Number of verbs (n_v)*, analysed using the MeaningCloud Part-Of-Speech tagger. We consider the number of verbs, rather than the number of other content words, because after named entities, verbs are the words containing the most semantic information for the query.

7. *Topic (o)*: Topic of the query, extracted using MeaningCloud Topics Extraction.

As an example, Figure 6 presents a graphical representation of a regular text query together with its characteristics.

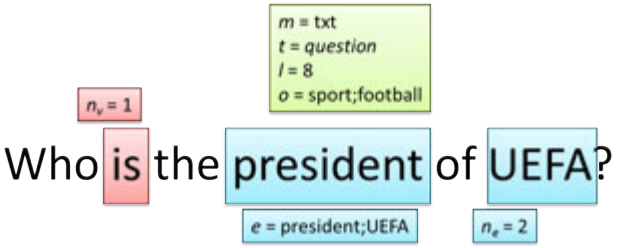

Figure 6: Example showing a question as a query with its features

The final element required for training the classification algorithms is the class of each training data. In this case, this class is an ordered list of engines whose label is represented as an ordered list of REs.

We base our approach on the work of Balog (2013) and Pal and Mitra (2013), which define the score of an RE as a linear combination of the following three scores: (1) *context score*, referring to the environmental characteristics of the RE; (2) *content score*, referring to the similarity between the content of the collections used by the RE; and (3) *past users behaviour score*, referring to the actions (recorded as interactions) that have previously been performed by users.

Our approach only considers past behaviour, and the resulting simplified equation is presented in Equation 31:

$$\alpha_i = score^i_Q = \frac{\sum_{j=1}^{N} score(d_{ij}, Q)}{N} \quad (31)$$

where $score(d_{ij}, Q)$ is the score of a document d_{ij} with respect to the input query Q, and N is the number of considered documents.

The order of the engines is computed using user interactions and the scores described in Table 2.

6. Experimental Setups of IR Adaptation based on User Interactions

The complete evaluation process of the prototype took two months to complete (April and May 2013), and 233 users participated in it. A total of 981 queries were gathered. Each user made an average of $4,58$ (max. 37 and min. 1) searches per session.

We could have instructed the users to try every query mode, but we wanted to obtain an unconstrained evaluation and also check the type of queries that users wanted to employ independently.

Although text is the most intuitive way of searching for information (based on current IR systems), we expected to find more users trying voice and combined (text + image) queries. Only $9,9\%$ of searches used voice queries, and this was even lower for combined queries $(3,6\%)$. The majority of the searches consisted of textual queries $(86,5\%)$.

These interactions have been used to generate models (decision trees, multilayer perceptron, and K-means) that are used to generate rules for the handler.

The validation of the functionality adaptation is defined as a Cranfield experiment (Project and Cleverdon, 1962). It is composed of the definition of a silver standard corpus, and the application of techniques (scores for ranking REs) and algorithms (classification models) to the interactions, which have been defined in the silver standard by obtaining a set of rules for the handler and analysing the results.

A silver standard follows the same concept as a gold standard, with two difference, the relevance judgments are assigned after querying the RE, and not all documents in the collection are assigned a relevance judgment. In fact, not even all of the documents returned by the retrieval system receive a judgment. Relevance judgments are provided to only the N top results (typically N={2, 5, 10, 20, 50, ...}).

Therefore, in order to simplify the readability of the results, acronyms are assigned to each classification algorithm, RE score, and query feature.

The new rules are obtained by querying the model with every possible query feature and obtaining the corresponding ordered list of *REs*. The rules are ordered by the features of the query, and so they do not have any relevance value.

Name	Equation	Description
Interactions-based score	$score(d_{ij}, Q) = \begin{cases} 1 \; if \; interaction \; exists \; over \; doc \; d_{ij} \\ 0 \; otherwise \end{cases}$	
Lowest-position score	$score(d_{ij}, Q) = \min_j \frac{1}{rank(d_{ij}, Q)}$	$rank(d_{ij}, Q)$ is the ranking of d_{ij} within the RE_i result list
Average-position score	$score(d_{ij}, Q) = \frac{1}{rank(d_{ij}, Q)}$	$rank(d_{ij}, Q)$ is the ranking of document d_{ij} in the list of results from RE_i.
Combined score	$score(d_{ij}, Q) = \frac{1}{1+rank(d_{ij}, Q)} \cdot \frac{1}{\log(1+iteration(d_{ij}, Q))}$	$rank(d_{ij}, Q)$ is the ranking of d_{ij} within the list of results, and $iteration(d_{ij}, Q)$ is the number of interactions made over d_{ij}. This equation has been taken from (Womser-Hacker, 1996), and we adapted it by adding $log(\cdot)$ to also consider the decrease resulting from not being the first 'used' result

Table 2: Scores for generating ordered lists of REs

Only one example of rules is presented in this article, but a more detailed description of the results is given in (Schneider, 2015).

An example of a set of rules is provided in Table 3, considering "query mode" and "question type" features. The displayed set of rules were obtained by the j48 implementation of the decision tree C4.5 algorithm (*'J4.8'*), the mode and type of the query (*'mt'*) and the rankings of *REs* are determined by the first-used score (*'FUS'*). It can be observed that there are some rules that return the same ordered list.

```
qmode=t;qtype=question; -> qa,ft,ont
qmode=t;qtype=short; -> ont,qa,ft
qmode=t;qtype=long; -> ont,qa,ft
qmode=t;qtype=concept; -> ont,qa,ft
qmode=ti;qtype=long; -> ft,qa,ont
qmode=ti;qtype=question; -> ft,qa,ont
qmode=ti;qtype=short; -> ont,qa,ft
```

Table 3: Rules obtained by decision trees (*'J4.8'*), with the mode and type of the query (*'mt'*) and the rankings of *REs* determined by the first-used score (*'FUS'*).

6.1. Analysis of Results for Different Approaches

Once the rules have been obtained for every possible combination of the adaptation algorithm, we proceeded to assess them. The *normalised discounted cumulative gain (NDCG)* values (as shown in Table 4) measure the usefulness of a document based on its position in the results list.

The results for these combinations of the algorithm offer small improvements over the baseline. The baseline is the prototype, using the rules predefined by experts in the first evaluation. The comparison is performed using the following four algorithms. *Probs* is a simple probability-based method, which does not use any classification techniques or algorithms, but considers the probabilities for the use of every source. The C4.5 decision tree algorithm is labelled as *J4.8*, the multilayer perceptron technique is referred to as *MLP*, and the simple K-means (two groups) algorithm is named as *SKM2*.

We compare the NDCG measure obtained for two different approaches: the multimodal system using predefined rules, and the same multimodal system after the functionality is adapted by past user interactions. The NDCG achieves an improvement ranging from $-2,92\%$ and $2,81\%$, depending on the approach used. We have considered three features to classify the approaches, namely (i) the classification algorithm, (ii) the query features, and (iii) the scores for computing the orders of REs.

It is interesting to note that there are some combinations that perform worse than the baseline. These cases occur when the query information is too simple, consisting of only the mode or the mode and type. Query classification fails because the information regarding the query is too generic, and the model classifies very different queries as similar. A question and a concept are two completely different queries, but both have the same mode (text). This indicates that the more effectively the model can sort the query, the better the results will be, and thus the better the rules. This occurs in the final case. The results become worse when the topic is added to the query features. This may be because of two reasons. Either the topics are not well allocated and are introducing noise, or the topics are so generic that they spoil the classification of the queries. One case did not return any results, possibly due to a problem in the execution during the evaluation.

The results demonstrate that the IR performance can be improved by considering user behaviour information (in

Ranking Score	Algorithm	m	mt	mtl	mtle	mtleNe	mtleNeNv	mtleNeNvT
Prototype		79.31						
Interactions-based score (IbS)	Probs	79.22	79.38	80.72	80.53	79.62	80.61	80.71
	J48	79.21	80.59	80.71	80.24	80.72	80.44	80.33
	MLP	80.34	80.64	80.3	80.13	80.84	81.38	80.78
	SKM2	76.99	79.3	80.05	80.77	79.68	79.95	80.17
Lowest-Position score (LPS)	Probs	78.38	80.96	80.08	80.13	80.46	80.84	79.95
	J48	79.96	80.02	80.21	80.07	80.52	81.21	80.35
	MLP	79.66	79.52	80.06	81.05	80.63	80.91	80.15
	SKM2	77.87	80.59	80.58	79.46	80.5	79.55	80.2
Averaged-Position score (APS)	Probs	80.31	80.84	80.03	81.54	79.73	80.83	80.05
	J48	79.06	80.02	80.16	80.67	80.7	80.38	0.0
	MLP	78.83	80.56	80.65	79.89	80.38	80.76	80.06
	SKM2	78.65	80.71	80.58	79.77	79.22	79.54	79.6
Combined score (CS)	Probs	79.85	80.28	81.18	80.29	80.05	80.15	79.79
	J48	79.12	80.02	80.1	79.59	79.68	80.52	81.33
	MLP	79.23	80.43	80.37	80.78	80.28	80.38	80.9
	SKM2	78.28	78.47	80.13	79.39	79.58	80.18	80.49

Table 4: NDCG for ranking scores, machine learning algorithm, and query types. The red cell represents the worst result, while the green cell represents the best, among all the combinations of algorithms, features, and scores.

this case past interactions). The numeric results also indicate that the IR performance improvements are limited. This is because of the fact that the IR performance of every individual engine was comparable with the state-of-art systems by themselves. Therefore, the combination of a set of REs with those performances can only improve slightly when employing them in combination.

We applied statistical significance tests to the measures (see Table 5). In this case, we have used the t-Student test. In order to assure the correctness of the significance test, we have grouped the results into different vectors, to apply the t-test in every group. The first grouping has been generated by creating a vector for every row of Table 4. The second grouping combines all of the results for every algorithm (*Probs*, *J48*, *MLP*, and *SKM*). The third grouping is composed of vectors containing all of the results for every score (*IbS*, *LPS*, *APS*, and *CS*). The fourth grouping collects the results of every query mode, i.e., the columns of Table 4.

It can be observed in Table 5 that there are some cases where the p-value is higher than 0.05. Thus, these cases can be considered as insignificant. However, if a closer analysis is performed for these cases, it can be noted that there are four cases where the value is influenced by the execution that returned a value of 0.0 in Table 4. If we manually change the 0.0 value to 80.05 (adopted from the previous algorithm, simply to prove its effect), then the p-values for these four approaches decrease (0.0035, $1.18e^{-09}$, $2.99e^{-07}$, $1.02e^{-07}$).

The other four values are those associated to the simple K-means algorithm. The last p-value that is higher than 0.05 is associated with the results obtained using the query features mode (*'m'*). This occurs because using only this mode results in an ineffective query classification.

The best result is obtained using the probability-based classification algorithm (*Probs*), with the ranking of REs generated with averaged-position score (*APS*), and the mode, type, length, and entities of the query considered (*mtle*). Here, the NDCG value is *81,54%*. By contrast, the worst approach uses the K-means classification algorithm (*SKM2*) and considers the mode of the query (*m*), with the ranking of REs generated with the interactions-based score (*IbS*). This achieved an NDCG of *76,99%*.

7. Conclusions and Future Research

The objective of this study was to define a formal model that aids with the definition of a multimodal retrieval system, and allows a standardised design of multimodal IR components. Furthermore, we aimed to implement a basic multimodal IR prototype, based on the previously defined model, which is composed of elements that are easily replaceable by others that have been similarly defined by the model. Finally, the prototype was extended to adapt its functionality to past user interactions, in order to satisfy the need to create a multimodal IR system that adapts its functionality to user behaviour.

Regarding the adaptation of the multimodal IR, the best result was obtained using a probability-based classification algorithm (*Probs*), with the ranking of REs generated using an averaged-position score (*APS*), where the mode, type, length, and entities of the query were considered (*mtle*). Its NDCG value is *81,54%*. By contrast, the worst approach used a K-means classification algorithm (*SKM2*) and considered the mode of the query (*m*), with the ranking of REs generated using an interactions-based score (*IbS*). This achieved an NDCG of *76,99%*.

Grouping	p-Value							
Rows of the table	9.87e-03	1.10e-03	8.51e-05	3.04e-01	2.37e-02	3.87e-04	2.51e-03	1.09e-01
	1.32e-03	8.05e-01	7.40e-03	9.10e-02	9.67e-04	1.69e-02	1.19e-03	2.87e-01
Algorithms	2.16e-08	7.49e-01	9.51e-11	2.95e-02				
Scores	5.40e-06	1.16e-06	7.59e-01	5.12e-06				
Columns of the Table	8.59e-01	9.24e-05	9.68e-10	5.61e-06	3.43e-06	1.13e-07	7.81e-01	

Table 5: p-values of the statistical significance tests. The red cells are those with p-values higher than 0.05

The first remarkable conclusion is that the small improvements result from good performances achieved when the REs are employed by themselves. That is, when only a single RE is employed. Therefore, the combination of several REs cannot result in a big improvement, because there is a limited improvement margin in such a cases.

The application of the lessons learned here (model, prototype, adaptation) to new domains appears to be the most promising line of future research, if commercial applications are considered. There are two domains for which this work fits:

1. *Second screen* is a second electronic device used by television viewers to connect to a program they are watching. A second screen is often a smartphone or tablet, where a special complementary app may allow the viewer to interact with a television program in a different way — the tablet or smartphone becomes a TV companion device. The second screen phenomenon represents an attempt to make TV more interactive for viewers, and help promote social buzz around specific programs. This is becoming popular for users watching television. The Digital Consumer Report 2014 Nielsen[5] claims that 66% of tablet and 49% of smartphone owners surf the web while watching TV. Among the most common usages are shopping, checking sports scores, emailing/texting friends regarding the program, and looking up information regarding actors, plotlines, or athletes.

2. *Health social media streams analysis* is a domain that is currently attracting significant research attention (Martínez et al., 2016). This refers to the application of text analysis techniques to social media streams with health content. This domain is interesting in relation to multimedia retrieval, because it handles many different information modes, including clinic reports (text), X-ray (images), and ultrasound (video). Time constraints are highly important. The faster a doctor or a patient obtains information, the more effective the treatment can be.

[5] http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014%20Reports/the-digital-consumer-report-feb-2014.pdf

Acknowledgments

This work was partially supported by eGovernAbility-Access project (TIN2014-52665-C2-2-R).

References

Agichtein, E., Brill, E., Dumais, S., 2006. Improving Web Search Ranking by Incorporating User Behavior Information. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '06. ACM, New York, NY, USA, pp. 19–26.
URL http://doi.acm.org/10.1145/1148170.1148177

Arampatzis, A., Zagoris, K., Chatzichristofis, S. A., 2011. Fusion vs. two-stage for multimodal retrieval. In: Proceedings of the 33rd European Conference on Advances in Information Retrieval. ECIR'11. Springer-Verlag, Berlin, Heidelberg, pp. 759–762.
URL http://dl.acm.org/citation.cfm?id=1996889.1996996

Baeza-Yates, R., Ribeiro-Neto, B., 2011. Modern Information Retrieval: The Concepts and Technology behind Search (2nd Edition) (ACM Press Books), 2nd Edition. Addison-Wesley Professional.

Balog, K., 2013. Collection and Document Language Models for Resource Selection. In: Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013.

Balog, K., Neumayer, R., Nørvåg, K., 2012. Collection ranking and selection for federated entity search. In: Proceedings of the 19th international conference on String Processing and Information Retrieval. SPIRE'12. Springer-Verlag, Berlin, Heidelberg, pp. 73–85.
URL http://dx.doi.org/10.1007/978-3-642-34109-0_9

Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., 2008. Speeded-Up Robust Features (SURF). Comput. Vis. Image Underst. 110 (3), 346–359.
URL http://dx.doi.org/10.1016/j.cviu.2007.09.014

Bellogin, A., Gebremeskel, G. G., He, J., Said, A., Samar, T., de Vries, A. P., Lin, J., Vuurens, J. B. P., 2013. CWI and TU Delft Notebook TREC 2013: Contextual Suggestion, Federated Web Search, KBA, and Web Tracks. In: Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013.

Benavent, X., García-Serrano, A., Granados, R., Benavent, J., de Ves, E., 2013. Multimedia Information Retrieval based on Late Semantic Fusion Approaches: Experiments on a Wikipedia Image Collection. IEEE Transactions on Multimedia Journal.
URL http://dx.doi.org/10.1109/TMM.2013.2267726

Bota, H., Zhou, K., Jose, J. M., Lalmas, M., 2014. Composite Retrieval of Heterogeneous Web Search. In: Proceedings of the 23rd International Conference on World Wide Web. WWW '14. ACM, New York, NY, USA, pp. 119–130.
URL http://doi.acm.org/10.1145/2566486.2567985

Brin, S., Page, L., 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. Comput. Netw. ISDN Syst. 30 (1-7), 107–117.
URL http://dx.doi.org/10.1016/S0169-7552(98)00110-X

Burges, C. J. C., 1998. A Tutorial on Support Vector Machines for Pattern Recognition. Data Min. Knowl. Discov. 2 (2), 121–167.
URL http://dx.doi.org/10.1023/A:1009715923555

Camargo, J. E., González, F. A., 2016. Multimodal Latent Topic Analysis for Image Collection Summarization. Inf. Sci. 328 (C), 270–287.
URL http://dx.doi.org/10.1016/j.ins.2015.08.044

Chernov, S., Kohlschütter, C., Nejdl, W., 2006. A plugin architecture enabling federated search for digital libraries. In: Proceedings of the 9th international conference on Asian Digital Libraries: achievements, Challenges and Opportunities. ICADL'06. Springer-Verlag, Berlin, Heidelberg, pp. 202–211.
URL http://dx.doi.org/10.1007/11931584_23

Cintra, M. E., Monard, M. C., Camargo, H. A., 2013. A Fuzzy Decision Tree Algorithm Based on C4.5. Mathware & Soft Computing Magazine. The Magazine of the European Society for Fuzzy Logic and Technology 20, 56–62.
URL http://www.eusflat.org/msc/docs/vol20n1_brasil4.pdf

Daras, P., Axenopoulos, A., Darlagiannis, V., Tzovaras, D., Bourdon, X. L., Joyeux, L., Verroust-Blondet, A., Croce, V., Steiner, T., Massari, A., Camurri, A., Morin, S., Mezaour, A.-D., Sutton, L., Spiller, S., 2011. Introducing a unified framework for content object description. IJMIS 2 (3/4), 351–375.
URL http://dblp.uni-trier.de/db/journals/ijmis/ijmis2.html#DarasADTBJVCSMCMMSS11

Demeester, T., Trieschnigg, D., Nguyen, D., Hiemstra, D., 2013. Overview of the TREC 2013 Federated Web Search Track.

Demner-Fushman, D., Antani, S., Simpson, M. S., Thoma, G. R., 2012. Design and Development of a Multimodal Biomedical Information Retrieval System. JCSE 6 (2), 168–177.
URL http://dblp.uni-trier.de/db/journals/jcse/jcse6.html#Demner-FushmanAST12

Dwork, C., Kumar, R., Naor, M., Sivakumar, D., 2001. Rank Aggregation Methods for the Web. In: Proceedings of the 10th International Conference on World Wide Web. WWW '01. ACM, New York, NY, USA, pp. 613–622.
URL http://doi.acm.org/10.1145/371920.372165

Galiano, M. C. D., 2011. Recuperación de información multimodal basada en integración de conocimiento. Ph.D. thesis.

Golovchinsky, G., Diriye, A., 2011. Session-based search with Querium. In: HCIR 2011.

González, M., Moreno Schneider, J., Martínez, J. L., Martínez, P., 2013. An Illustrated Methodology for Evaluating ASR Systems. In: Proceedings of the 9th International Conference on Adaptive Multimedia Retrieval: Large-scale Multimedia Retrieval and Evaluation. AMR'11. Springer-Verlag, Berlin, Heidelberg, pp. 33–42.
URL http://dx.doi.org/10.1007/978-3-642-37425-8_3

Görg, C., Kihm, J., Choo, J., Liu, Z., Muthiah, S., Park, H., Stasko, J., 2010. Combining Computational Analyses and Interactive Visualization to Enhance Information Retrieval. In: 2010 Workshop on Human-Computer Interaction and Information Retrieval, New Brunswick, NJ.

Guan, F., Xue, Y., Yu, X., Liu, Y., Cheng, X., 2013. ICTNET at Federated Web Search Track 2013. In: Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013.

Gutiérrez, P. A., Hervás-Martínez, C., Lozano, M., 2010. "Designing Multilayer Perceptrons using a Guided Saw-tooth Evolutionary Programming Algorithm". Soft Computing 14 (6), 599–613.

Hauptmann, A. G., Jin, R., Ng, T. D., 2002. Multi-modal information retrieval from broadcast video using OCR and speech recognition. In: Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries. JCDL '02. ACM, New York, NY, USA, pp. 160–161.
URL http://doi.acm.org/10.1145/544220.544252

Hong, D., Si, L., 2012. Mixture model with multiple centralized retrieval algorithms for result merging in federated search. In: Hersh, W. R., Callan, J., Maarek, Y., Sanderson, M. (Eds.), SIGIR. ACM, pp. 821–830.
URL http://dblp.uni-trier.de/db/conf/sigir/sigir2012.html#HongS12

Hu, X., Kando, N., Yuan, X., 2011. User Evaluation of an Interactive Music Information Retrieval System. In: In Proceedings of HCIR 2011 Workshop, Mountain View, CA, USA.

Huiskes, M. J., Lew, M. S., 2008. The MIR Flickr Retrieval Evaluation. In: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval. MIR '08. ACM, New York, NY, USA, pp. 39–43.
URL http://doi.acm.org/10.1145/1460096.1460104

Itti, L., Koch, C., 2000. A Saliency-Based Search Mechanism for Overt and Covert Shifts of Visual Attention. Vision Research 40, 1489–1506.

Jou, B., Li, H., Ellis, J. G., Morozoff-Abegauz, D., Chang, S.-F., 2013. Structured Exploration of Who, What, when, and Where in Heterogeneous Multimedia News Sources. In: Proceedings of the 21st ACM International Conference on Multimedia. MM '13. ACM, New York, NY, USA, pp. 357–360.
URL http://doi.acm.org/10.1145/2502081.2508118

Kalpathy-Cramer, J., Müller, H., Bedrick, S., Eggel, I., García Seco de Herrera, A., Tsikrika, T., 2011. The CLEF 2011 medical image retrieval and classification tasks. In: Working Notes of CLEF 2011 (Cross Language Evaluation Forum).

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., Wu, A. Y., 2002. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 881–892.
URL http://dx.doi.org/10.1109/TPAMI.2002.1017616

Lana-Serrano, S., Villena-Román, J., Cristóbal, J. C. G., 2011. DAEDALUS at ImageCLEF Medical Retrieval 2011: Textual, Visual and Multimodal Experiments. In: Petras, V., Forner, P., Clough, P. D. (Eds.), CLEF (Notebook Papers/Labs/Workshop).

Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., Yamada, A., Jun. 2001. Color and Texture Descriptors. IEEE Trans. Cir. and Sys. for Video Technol. 11 (6), 703–715.
URL http://dx.doi.org/10.1109/76.927424

Manning, C. D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.

Manoj, M., Jacob, E., 2008. Information retrieval on Internet using meta-search engines: A review. Journal of Scientific & Industrial Research 67 (10), 739–746.

Manoj, M., Jacob, E., 2013. Article: Design and Development of a Programmable Meta Search Engine. International Journal of Computer Applications 74 (5), 6–11, full text available.

Manral, J., Hossain, M. A., 2015. An Innovative Approach for online Meta Search Engine Optimization. CoRR abs/1509.08396.
URL http://arxiv.org/abs/1509.08396

Marchand-Maillet, S., Morrison, D., Szekely, E., Kludas, J., Vonwyl, M., Bruno, E., 2011. Mining Networked Media Collections. In: Detyniecki, M., García-Serrano, A., Nürnberger, A. (Eds.), Adaptive Multimedia Retrieval. Understanding Media and Adapting to the User. Vol. 6535 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 1–11.
URL http://dx.doi.org/10.1007/978-3-642-18449-9_1

Martínez, Á., Lana Serrano, S., Martínez-Fernández, J. L., Martínez, P., 2012. Multimodal Queries to Access Multimedia Information Sources: First Steps. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 35–40.
URL http://dx.doi.org/10.1007/978-3-642-35145-7_5

Martínez, P., Martínez, J. L., Segura-Bedmar, I., Moreno-Schneider, J., Luna, A., Revert, R., 2016. Turning User Generated Health-related Content into Actionable Knowledge Through Text Analytics Services. Comput. Ind. 78 (C), 43–56.
URL http://dx.doi.org/10.1016/j.compind.2015.10.006

Medina-Ramírez, R. C., 2007. Semantic Information Retrieval: a return on experience. Engineering Letters 15 (2), 234 – 239.
URL http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=27952768&lang=es&site=ehost-live

Mourao, A., Magalhaes, F. M. J., 2013. NovaSearch at TREC 2013 Federated Web Search Track: Experiments with rank fusion. In: Proceedings of The Twenty-Second Text REtrieval Conference, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013.

Pal, D., Mitra, M., 2013. ISI at the TREC 2013 Federated task. In: Proceedings of The Twenty-Second Text REtrieval Confer-

ence, TREC 2013, Gaithersburg, Maryland, USA, November 19-22, 2013.

Pérez-Iglesias, J., Pérez-Agüera, J. R., Fresno, V., Feinstein, Y. Z., 2009. Integrating the Probabilistic Models BM25/BM25F into Lucene. CoRR abs/0911.5046.
URL http://arxiv.org/abs/0911.5046

Porter, M. F., 2001. Snowball: A language for stemming algorithms. Published online, accessed 11.03.2008, 15.00h.
URL http://snowball.tartarus.org/texts/introduction.html

Project, A. C. R., Cleverdon, C., 1962. Report on the Testing and Analysis of an Investigation Into Comparative Efficiency of Indexing Systems. College of Aeronautics.
URL http://books.google.es/books?id=vr8YAAAAMAAJ

Prud'hommeaux, E., Seaborne, A., 2008. SPARQL Query Language for RDF. W3C Recommendation.
URL http://www.w3.org/TR/rdf-sparql-query/

Rekha, C., Usharani, J., Iyakutti, K., 2011. Improving the Information Retrieval System through Effective Evaluation of Web Page in Client Side Analysis. International Journal of Computer Applications 15 (6), 35–39, published by Foundation of Computer Science.

Renaud, G., Azzopardi, L., 2012. SCAMP: a tool for conducting interactive information retrieval experiments. In: Proceedings of the 4th Information Interaction in Context Symposium. IIIX '12. ACM, New York, NY, USA, pp. 286–289.
URL http://doi.acm.org/10.1145/2362724.2362776

Romberg, S., Lienhart, R., Hörster, E., 2012. Multimodal Image Retrieval - Fusing modalities with multilayer multimodal pLSA. IJMIR 1 (1), 31–44.

Salton, G., Buckley, C., 1997. Readings in Information Retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Ch. Improving Retrieval Performance by Relevance Feedback, pp. 355–364.
URL http://dl.acm.org/citation.cfm?id=275537.275712

Salton, G., Wong, A., Yang, C. S., 1975. A Vector Space Model for Automatic Indexing. Commun. ACM 18 (11), 613–620.
URL http://doi.acm.org/10.1145/361219.361220

Schneider, J. M., 2015. New Approaches to Interactive Multimedia Content Retrieval from different Sources. Ph.D. thesis, Universidad Carlos III de Madrid.

Schneider, J. M., Fernández, J. L. M., Martínez, P., 2014. A Proof-of-Concept for Orthographic Named Entity Correction in Spanish Voice Queries. In: Nürnberger, A., Stober, S., Larsen, B., Detyniecki, M. (Eds.), Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation. Lecture Notes in Computer Science. Springer International Publishing, pp. 181–190.
URL http://dx.doi.org/10.1007/978-3-319-12093-5_10

Schneider, J. M., Salazar, M. G., Martínez, P., Fernández, J. L. M., 2009. Some Experiments in Evaluating ASR Systems Applied to Multimedia Retrieval. In: Adaptive Multimedia Retrieval. pp. 12–23.

Shah, U., Finin, T., Joshi, A., Cost, R. S., Matfield, J., 2002. Information Retrieval on the Semantic Web. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management. CIKM '02. ACM, New York, NY, USA, pp. 461–468.
URL http://doi.acm.org/10.1145/584792.584868

Silberschatz, A., Galvin, P. B., Gagne, G., 2008. Operating System Concepts, 8th Edition. Wiley Publishing.

Smalheiser, N. R., Lin, C., Jia, L., Jiang, Y., Cohen, A. M., Yu, C., Davis, J. M., Adams, C. E., McDonagh, M. S., Meng, W., 2014. "Design and implementation of Metta, a metasearch engine for biomedical literature retrieval intended for systematic reviewers". Health Information Science and Systems 2 (1), 1–9.
URL http://dx.doi.org/10.1186/2047-2501-2-1

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-Based Image Retrieval at the End of the Early Years. IEEE Trans. Pattern Anal. Mach. Intell. 22 (12), 1349–1380.
URL http://dx.doi.org/10.1109/34.895972

Smiley, D., Pugh, E., 2009. Solr 1.4 Enterprise Search Server. Packt Publishing.

Smith, R., 2007. An Overview of the Tesseract OCR Engine. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02. ICDAR '07. IEEE Computer Society, Washington, DC, USA, pp. 629–633.
URL http://dl.acm.org/citation.cfm?id=1304596.1304846

Strang, G., 2006. Linear Algebra and Its Applications. Thomson Brooks/Cole.
URL https://books.google.co.uk/books?id=q9CaAAAACAAJ

Sushmita, S., 2012. Study of result presentation and interaction for aggregated search. SIGIR Forum 46 (1), 86–87.
URL http://doi.acm.org/10.1145/2215676.2215692

Torres, J. M., 2005. Visual Information Retrieval through Interactive Multimedia Queries. Ph.D. thesis, Lancaster University.

Vallet, D., Cantador, I., Jose, J., 2012. Exploiting semantics on external resources to gather visual examples for video retrieval. International Journal of Multimedia Information Retrieval, 1–14.
URL http://dx.doi.org/10.1007/s13735-012-0017-1

Womser-Hacker, C., 1996. Das MIMOR-Modell: Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval.
URL http://books.google.es/books?id=KHr4HAAACAAJ

Wong, K.-M., Cheung, K.-W., Po, L.-M., 2005. MIRROR: an interactive content based image retrieval system. In: Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on. pp. 1541 – 1544 Vol. 2.

Worring, M., Snoek, C. G. M., de Rooij, O., Nguyen, G., Smeulders, A. W. M., 2007. The Mediamill Semantic Video Search Engine. In: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. Vol. 4. pp. IV–1213–IV–1216.

Wu, S., Crestani, F., 2015. A geometric framework for data fusion in information retrieval. Information Systems 50, 20 – 35.
URL http://www.sciencedirect.com/science/article/pii/S0306437915000113

Yang, J., Li, Q., Zhuang, Y., 2002. OCTOPUS: Aggressive Search of Multi-modality Data Using Multifaceted Knowledge Base. In: Proceedings of the 11th International Conference on World Wide Web. WWW '02. ACM, New York, NY, USA, pp. 54–64.
URL http://doi.acm.org/10.1145/511446.511454

Yang, L., Cai, Y., Hanjalic, A., Hua, X.-S., Li, S., 2012. Searching for images by video. International Journal of Multimedia Information Retrieval, 1–13.
URL http://dx.doi.org/10.1007/s13735-012-0023-3

Yilmaz, T., Gulen, E., Yazici, A., Kitsuregawa, M., 2012. A RELIEF-based modality weighting approach for multimodal information retrieval. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval. ICMR '12. ACM, New York, NY, USA, pp. 54:1–54:8.
URL http://doi.acm.org/10.1145/2324796.2324858