# Aggregated search: a new information retrieval paradigm

Arlind Kopliku[1], Karen Pinel-Sauvagnat[1], Mohand Boughanem[1]

Traditional search engines return ranked lists of search results. It is up to the user to scroll this list, scan within different documents and assemble information that fulfill his/her information need. *Aggregated search* represents a new class of approaches where the information is not only to be retrieved but also assembled. This is the current evolution in Web search, where diverse content (images, videos, ...) and relational content (similar entities, features) are included in search results.

In this survey, we propose a simple analysis framework for aggregated search and an overview of existing work. We start with related work in related domains such as federated search, natural language generation and question answering. Then we focus on more recent trends namely *cross vertical aggregated search* and *relational aggregated search* which are already present in current Web search.

Categories and Subject Descriptors: H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms: Theory, Measurement, Algorithms

Additional Key Words and Phrases: information retrieval, aggregated search, focused retrieval, result aggregation, vertical search, information extraction, relational search

## 1. INTRODUCTION

### 1.1 Context and definitions

In response to a query, traditional information retrieval systems return a list of ranked documents that match the query. In general, the user looking for relevant material browses and examines the returned documents to find those that are likely to fulfill his need. If lucky, the user will find in this list the document that satisfies completely his/her need. However, when one document alone is not enough i.e. the relevant information is scattered in different documents, the user has to collect and aggregate information coming from different documents to build the most appropriate response to his/her need. Combining these different information to achieve at the same time better focus and better organization within the search results is the scope of *aggregated search*, which is defined to go beyond the uniform ranking of snippets. *Its goal is to assemble useful information from one or multiple sources through one interface* [Murdock and Lalmas 2008, Kopliku 2011].

If we think of a perfect search engine, we will soon realize that it cannot answer all queries in the same way. For some queries, we just need a precise answer (e.g. weight of Nokia e72). For some others we need a list of features (e.g. features of Nokia e72). For the query "Nokia e72", the useful content can include images, videos, a description, features, the Nokia homepage, dedicated pages to the Nokia

[1]IRIT, University Paul Sabatier, 118 Rte de Narbonne, 31062, Toulouse, France; email:{kopliku, sauvagnat, bougha}@irit.fr

e72 phone, etc. Another interesting query can be the name of a person, for example the US president Barack Obama. It would be interesting to see in response to the query a short bio of the president, associated news, as well as associated entities with the name of the concerned association (president of the United States, married with Michelle Obama, adversary of Mitt Romney, visiting this month Great Britain, ...). Answering perfectly every query is impossible because the user will not explain his/her need in an unambiguous and detailed fashion and often the user does not even know what he/she is exactly looking for. However, **for many queries it is possible to do better than a simple uniform list of document snippets**. Major search engines are moving on from this approach towards aggregated search. They pay more attention to search result snippets, they include images, videos, news, ... and in recent months they also group on a separate frame related information. For instance, the query "Barak Obama" is provided by Google[2] a frame that groups a description, images and a very short bio (see Figure 1). The query is answered with traditional Web search results, vertical search results and relational content provided by the Google Knowledge Graph [Singhal 2012]. Upon query, search engines should be able to integrate diverse content, provide precise answers, put together related content, etc. In this perspective, aggregated search fits naturally in an effort to group all IR approaches that aim more than ranking. It pushes research effort towards result aggregation and query interpretation, as well as it helps understanding current IR evolution.

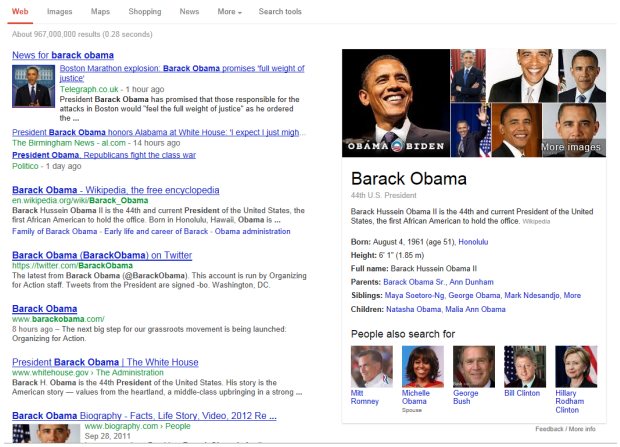


Fig. 1. Google search results on the query "Barack Obama", accessed on April 2013

Aggregated search is not purely a new research area in it derives from many previous work from many IR sub-fields such as federated search, meta-search, natural language generation, question answering, semantic search, entity-oriented search .... At first sight, one might find it hard to see all these domains fit in aggregated search. We will later show how they all can be integrated in one framework

[2]April 2013

that enables having more relevant content, more focus and better result aggregation. There are however some research directions that directly come within the aggregated search *paradigm* and that concern the major recent trends in IR. We will highlight here two research directions that group many existing and current research namely: *cross-vertical aggregated search (cvAS)* and *relational aggregated search (RAS)*.

—**Cross-vertical aggregated search (cvAS)** is almost omnipresent in Web search nowadays. It consists of including diverse content such as (images, videos, news ...) within traditional search results through the use of vertical search engines. *Vertical search engines are specialized search engines which work well for a specific kind of content (maps, video, image, news, ...)*. Research in this direction includes studies on the interest of the inclusion of vertical content, the proposal of different approaches on reassembling or re-ranking search results as well as proposal on evaluation. Federated search is the ancestor of this field and has inspired most of the approaches. However, we find here many unanswered research questions given the specificity of Web search and vertical search engines.

—**Relational aggregated search (RAS)**:
Relational information can often be useful to answer queries right away and/or to structure search results. For instance, the relation triple ("Paris", *capital of*, "France") can be used to answer the query "capital of France". On the other hand, the results for the query "Oscar Wilde" can contain a list of his books shown with a title and description each. This is a simple example where relations such as ("X", *written by*, "Y") and ("A", *description of*, "B") come into play. Relations can be the implicit output of a vertical search (e.g. ("I", *image of*, "Y") can be seen as the output of image search), but most of relations are to be extracted explicitly from the Web (Information Extraction) or to be identified within existing relational sources (knowledge graphs, databases). Forms of relational aggregated search are already met in major search engines, where features, descriptions and related entities are being shown aside search results (see Figure 1). RAS is thus complementary to cvAS where no explicit result aggregation is done and where search results remain a list of unrelated items from different sources with some grouping (by content type).

In its broadest definition, we say that aggregated search aims at retrieving and assembling *information nuggets* where an *information nugget* can be of whatever format (text, image, video, ...) and granularity (document, passage, word, ...). The key problem which makes the aggregated search complex comes mainly from the fact that the assembly of information is topic-dependent i.e. aggregated results cannot be built a priori. Indeed, one cannot predict and construct all possible combinations that can potentially answer user queries. The composition of information nuggets that meet the query constraints is made upon the query. This leads to several research issues including the identification of candidate information nuggets for aggregation, the ways to assemble content and a theoretical framework that may support the evaluation of the quality of an aggregated result.

The contribution of this paper can be summarized in some main points:

—We group in one class (aggregated retrieval class) multiple approaches which do

not fit well in the boolean or ranked retrieval paradigm

—We propose a general framework for aggregated search that can be used to analyze and classify the different existing approaches

—We propose an overview of the broad research domains that are related to aggregated search

—We propose a detailed survey on cross-vertical aggregated search and relational aggregated search

—We list different challenging and promising research directions

The rest of the paper is structured as follows. In the next section, we introduce the motivation behind aggregated search. Then, in section 3 we present a general framework for aggregated search decomposed in three main components. In section 4, we provide an overview of existing approaches organized in broad classes such as federated search, natural language generation, etc. From the existing approaches, we select cross-vertical aggregated search and relational aggregated search as most promising and novel. Approaches of these two classes are analyzed in detail in two dedicated sections respectively section 5 and section 6. We then discussed in section 7 how evaluation of aggregated search is carried on in the literature. The last section is about conclusions and future research directions.

## 2. MOTIVATION FOR AGGREGATED SEARCH

Most of current Information Retrieval (IR) systems fall into the ranked retrieval paradigm i.e. in response to a query they return a list of ranked documents that match the query. Typically, results are ranked by scoring functions which combine different features generated from the query and the documents. This matching process is driven by different theoretical models such as the vector space model [Salton et al. 1975], the probabilistic model [Robertson and Walker 1994], language models [Ponte and Croft 1998] or fuzzy models [Baziz et al. 2007]. However, in all these models *the relevance score is computed at the document level i.e. we assume that the entire document is relevant or not.* Different arguments exist with respect to the definition of relevance at document level and its impact on system performance [Boyce 1982, Spärck-Jones et al. 2007, Murdock and Lalmas 2008]. In this section we will list some of the limitations and challenges of the ranked document retrieval paradigm:

—**Data sparseness:** The relevant information can be scattered in different documents [Murdock and Lalmas 2008]. The ranked list for these cases is inadequate, because the user has to scan within different documents to satisfy his information need. This can be a time-consuming and burdensome process.

—**Lack of focus:** Ranked retrieval approaches provide a ranked list of uniformly presented results. Typically in Web search, each result is a snippet composed of the result title, a link to the document and a summary of the linked document. Instead of directing the user to the whole document, for queries when the answer is just a part of document, it might be better to return this part of document right away. The uniform snippets usually returned by search engines might not always be appropriate.

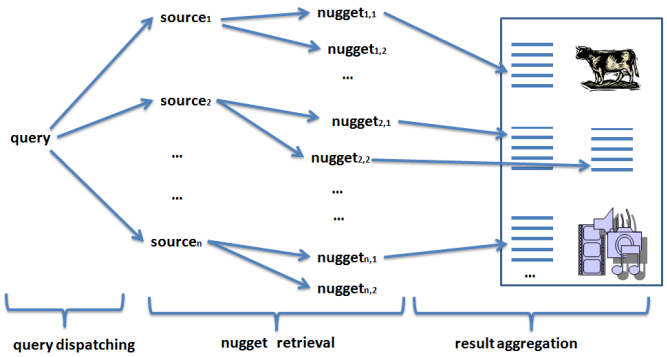


Fig. 2. The general framework of the aggregated search process

—**Ambiguity:** Many queries can be ambiguous in terms of information need. The reference example is *Jaguar* which can refer to a car, an animal, an operating system and so on. Ideally, we should return one answer per query interpretation [Spärck-Jones et al. 2007]. This can be multiple ranked lists or linked sets of results.

Data sparseness, lack of focus and ambiguity are just some of the limitations of traditional ranked retrieval. In *aggregated search (aggregated retrieval)*, the final result is not necessarily a ranked list of documents or snippets, but it can be whatever combination of content that can turn useful to the user. Aggregated search can thus propose more focus, more organization, and more diversity in search results.

## 3. A GENERAL FRAMEWORK FOR AGGREGATED SEARCH

The first definiton of aggregated search is met in a dedicated workshop in SIGIR 2008 [Murdock and Lalmas 2008]:

**Definition:** *Aggregated search is the task of searching and assembling information from a variety of sources, placing it into a single interface.*

In a broad interpretation, aggregated search concerns at the same time the retrieval and the assembly of search results. Here, we propose a unified framework for aggregated search that facilitates the analysis of the many and diverse approaches related to aggregated search. The framework, shown in figure 2, involves three main components namely *query dispatching (QD)*, *nuggets retrieval* (ND) and *result aggregation (RA)*. In other terms, a query is processed and sent to potentially multiple sources of information. Each source returns information nuggets of some granularity and media type, which are to be assembled in the final answer. The answer can be whatever sensed organization of information (though not just the ranked list). This framework has the following multiple purpose: it enables a top-down view of aggregated search; it generalizes well most existing approaches and it identifies the main components of aggregated search; it helps to classify and analyze existing approaches; and it makes a clear distinction of the result aggregation process.

Two terms are important to be well-defined at this point for the understanding of

this framework. *(i)* An *information nugget*[3] is a generalization of content of some granularity and multimedia format. *(ii)* A *source* refers to a search engine that relies on at least one collection and one search algorithm. A good definition of the term "source" is important here, because we use it to classify aggregated search systems as *mono-source* or *multi-source*. When there are multiple sources, it is up to the *query dispatching* step to select the sources to be used.

Each of the three components of this framework will be detailed further in the next sub-sections.

### 3.1 Query dispatching

We include in the query dispatching step the actions that precede query matching, i.e. initial interpretation of the query (query analysis) and other actions that depend mainly on the query and knowledge about the collections. We can also see this step as deciding which solutions should be triggered for a given query. We distinguish between the approaches that aim at selecting the right sources to be used, the approaches that try to understand more about the query, and the approaches that try to extend the query to have more chances to find good search results. We will list here briefly well-known problems in this context:

**Source selection:** Source selection is one of the most well-known problems in aggregated search with multiple sources. Given a set of sources, its goal is to select the sources that are likely to answer the query. Identification of key terms can for instance be used to select sources. The presence of special keywords is often useful to understand the type of answer the user is expecting. For example the presence of words such as "weather" or "definition" is useful clue that is used by major search engines to trigger a personalized answer. Facebook graph search recognizes terms such as "likes", "friends", "photos". Named entities have also been shown to be very frequent within search queries [Guo et al. 2009]. Identifying the named entities and their types can assist the IR process. For instance, if the query is "Oscar Wilde", we can query a knowledge base to provide the user a description and an extract of important features on the writer. As well, we can propose similar authors or authors of the same period. We will discuss this task in more details when we will discuss multi-source aggregated search approaches and in particular cross-vertical aggregated search.

**Question intent:** In question answering, queries are analyzed to find what type of question is being asked (why, where, when, yes/no questions ...) and to identify syntactic or semantic relations that might exist between the answer and entities or events mentioned in the question [Hirschman and Gaizauskas 2001]. Queries can also be decomposed to ease the retrieval process [Moriceau and Tannier 2010, Lin and Liu 2008, Katz et al. 2005]. Understanding the query intent can also be of interest for some sources: one can for example cite some approaches that try to correctly interpret queries with little text on knowledge bases [Pound et al. 2012, Tran et al. 2009].

[3]The term *"information nugget"* has been used frequently in research to denote semantic pieces of information [Clarke et al. 2008, Goecks 2002] and in particular in question answering [Kelly and Lin 2007, Voorhees 2003], although without any common agreement on its meaning. Here, we give a general definition which suits also the context of aggregated search.

**Query reformulation** Depending on the sources that will be queried, some adaption/personalization of the query may be needed. For instance in a data integration perspective, the user only sees the general schema (i.e. the unified view of the different sources of data). A reformulation step is needed: the query on the global schema has to be be reformulated in a set of queries on the different sources [Lenzerini 2002].

### 3.2 Nugget retrieval

Nugget retrieval, as presented here, is situated between query dispatching and result aggregation i.e. it corresponds basically to the definition of a source which takes as input a query and matches it with a (scored) set of potentially relevant information nuggets. In IR, there are many types of nugget retrieval mostly depending on the type of nugget being returned. We enumerate here some of the most well-known approaches.

As we mentioned, it is possible to retrieve entire documents or parts of documents. This corresponds to the distinction between *document retrieval* and *passage retrieval* (also called *focused retrieval* in semi-structured collections) [Geva et al. 2009, Kamps et al. 2008, Trotman et al. 2010, Sauvagnat et al. 2006]. The retrieval process also depends on the multimedia type (text, image, video, ...). We can distinguish here *textual retrieval* and *multimedia retrieval*. Some search engines such as Web search engines retrieve content of heterogeneous type (Web pages, images, news articles, videos, ...). When the one-size-fits-all solution does not work well, it is common to derive *vertical search* solutions which are specialized on a media type, query type, task, result type, etc. We should also mention the approaches that retrieve from relational databases, knowledge graphs or extracted information. They deal with relational information i.e. information with some explicit semantic. The results can be tables, table lines, semantic triples, etc.

In aggregated search systems with multiple sources, each source triggers its own nugget retrieval process. The retrieved results will not necessarily have comparable scores. Assembling search results from multiple sources has been an intensive area of research for at least two decades. It starts with *federated search* (Distributed Information Retrieval) [Callan 2000] in the context of distributed data collections (hidden Web, databases, etc). Then, it has evolved into areas such as *meta-search* [Selberg and Etzioni 1995, Srinivas et al. 2011] and more recently into *cross-vertical aggregated search* [Arguello et al. 2009, Lalmas 2011]. The latter represents today the most succesful class of approaches and it is applied by almost all major Web search engines. To distinguish between these approaches it is important to have a clear definition of the term "source". The terminology in literature is a little messy as terms such as search engine, resource, collection are often used as synonyms. We recall that we prefer using the term source to refer to a search engine or a component of a search engine that uses at least one collection and one search algorithm. This enables us to classify multi-source retrieval approaches such as meta search, federated search, mashups and cross-vertical aggregated search. We will describe multi-source approaches more in detail in section 4.

To conclude, we can say that the nugget retrieval output depends on the type of sources and the number of sources being used. This affects the focus and the diversity within retrieved information nuggets.

### 3.3 Result aggregation

Result aggregation starts from the moment we have a set of potential relevant information nuggets. It involves the different ways of putting content together. In this section, we list the most generic actions that can be applied on candidate search results before showing them to the user. The goal is to show that we can do more than ranking (one possible action). More details on specific result aggregation approaches will be provided later. We could identify 5 basic aggregation actions: *sorting*, *grouping*, *merging*, *splitting* and *extracting*. These actions can be met alone or combined (for instance sorting can be applied on nuggets or group of nuggets). We will detail each of them below:

—**Sorting:** Given a set of information nuggets $n_1, n_2 \ldots n_m$ the *sorting* action produces another list of nuggets $n_{l_1}, n_{l_2} \ldots n_{l_m}$ where all elements are ranked with respect to some features. These features can be relevance scores, but also time, author, location, popularity scores, and so on. Although we mention that the goal of aggregated search is to go beyond the ranked list approach, ranking (sorting) by relevance or by other features remains fundamental for most IR systems.

—**Grouping:** Given a set of information nuggets $n_1, n_2 \ldots_m$ the *grouping* action produces groups (sets) of nuggets $G_1, G_2 \ldots G_i$ where elements of the same group share some property. A group of results can be composed of results with similar content, results that have happened in the same period in time, results with a common feature. Among grouping approaches we can mention clustering [Hearst and Pedersen 1996] and classification [Manning et al. 2008] as special and illustrative cases.

—**Merging:** We refer to *merging* as an action that takes a set of information nuggets $n_1, n_2 \ldots n_m$ and produces a new cohesive aggregation unit. This is different from grouping, because it produces one final grouped unit and not multiple groups. The output of this action can be a new document, a book, a summary, an object. For instance, multi-document summarization [Dang 2006] takes multiple documents and it produces one new aggregate unit, the summary. In object-level search [Nie et al. 2007, Nie et al. 2007], information is merged around named entities (e.g. Nokia e72, France, Paris, ...) into so-called *objects*.

—**Splitting:** Given some content, we can do the opposite action of merging. We can decompose the content in other smaller nuggets. The result of this decomposition can be one smaller nugget or a set of smaller nuggets. Given some content $n$, the result of splitting is a set of information nuggets $n_1, n_2 \ldots n_m$ with $m \geq 1$. Typically, splitting can produce a total partition over the initial content $n$ such that $n_1 \cup n_2 \cup \ldots \cup n_m = n$. Splitting can be as simple as decomposing pure text into passages or sentences, but it quickly becomes a complex task when markup language is present (e.g. HTML pages).

—**Extracting:** Extracting is more about identifying one or more semantically sound information nuggets within some content. This can be part-of-speech items (e.g. nouns, verbs, pronouns, ...), named entities (e.g. Italy, Rome, Roberto Carlos, Hotel Bellagio, ...), images, videos, etc. The target of this action is not a total partition of the initial content rather than specific type of stand-alone

content (i.e. content that can exist alone). The last two actions are not simply useful to decompose some content, but they can precede some sensed re-use of the content.

Whatever the generic actions applied on candidate nuggets, the resulting aggregate should be coherent, with complementary nuggets, and without redundancy. Finding multiple times the same nugget can however help to determine its importance. Probabilistic sampling can for instance be used as a way to select appropriate nuggets [Downey et al. 2005].

In the next section, we list different forms of aggregated search listed around research domains.

## 4. ANALYSIS OF AGGREGATED SEARCH APPROACHES

In this section, we overview work from different IR domains which relate directly to the definition of aggregated search or may be inspiring to its future development. In Natural Language Generation we find interesting approaches that focus on result aggregation (generation of documents from scratch). In Question Answering, there are inspiring case studies with respect to query interpretation and result aggregation. On the other hand, we will see how work from federated search evolves naturally to a specific and promising research direction we call cross-vertical aggregated search. In the quest of aggregation, we group together all approaches that retrieve relations in one broad class named relational aggregated search. The latter is claimed complementary to cvAS because relations can indeed bring to real assembly of information. We will not forget to mention here domain-specific case studies. All of the approaches will be analyzed in the light of our framework.

### 4.1 Natural Language Generation

The goal of natural language generation (NLG) is similar to the broad goal of aggregated search i.e. it aims to avoid the user the burden to browse several documents through the automatic generation of answers in an appropriate linguistic form [Paris et al. 2010]. Let us analyze NLG in the light of our framework. NLG cares less about query dispatching. Indeed, the query is not always present i.e. some NLG approaches start from an information need input as a query and others are designed starting from a known context of use (information need is implicit). In [Grice 1975, Paris et al. 2010], the final goal is the generated document and this goal is to be decomposed in subgoals. The query can also be manually decomposed in sub-queries (beforehand) [Paris et al. 2010]. Nugget retrieval is not the major goal either. In some approaches, the content to be assembled is known beforehand or filtered in a pipeline [Paris and Colineau 2006]. In others, we rely on existing search engines or information coming from databases or XML collections [Paris et al. 2001, Paris et al. 2005]. *The assembly of content in a coherent linguistic fashion is indeed the main goal of NLG.* The retrieved content is aggregated using prototypical ways of organizing information known also as *discourse strategies* (result aggregation). This prototypical structures can be observed (e.g. from linguists) or learned. Returned facts can be organized based on their relationships. For instance, they can be ordered chronologically, they may have a cause-effect relation, background information is shown first and so on [Paris 1988]. Discourse strategies

can be static (i.e. known before hand) or dynamic. The dynamic strategies depend on the information need, but also on the availability of information.

To illustrate, we will provide some examples from literature across different application domains [Paris and Colineau 2006, Paris et al. 2001, Paris et al. 2005]. In [Paris et al. 2005], NLG is applied on a surveillance context. Instead of having monitors poll from different sources, authors propose an NLG approach. Here, data comes from various public-domain information sources (databases, reports, ...) and it is assembled as a multi-page website for port surveillance operators. There is no explicit query, but the abundant information that comes from different sources in different interfaces is put together into one aggregate answer. The second application [Paris et al. 2001] concerns traveling. The user can input its travel destination and optionally budget constraints. Using samples of travel guides, a discourse strategy is learned and then applied to generate coherent travel guides from scratch upon query. In [Paris and Colineau 2006], the NLG approaches are used to produce brochures that summaries on given organizations. Another interesting example is met in [Sauper and Barzilay 2009]. Authors propose an approach to automatically generate Wikipedia-like documents through machine learning on Wikipedia pages of the same class (e.g. diseases, actors). The learned document structure (e.g. causes, treatment, symptoms, ...) is then used to automatically build new documents for other instances of the class extracting missing passages from the Web.

NLG is an interesting case study in particular for result aggregation. Its limits relate to query dispatching and nugget retrieval, which might be the reason NLG approaches are more successful in domain-specific applications rather than in uses through free natural language querying. We do not focus more on this class of approaches, but we recommend the interested reader a qualitative survey in [Paris et al. 2010].

### 4.2 Question Answering

Question answering (QA) differs from traditional ranked retrieval as it does not aim a list of documents, but one or multiple answers. It represents an interesting case study for aggregated search because these answers might not exist, they have to be produced, extracted, assembled [Dang et al. 2007].

In QA, we also find the three main components of our AS framework. Concerning query dispatching, queries in QA are not any free text rather than question-like queries. For questions there exist different taxonomies and it is in the interest of the QA system to understand the query type. We can list here some well-known question types such as the "Who", "What", "Where", "When" questions or "Yes/No" questions. As well, existing approaches look for named entities and other helpful facts within the query e.g. "Where in Africa did the scientists sent by Napoleon find Rosetta Stone?" This question contains different named entities (Napoleon, Africa ...) and facts (in Africa, scientists). Interesting case studies and observations of the relation between aggregation search and QA can be found more detailed in [Moriceau and Tannier 2010].

Nugget retrieval is not the major target of QA i.e. experimenting the different query matching algorithms is not the primary goal. Given a set of documents that match the query or queries derived from the initial question, QA approaches go

within the documents to find and assemble answers. This task is not easy. Often, wrong passages are selected and the result is useless.

From the perspective of result aggregation, there are multiple ways to put results together and the result is often dependent on the query. As providing one unique answer may be risky, QA systems often return a list of candidate answers. Answers are often juxtaposed with supporting texts extracted from the documents. A different approach is presented in [Wu and Fuller 1997]. Instead of returning a list of documents to answer questions and instead of very focused answers, authors propose different intermediary approaches which can assist the answering process. This corresponds to 5 different result aggregations: *(i)* an abstract and a document; *(ii)* sentence fragments with keywords highlighted and a document; *(iii)* an abstract with one document and other related documents; *(iv)* sentence fragments and multiple documents; *(v)* a set of paragraphs.

### 4.3 Federated search

Most of the work in IR with multi-sources is classified as *federated search* [Avrahami et al. 2006, Aditya and Jaya 2008, Callan 2000] also known as *distributed information retrieval*. In federated search, instead of having one central collection indexed by one search engine, there are many distributed text collections each indexed by a search engine [Arguello et al. 2012][4]. We may have as many query match algorithms as the number of sub-collections (see figure 3 on the left). At query time, the federated search system has to select the sources which are likely to be useful for the query. To do so, local representations of significantly reduced size of each collection are used. The obtained results from different sources are then assembled with each other. Typically, the final answer is a ranked list.

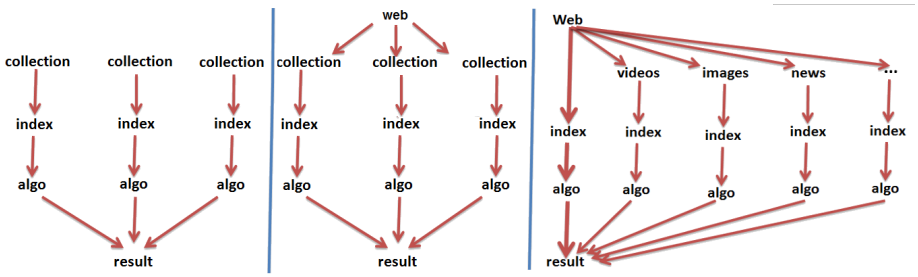


Fig. 3. Simple schemas for federated search (left), meta-search (center), and cross-vertical aggregated search (right)

The FedLemur project [Avrahami et al. 2006] is an illustrate example of federal search engine that is built on top of statistical data issued from 100 US federal agencies. Instead of building a centralized collection which can quickly get outdated,

[4]Federated search should not be confused with data fusion or data integration. Data fusion aims at finding optimal combinations of query matching algorithms running on the same index/collection, while in data integration the problem is to combine data from different sources, and provide the user a unified view of these data (i.e. a global schema) [Lenzerini 2002].

authors show that it is better to build local search engines in many distributed nodes (one per agency).

Federated search approaches with multiple distributed collections have not been shown to produce very significant improvement with respect to approaches with one unique index. Nevertheless, they are commonly used when the sources are too heterogeneous or when some collection is hidden behind some API. In all approaches with multiple sources, we can easily identify the query dispatching step, nugget retrieval step and result aggregation step.

*Meta-search* can be seen as a form of federated search in the context of Web search [Lalmas 2011]. Initial meta-search engines used to wrap different Web search engines with the goal of improving precision and recall of Web search [Selberg and Etzioni 1995]. This was reasonable at the time because the indexes of the existing Web search engines covered small fractions of the Web. Though, the chances to have other relevant results in another source were high.

The sources in meta-search are typically black-box search engines which receive queries and return a ranked list of results. The returned results from different sources are then combined into one interface [Selberg and Etzioni 1995, Manoj and Jacob 2008, Srinivas et al. 2011]. Typically results are sorted by source, but they can also be ranked with each other in one list. In contrast with most federated search work, sources can target the same task and collection (see figure 3 on the center).

### 4.4 Mashups

Mashups are an interesting case-study from the perspective of aggregated search. A mashup is a web application that integrates data as well as services (databases, search engines, web services, ...) into a single tool. The mashup allows canalizing the information flow and the way information is combined. The aim is to provide a new service to user.

Manually coding mash-upsv we can build documents on demand, where we predefine where to put content coming from each component of a mash-up. Although they are an interesting case-study, mash-ups are mostly used for expert tasks rather than for search [Greenshpan et al. 2009, Ranganathan et al. 2009].

### 4.5 Cross-vertical aggregated search

*Cross-vertical aggregated search (cvAS)* [Arguello et al. 2009, Lalmas 2011, Kopliku 2011] is the task of searching and assembling information from vertical search and Web search (see figure 3 on the right). This is usually done in a Web search context. A vertical is a specialized sub-collection [Diaz 2009a, Zhou et al. 2012], such as images, video, news, etc. *cvAS* is met in literature as an instance of both federated search[Aditya and Jaya 2008] and meta-search [Gulli and Signorini 2005]. In cvAS however, verticals are a core-concept [Arguello et al. 2012]. Since the definition of aggregated search, a lot of recent work [Arguello et al. 2009, Lalmas 2011, Sushmita et al. 2009, Kopliku et al. 2011, Kopliku et al. 2009] is classified within this later direction. We will concentrate on cross-vertical aggregated search in section 5.

### 4.6 Relational aggregated search

We define as *relational aggregated search (RAS)* a broad class of approaches that involve retrieving and aggregating information nuggets using their relations. For example, the query "all about iPhone 4S" would clearly benefit from this approach. We may assemble related attributes (weight, price, ...), related products, related images, etc. So defined, RAS can be complementary to cvAS. If cvAS provides diversity within search results, RAS can find the sense of aggregation and provide more related content within search results.

Let relational information be whatever triple that puts into relation two information nuggets. A relation can be deduced as simply as by performing a vertical search (image search, video search, ...). However, a huge amount of relations exist in the Web and it can be extracted (with for instance Information Extraction techniques). As well, huge amounts of relational information already exist within knowledge graphs or databases. Being able to extract and use these relations in the context of IR is the goal of RAS.

So defined, RAS has to rely on advances of many well-known fields including Information Extraction, Entity-oriented search, Object-level search, semantic search, IR from databases and so on. The definition of RAS and its situation in research work might seem vague, but we will get back to RAS in a dedicated section (section 6).

### 4.7 Domain-specific applications

Instances of aggregated search can be found in domain-specific applications. These approaches can sometimes be too specific, but they remain interesting to study, because some of the work can be generalized to larger use.

In [Kaptein and Marx 2010], Kaptein et al. investigate focused retrieval and result aggregation on political data. Their documents are long (50-80 pages) containing transcripts of the meetings of the Dutch Parliament. Instead of returning entire documents, authors choose speeches as best entry points into documents. For a given query, they provide a summary of the results as a graph with 3 axes: year, political party and number of search results. Search results can be browsed through three facets: person, political party and year. Each speech is summarized in both terms of structure and content. The latter corresponds to word clouds and interruption graphs. We may observe at least 4 different aggregation formats within this work: interruption graph (structure summary), content summarization, facets, and results graph with 3 axes.

Another application is met in social science [Ou and Khoo 2008]. Here, authors provide search across academic papers. They extract and aggregate research concepts, their relations, research methods and contextual information. The results can then be browsed by method, relation or research concept. For each research concept, the user is provided a summary of contextual information. Strotmann and Zhao [2008] also focus on academic papers. They introduce two graph-based structures to help browsing search results. The first is an aggregated graph on authors. The second is a graph on authors linked through co-citation analysis. Academic research is also considered in sites such as Microsoft Academic Research (`http://academic.research.microsoft.com/`) or Google Scholar citations

(http://scholar.google.com/citations), which aggregate information about a given researcher : publications, conferences, related keywords, co-authors, global indicators such as number of publications, h-index, ....

*News search aggregators* represent good examples of result aggregation that go beyond the ranked list visualization. News articles with similar topic and publication time are often clustered into news stories [Sahoo et al. 2006, Hennig and Wurst 2006]. This can also help users focus their search within a topic or time interval. In addition, in [Rohr and Tjondronegoro 2008] authors propose an interesting timeline visualization showing the evolution of the topic. As well, we can attach to the news stories related multimedia [Rohr and Tjondronegoro 2008]. This is the case for Google News (http://news.google.com/) and Ask News (http://www.ask.com/news).

*Geographic Information Retrieval* is another interesting case-study. It corresponds to search on geographic entities mainly through free text queries [Vaid et al. 2005, Jones and Purves 2009]. Usually a list of candidate results is returned unless the query is specific enough to identify exactly one result. The list of results is shown in a map juxtaposed with the corresponding ranked list. This is the case for major search engines such as Google Maps (http://maps.google.com/). In addition geographic proximity can be used to group and organize search results [Bouidghaghen et al. 2009, McCurley 2001]. If the query is good enough to identify one single geographic entity or the user clicks on one of the proposed results, a geographic search engine can support the user with more information about the entity. For instance, for a hotel it is possible to provide a phone number, reviews, images and so on. In fact, geographic entities can be associated with other types of content such as images [Naaman et al. 2006, Kennedy and Naaman 2008], related named entities [Vallet and Zaragoza 2008], news articles and so on. This is useful to enrich and organize search results.

## 5. CROSS-VERTICAL AGGREGATED SEARCH

Cross-vertical aggregated search has already a consecrated place within aggregated search to the extent that it is often used as a synonym of aggregated search [Lalmas 2011], [Arguello et al. 2009]. The most well-known examples of cross-vertical aggregated search are met in the major Web search engines. In recent years, these search engines do not return uniform lists of Web pages. They also include results of different type such as images, news, videos, definitions, etc. (see Figure 4). A simple definition of cross-vertical aggregated search is met in [Lalmas 2011]:

> [Cross-vertical aggregated search] attempts to achieve diversity by presenting search results from different information sources, so-called verticals (image, video, blog, news,etc.), in addition to the standard Web results, on one result page.

*Motivations* for cross-vertical aggregated search are multiple. We will list here some of the main highlighted advantages already identified in literature:

—First, Web data is highly heterogeneous, and keeping a fresh index of all real-time data is difficult [Arguello et al. 2012]. Specialized search engines that focus on particular types of media are thus required.

—From an user point of view, it has been shown that **vertical search intent** is often present among Web search queries [Arguello et al. 2009, Liu et al. 2009, Kopliku et al. 2011]. For example, queries such as "Avatar trailer" or "Eiffel Tower images" can be found in Web search logs, although these queries might have been issued for videos and images. Liu et al. [2009] analyzed 2153 generic Web queries into verticals, using query logs. They found that 12.3% have an image search intent, 8.5% have a video search intent and so on. In [Arguello et al. 2009], authors classified 25195 unique queries, randomly sampled from search engine logs, into 18 verticals. 26% of the queries (mostly navigational) were assigned no vertical, 44% of the queries were assigned one vertical and the rest of the queries were assigned more than one vertical. The latter were mostly ambiguous. Cross-vertical aggregated search allows the user to search multiple specialized collections with a single-query access.

—At last, it has been shown that *cvAS* increases the **diversity** of relevant results [Sushmita et al. 2009, Kopliku et al. 2011, Santos et al. 2011]. Results from different relevant sources have been shown to be often **complementary** with each other and they have been shown useful in the presence of **ambiguous queries** [Kopliku et al. 2011].

Cross-vertical aggregated search can be seen as a "divide and conquer" approach, where customized search engines are used to answer queries on specific collections and where a complete answer is then built to meet the user need.

*Cross-vertical aggregated search in the aggregated search framework.* In cross-vertical aggregated search it is easy to identify the components of our general framework for aggregated search. Query dispatching will correspond to the selection of sources. Each source will perform its nugget retrieval process and finally retrieved nuggets (Web pages, images, videos, . . . ) will have to be assembled in one interface. Although the tasks are well distributed, the problem is far from being solved. It is not easy to decide which sources should be used and how the retrieved results should be assembled and presented. In particular, we can list some major issues which have the attention of current research:

—Source representation and selection: Which source should be used? How should they be represented internally in terms of features?

—Result aggregation: How should search results from different sources be assembled (ranking by item, ranking by block, . . . )?

—Result presentation: Which are adequate interfaces for cross-vertical aggregated search?

We will describe in more details each of the above issues.

### 5.1 Source representation and selection

Some aggregated search systems make use of all sources they can reach for all queries, while others are *source selective* i.e. they make use only of the sources considered useful for the query. Most of major Web search engines are source selective. This is done to avoid long answering delay querying many sources and waiting for all results can be too slow for standard Web search answering time. *The*

| | Feature based on | Description |
|---|---|---|
| f1 | Vertical intent terms [Arguello et al. 2011; Arguello et al. 2009;Ponnuswami et al. 2011] | Some terms indicate vertical intent such as image, video, photo. This feature combines hard-coded and learned association rules for queries and sources |
| f2 | Named-entity type [Arguello et al. 2011] | These features indicate the presence of named entities of some type in the query. |
| f3 | Query length [Ponnuswami et al. 2011] | This feature corresponds to the length of the query in terms. |
| f4 | Query logs [Arguello et al. 2011; Arguello et al. 2009] | These features indicate if the query has been met in a source query log. |
| f5 | Recent popularity of the query [Diaz 2009b] | This feature indicates how often the query has been met in a source query log recently. |
| f6 | Clickthrough analysis [Arguello et al. 2011; Ponnuswami et al. 2011, Li et al. 2008] | These features are generated from the documents that have been clicked for the query. The click is considered implicit feedback. |
| f7 | Source representation [Diaz and Arguello 2009] | This feature is computed based on explicit and implicit feedback on the query and its intent. |
| f8 | Category representation [Arguello et al. 2009] | This feature is generated through classification of the query into predefined domains such as sport, arts, technology. |
| f9 | Evidence for navigational need [Ponnuswami et al. 2011] | This feature indicates the chances of the query to come from navigational needs. |

Table I. Pre-retrieval features

*goal of source selection is to predict the query intent combining evidence from the query, user profiles, query logs and sources themselves.* This problem is also known as the *vertical selection* problem [Arguello et al. 2011].

To enable efficient source selection, sources have some internal representation in the cvAS systems. This representation can be as simple as a textual description of the source, but in general it contains representative terms and features for the source extracted from sampling or other mining techniques. Some techniques of source selection can be met in federated search [Gravano et al. 1997, Callan 2000, Shokouhi et al. 2007], but we will focus here on the approaches met specifically for cvAS.

Source selection demands the collection of evidence that associates the potential information need behind a query to one source. This evidence is turned into features that can be used with some classification or ranking algorithm. Tables I and II represent an overview of the experimented features in literature. They can be split into pre-retrieval and post-retrieval features, because there exist approaches that select sources after having some search results in return. Within the pre-retrieval features, we account for simple features such as the presence of terms with strong vertical intent evidence such as “image”, “video” (feature f1), the occurrence of named entities of a given type (location, organization, etc) (feature f2), the length of the query (feature f3). Evidence is collected from query logs (f4,f5) in vertical search engines and clickthrough analysis in Web search (feature f6). As well, the query can be matched to feature/term sampled representations of the source collection (feature f7) or of given categories/subjects (f8). As well, it has been shown useful

| | Feature based on | Description |
|---|---|---|
| f10 | Vertical relevance score [Diaz 2009b] | This feature corresponds to the relevance score of the vertical source itself for a result or a block of results. |
| f11 | Uniform Match Score [Arguello et al. 2011; Ponnuswami et al. 2011, Li et al. 2008] | This feature corresponds to match scores on the results computed uniformly across multiple sources. |
| f12 | Number of results [Arguello et al. 2011] | This feature corresponds to the results count of a source. |
| f13 | Freshness of documents [Arguello et al. 2011, Diaz 2009b] | This feature indicates how fresh are search results for the query within a source. |
| f14 | Contextual score of results [Kopliku et al. 2013] | These features indicate the relatedness of search results with respect to some context. |
| f15 | Geographic-context score of results [Kopliku et al. 2013] | These features indicate the relatedness of search results with respect to some geographic context. |

Table II. Post-retrieval features

to apply a classifier that can tell if a query hides a navigational intent (feature f9).

Post-retrieval features include different scores computed on search results for a given query. We find relevance scores as computed by each source (f10) as well as match scores computed on search results in a uniform fashion across all sources (f11). We find also the use of simpler features such as the number of search results (f12) and the freshness of search results (f13). Features related to the location and context of a user/task (f14, f15) can be used to rerank or filter search results i.e. the vertical search engine is personalized for a given contextual use and geographical location.

There are different interesting works with respect to source selection. Diaz studies the integration of news search results within Web search [Diaz 2009a]. He estimates newsworthiness of a query to decide whether to introduce news results on top of the Web search results. They make use of clickthrough feedback to recover from system errors. Li et al. [2008] also rely on clickthrough data to extract implicit feedback for source selection. They represent queries and documents as a bipartite graph and they propagate implicit feedback in this graph. Their approach is experimented for the integration of product search and job search results. Arguello et al. [2009] list various sources of evidence that can be used to tackle source selection such as query-log features, vertical intent terms, corpus features. In later work [Diaz and Arguello 2009], they show how they can integrate implicit and explicit feedback for vertical selection. From a set of 25195 labeled queries, they propagate implicit feedback across about ten million queries. In [Arguello et al. 2010], the same authors show how to adapt source selection methods to new unlabeled verticals.

### 5.2 Result aggregation

There are different ways to assemble results in cross-vertical aggregated search. The most simple approach is to rely only on source selection [Li et al. 2008, Diaz 2009b, Arguello et al. 2009] i.e. search results from one source are integrated only when the source is considered potentially useful. In this case, results are placed in pre-defined locations, usually on top of traditional Web results. We have mentioned

related approaches in the above section.

The task becomes more complex when we want to rank results of different sources with each other. There exist two main approaches namely block ranking and single-result ranking:

In block-ranking, results are ranked in blocks of results of the same source [Ponnuswami et al. 2011, Arguello et al. 2011] e.g. 3 images versus 3 videos, while in single-result ranking results may be interleaved (1 image versus 1 Web page for instance) [Liu et al. 2009].

In [Ponnuswami et al. 2011] and [Arguello et al. 2011], search results are ranked in blocks of vertical content (e.g. 3 images versus 3 Web search results). In both cases ranking functions are trained using pairwise preferences between blocks. Ponnuswami et al. [2011] propose a machine-learning framework which can assist the way search results are ranked in blocks by major search engines. In [Arguello et al. 2011] authors compare different algorithms for block-ranking namely: classification, voting and learning to rank techniques. The latter (learning to rank) outperforms other algorithms.

Ranking results of different sources one by one (not in blocks) has not been largely adapted and studied yet. Although ranking results from very different systems is a difficult problem, interleave results from different vertical search engines may be of interest. Indeed, results in verticals are not equally relevant. A user issuing the query "French military intervention in Mali" may be interested in a result page showing first the last new about Mali intervention and then a Wikipedia article on the Northern Mali conflict, before seeing other less recent news on the subject. Images of the intervention are also more relevant than images of a speech of the French president. In [Liu et al. 2009], authors propose a probabilistic framework for this purpose which also lists different useful features for this task. Current evolution of major Web search engines is showing that we can go beyond block ranking. For instance, for some queries the user is answered with focused information or Wikipedia extracts.

### 5.3 Result presentation

While result aggregation deals with the ranking of results, result presentation concerns the presentation interface. One way to aggregate vertical searches is to place content of the same type into predefined layout panels: this type of approach is called *unblended*. This is the approach chosen by Yahoo! Alpha[5] in its initial times (Figure 4 on the left) and Kosmix[6]. Whenever a minimal amount of vertical content is available for a query, the various predefined slots are filled and displayed around the natural search listings. The advantage of such an approach is that the user knows where the vertical content is shown. On the other hand, vertical content is almost always shown even if there is nothing relevant for such a type of content. Other approaches, called *blended*, merge search results from different sources with each other in the same panel (see Figure 4 on the right). This was first introduced by major search engines who started integrating vertical content only on top or bottom of their search results list (web search results are ranked as usual). When

[5]http://au.alpha.yahoo.com
[6]http://www.kosmix.com

it is probable that the user is looking for other types of content such as images, news, video, this content is placed in the bottom or the top of the Web results list. One of the advantages is that vertical search results are added only when needed and visualization remains simple.

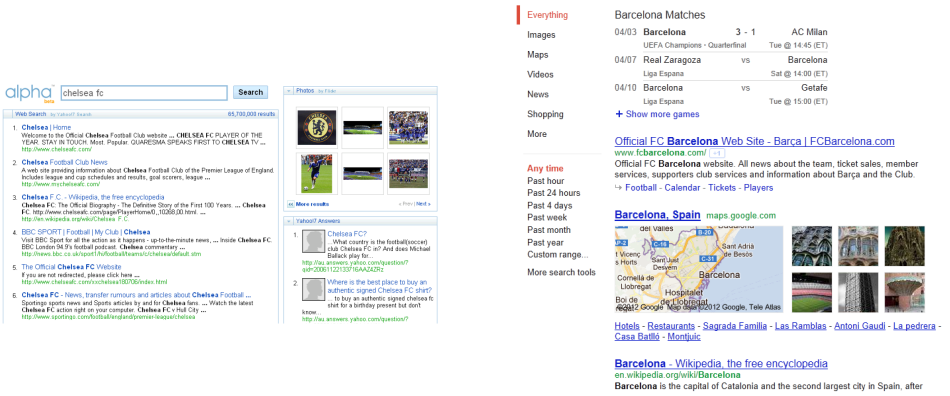


Fig. 4. Yahoo! Alpha search results on the query "Chelsea FC", accessed on April 2009 (on the left) and Google Universal search results on the query "barcelona", accessed on April 2012 (on the right).

Nowadays, major search engines still use a blended approach, but results are ranked by blocks, i.e groups (blocks) of results from each source are ranked with each other. Within the block, results are usually ranked i.e. they are shown in the order they are returned from the origin source (Figure 4 on the right). For instance, we may have a block of three Web search results followed by a block of images, followed by a block of news search results and then Web search results again. The advantage of this approach is that it is simple and at the same time flexible. We can add as many vertical searches as needed and only show those that are more relevant.

In the Web, we can also find other picturesque applications such as Spezify[7] which proposes an original way to exploit visualization space. Spezify aggregates results from different vertical sources with a slight preference for content with immediate visual impact such as images, videos. Each result fills a rectangle and rectangles are placed side to side to fill the visualization space. The results expand in all direction (up, down, left, right) and it looks as a mosaic filled of content. Google has also launched a new application called "what do you love[8]". This application aggregates results from several verticals as well as from other Google applications such as Google Translator, search term popularity measures and so on.

## 6. RELATIONAL AGGREGATED SEARCH

Cross vertical aggregated search does not perform any explicit assembly of search results. Results remain a group of unrelated information nuggets coming from dif-

[7]http://www.spezify.com
[8]http://www.wdyl.com

ferent sources. We define Relational Aggregated Search (RAS) as a complementary set of approaches where relations between information nuggets are taken into account. Relations can be very useful to assembly the search results and can be as simple as ("Paris", *capital of*, "France") or ("x.jpg", *image of*, "Eiffel Tower"). RAS can provide more focus within search results e.g the query "capital of France" can be answered right away with "Paris". It can also provide more structure and organization within search results. e.g. the answer to query "Oscar Wilde" we can include a list of attributes (date of birth, date of death, nationality, ...), a list of related images, a list of related books, similar authors and so on. The user can navigate across the books or the similar authors.

A good example of relational aggregated search is met in Google Squared[9]. Figure 5 shows its results for the query "arctic explorers". Each line corresponds to an arctic explorer. For each of the explorers there are *attributes* such as date of birth, date of death, but also an image and a description. The user can specify his own attributes to search for and he/she can search for a new arctic explorer which is not in the list. To answer this query it is necessary to rely on relations such as ("Roald Amundsen", *is a*, "arctic explorer") or ("Roald Amundsen", *date of birth*, "1872"). If Google Squared retrieves information from the Web, Wolfram Alpha[10] represents a different approach where relational facts are extracted from an internal pre-built collection of facts.

The Google Knowledge Graph introduced in 2012 by Google [Singhal 2012] is another example of the use of relational facts. The Graph is constructed using public sources such as Freebase or Wikipedia, but also using some real-time sources from the Web. It currently contains 500 million objects and 3.5 billion facts, as well as relationships between them. The Google Knowledge Graph is used by Google to disambiguate results, to present a summary of an entity (the right panel in Figure 1) to help the user go deeper in his/her query by chowing him/her new facts or new connections related to the result (explorative search suggestion).

All these applications show however that we can foresee another way of assembling search results based on fine-granularity information and relations.

Relational aggregated search as defined here intersect with many research directions including entity-oriented search, semantic search, object-level search, database information retrieval, ... In the next sections, we will provide main definitions and then we will analyze the related work to show that there is much to rely on in this promising research direction.

### 6.1 Definitions and framework

A relation can be seen as a semantic triple of the form (X, R, Y). Such a relation can be well-defined and explicit such as ("Paris", *capital of*, "France") or less explicit such as ("Paris", *relates to*, "France"). In relational databases, we do also find n-ary relations, but we can observe that most of the useful relations can be mapped into ternary relations which will be the case for this paper. Related items are usually instances[11] (also called named entities, class instances). By definition, an instance is

[9] http://www.google.com/squared/ no longer available online within Google labs
[10] http://www.wolframalpha.com/
[11] The term instance will be preferentially used instead of named entities

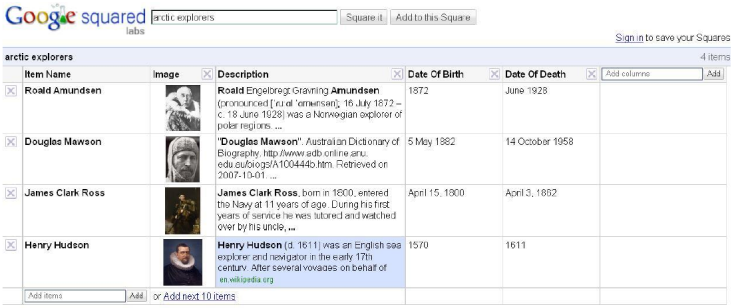


Fig. 5. Google Squared result for "artic explorers" accessed on May 2010

an object/concept belonging to some class (e.g. countries, biologists, cities, movies, . . . ). However, relations can apply to all types of nuggets including longer text e.g. ("The art or science of combining vocal or instrumental sounds (or both) to produce beauty of form, harmony, and expression of emotion.", *definition of*, "music") and multimedia content (`http://en.wikipedia.org/wiki/File:Flag_of_France.svg`", *flag of*, "France").

The RAS paradigm fits in our general aggregated search framework. Its main components are *query dispatching*, *relation retrieval* and *result aggregation*:

—**Query dispatching**: Inspired by the work in [Cafarella et al. 2006, Kopliku et al. 2011b], we distinguish three types of information needs where benefits from relational aggregated search are obvious. We call them relational queries:
  —**attribute query** ("GDP of UK", "address of Hotel Bellagio")
  —**instance query** ("Samsung Galaxy S", "Scotland", "Oscar Wilde")
  —**class query** ("Toshiba notebooks", "British writers")
  Each of these queries may demand a different treatment. From this perspective, query dispatching becomes important to detect the query type and to trigger the appropriate solution.

—**Relation retrieval**: In relational aggregated search, we are interested in all nugget-nugget relations that can be useful for information retrieval. However, we highlight three types of relations which are particularly interesting and useful namely:
  —**instance-class relation:** e.g. ("France",*is a*,"country"); ("Bill Clinton",*is a*,"US president")
  —**instance-instance relation:** e.g. ("Paris",*relates to*,"France"); ("Roger Federer",*plays against*,"Rafael Nadal")
  —**instance-attribute relation:** e.g. ("France",*has attribute*,"(motto:Liberty, Equality, Fraternity)"); ("France",*has attribute*,"population density: $116/km^2$")

—**Result aggregation**: Relations can enable new ways to assemble search results. When the query is specifically asking for an attribute, the best choice can be returning its value right away. When the query is an instance, we may show a summary of salient attributes, but also images, related instances, etc. When the

query is a class of instances, the result can be a comparative table of the class instances with their attributes (as it is done for instance in Google Squared).

We will now detail on the novel issues of relational aggregated search focusing on relational queries, relation retrieval and result aggregation.

### 6.2 Relational queries

To build relational aggregated search, we need to identify the queries that benefit the most from this paradigm. In [Cafarella et al. 2006], authors present one of the first query taxonomies for relational search. For these queries named entities and their relations find a crucial role.

—**qualified-list queries**: retrieve a list of instances that share multiple properties (e.g., west coast liberal arts college).

—**unnamed-item queries**: retrieve single object whose name the user does not know or cannot recall (e.g., the tallest inactive volcano in Africa).

—**relationship queries**: retrieve the relation(s) between two objects (e.g. Microsoft ? Bill Gates).

—**tabular queries**: retrieve a set of objects annotated by their salient properties (e.g., inventions annotated by their inventor and year of announcement).

In the above list, the query type and the type of result are bound i.e. the query definition includes the type of expected result.

In [Kopliku et al. 2011b], we proposed a simpler taxonomy of queries inspired by [Cafarella et al. 2006]. The type of query is binded to the semantic notion of instance, class and attribute, while it is not binded with the type of expected result:

—**attribute query** ("GDP of UK", "address of Hotel Bellagio", ...)

—**instance query** ("Samsung Galaxy S", "Scotland", "Oscar Wilde", ...)

—**class query** ("Toshiba notebooks", "British writers", ...)

Most of the above queries benefit from relations. We will use the last taxonomy, because it is simpler and it does not bind with the type of answer.

### 6.3 Relation retrieval: How to acquire relations?

The major sources of relational content are (i) the Web, (ii) knowledge graphs/ontologies, and (iii) relational databases. We will focus less on relational databases because they are less frequently used in broad IR applications. The Deep web is however a good provider of databases, and challenges such as data integration are a very important research area. In the following, we detail *information extraction (IE)* techniques and *knowledge bases (ontologies, linked data)* as the major sources for relations.

*Information extraction techniques* correspond to rules that can be applied to sites, documents, or parts of documents to automatically extract classes, instances, attributes and their relations. Most of the extraction rules have the form of $LxMyR$, where $x$ and $y$ are meant to be two information extracts and L, M and R are meant to be patterns that are found respectively before, in between and after the two extracts. For instance, the rule "the $x$ of $y$ is" can be used to identify attributes

of instances e.g. "the capital of France is Paris". Rules that rely only on lexicon (words, phrases) and part-of-speech tags are also referred as *lexico-syntactic rules*, while rules that are based on tag sequences are usually known as *wrappers*. Extraction rules can be hard-coded or learned [Chang and Lin 2001].

Information extraction techniques are quite heterogeneous and they make use of various evidence such as statistics on terms, tags, decoration mark-up, part-of-speech tags, etc. This evidence is then combined to define rules that match classes, instances, attributes and their relations. We provide below just a short taxonomy of features (evidence) that are commonly used for this purpose:

—**word statistics** [Etzioni et al. 2005, Agichtein and Gravano 2000, Crescenzi et al. 2001]: Some terms are more frequent within or around information extracts. Statistics on words (often in conjunction with part-of spech tags) are helpful to learn common lexico-syntactic patterns for information extraction.

—**part-of-speech tags** [Bellare et al. 2007]: Information extracts are usually nouns or compound noun phrases surrounded by verbs, adjectives, prepositions, etc. Part-of-speech tags are helpful to learn possible patterns.

—**tags (HTML, XML)** [Crescenzi et al. 2001, Cafarella et al. 2009, Kopliku et al. 2011b, Kopliku et al. 2011b, Kopliku et al. 2011a ]: Most of the documents have some structure denoted through tags. The structure of the document is often useful to determine relations. In particular, HTML tables and HTML lists are known to contain relational data.

—**decoration, visual appearance** [Aumann et al. 2006, Meng et al. 2003, Yoshinaga and Torisawa 2007]: Sometimes the structure of a document is easier to learn through its visual aspects, especially when a pattern in terms of tags is difficult to define or learn.

—**PMI and search hits** [Church and Hanks 1989; Turney 2001, Popescu and Etzioni 2005]: Pointwise Mutual Information (PMI) is a statistical measure that indicates possible correlation between two expressions. An easy way to estimate PMI is through search hits which indicate the number of search results a search engine has to return on a given query . PMI and similar statistical measures can play a crucial role to determine relations.

Most of the techniques in IE are domain-specific i.e. they are designed to work well for some classes or instances or attributes. A noticeable exception concerns Open Information Extraction (OIE) approaches [Banko et al. 2007, Zhu et al. 2009, Etzioni et al. 2011]. Initially introduced by [Banko et al. 2007] with their TextRunner system, the OIE paradigm aims at extracting large sets of domain-independent relational tuples (mainly on Web corpora) without requiring any human input. Open IE systems are now a hot subject of interest, since they can allow to automatically construct konwledge bases or ontologies [Lin and Etzioni 2010].

Most of the IE techniques are also oriented towards precision. To enable relational aggregated search we need high recall and reasonable precision. From this perspective, domain-independent methods and high recall methods become crucial.

A lot of semantic content and their relations are already available in existing *knowledge bases* such as (DBPedia, Freebase, YAGO, ... ) [Suchanek et al. 2007, Wu et al. 2008, Limaye et al. 2010]. These sources contain semantic (relational) informa-

tion manually input or automatically extracted from the Web or encyclopaedia-like collections (e.g. Wikipedia). These sources have also been used to learn IE extraction patterns or to reinforce confidence on extraction [Kopliku 2011]. They have attracted the attention of many recent research and they are already enough to illustrate the potential of RAS.

In the next paragraphs, we list relation-retrieval approaches with respect to the type of relation they target.

6.3.1 *Instance-class relation.* The class instance is often referred in literature as named entity. Initially, there were seven named entity categories defined in Message Undestanding Conference [Grishman and Sundheim 1996], while today standardized taxonomies with more than 150 classes can be found such as the extended named entities hierarchy [Sekine et al. 2002]. In reality, we cannot enumerate all possible named entity classes. A class can be as simple as "countries", but it can also be "members of the UN Security Council" or "my friends". Sometimes, we might not even be able to name the class in a reasonable way.

The definition of named entities as instances of some class makes the class-instance relation intrinsic for extraction techniques. These techniques are also known as Named Entity Recognition (NER). Hearst [Hearst 1992, Hearst 1998] proposes one of the pioneer domain-independent approaches to extract named entities and their classes. The author identifies 6 lexico-syntactic patterns which detect this relation. Other approaches can be found in [Guo et al. 2009] or [Etzioni et al. 2005]. The class-instance relation is also targeted in TREC Entity Tracks [Balog et al. 2009b]. One of the proposed tasks involves returning a list of named entities of a given class.

Existing techniques for this relation are promising, although they are mostly precision oriented.

6.3.2 *Instance-instance relation.* Inspired from [Suchanek et al. 2008], we distinguish four main relations that can relate an instance to another namely: synonymy, sibling relation, meronymy and non-taxonimic relations. To illustrate, "Big apple" is a synonym for "New York City". France and Italy are siblings in that they are instances of the same class "countries". Meronymy involves part-of-a-whole relations such as ("Italy", *is a member of*, "NATO"). The non-taxonomic relations are relations between two instances given by textual description such as: ("John Travola", *plays in*, "Pulp Fiction"), ("Windows", *is a product of*, "Microsoft").

Suchanek et al. [2007] presented YAGO, a large extensible ontology built through careful combination of heuristics on Wordnet and Wikipedia. They extract synonyms, instance-class relations (hyponomy) and non-taxonomic relations. The result is a set of about 1 million instances and about 5 million relations. Instance-instance relations can be extracted in a supervised [Agichtein and Gravano 2000; Etzioni et al. 2008] or unsupervised way [Etzioni et al. 2005]. Instance-instance relations are mostly extracted through lexico-syntactic rules based on term statistics and part-of-speech tags. Some of these techniques are particularly interesting in that they are appliable at large scale and domain-independent. Nevertheless, some relations are difficult to capture as their extraction depends on the training data or the technique being used.

6.3.3 *Instance-attribute relation.* Attribute acquisition can be domain-independent or domain-dependent. From domain-dependent approaches, we may mention approaches that focus on products. In this domain, attributes have been used to improve product search and recommendation [Nie et al. 2007], but also to enable data mining [Wong and Lam 2009]. To acquire attributes from the Web, it is common to use decoration markup [Yoshinaga and Torisawa 2007; Crescenzi et al. 2001] and text [Ben-Yitzhak et al. 2008, Tokunaga and Torisawa 2005]. A common technique to acquire attributes is through the use of lexico-syntactic rules [Pasca and Durme 2008; Alfonseca et al. 2010; Almuhareb and Poesio 2004; Popescu and Etzioni 2005].

HTML tags (for tables, lists and emphasis) have also been shown to help for attribute acquisition [ Yoshinaga and Torisawa 2007; Wong and Lam 2009]. In particular, HTML tables are known to be a mine for relational data and attributes. An approach to extract attributes from tables using column (or row) similarity is proposed in [Chen et al. 2000]. Another common technique is trough wrapper induction [Crescenzi et al. 2001; Wong and Lam 2004]. Cafarella et al. [2008; 2009] show that we can identify billions of relational tables in the Web. Kopliku et al. [2011b; 2011a] show how to extract and rank attributes from Web tables for whatever instance query combining table classification with relevance ranking.

At last, knowledge bases (such as DBPedia or Freebase) [Suchanek et al. 2007, Wu et al. 2008, Kopliku 2011] provide millions of relations in the form of linked data which make it easy to retrieve attributes.

Attribute retrieval from linked data will probably attract more research attention given the fast growth of user generated or automatically extracted linked data [Bizer et al. 2009, Bizer et al. 2008].

6.3.4 *Other nugget-nugget relations.* The nugget-nugget relation is the broader class of relations which can involve whatever type of information nugget i.e it includes instance-instance relations, class-instance relations and attribute-instance relations. We will list here just some other broad examples without aiming to be exhaustive:

—**Similarity** ("similar to"): This is a common relation which has already been targeted in clustering, classification, news aggregators, etc. For instance, news aggregators group similar news articles into stories.

—**Diversity** ("different to"): The inverse of similarity represents another useful relation. Studies on novelty and diversity claim that it is sometimes better to promote some diversity among the retrieved results.

—**Space-time** ("happens in time/space"): Content can also relate with respect to some features such as time and location. For instance, sometimes it is better to order information chronologically to favor freshness of information [Dong et al. 2010]. In Geographic Information Retrieval the location feature is fundamental to organize and visualize search results [Jones and Purves 2009].

## 6.4 Result aggregation

This section is about ways to aggregate search results in the context of relational aggregated search. We will consider one query type at a time.

Attribute queries (e.g. capital of France) should ideally be answered with the correct attribute value/s. However for many queries we might find many candidate values without being certain on their relevance and correctness. With respect to this issue, there is a lot of work in question answering (what is the capital of Congo?). Here, the result is one or more candidate answers, typically associated with supporting text.

Instance queries provide more space for result aggregation. In object-level search, these queries are answered through records and attributes extracted from one or more pages [Nie et al. 2007]. The answer can also be a set of related instances [Basu Roy et al. 2010, Kato et al. 2009], multimedia content [Taneva et al. 2010], passages [Sauper and Barzilay 2009], a summary of attributes [Kopliku et al. 2011b, Kopliku et al. 2011b]. Specific approaches have been adapted for people search [Macdonald 2009], product search [Nie et al. 2007; Popescu and Etzioni 2005], bibliographic search [Ji et al. 2009], etc.

Class queries can be answered with a list of instances. This is the case for most approaches that can be found in Question Answering [Kelly and Lin 2006] and TREC entity tracks [Balog et al. 2009b]. Google Maps[12] answers location-related class queries (e.g. hotels in Las Vegas) with a list of instances in a map sometimes associated with ratings, reviews and images. In [Kopliku et al. 2011b, Krichen et al. 2011], structure results of class queries in tables with instances in the rows and attributes in the columns.

We can conclude that there are several ways to answer queries in the relational framework. The quality of the result aggregation depends on the quality of the relations and the relevance of the result components.

## 7. EVALUATION OF AGGREGATED SEARCH

Aggregated search in its broadest definition is difficult to evaluate i.e. an aggregate answer can be organized in many ways and contain whatever information nuggets (images, sentences, Web search results, information extracts, ...). Aggregated search can provide immediate and organized access to information. The quantity of relevant information, its diversity, coherence and organization can affect directly user satisfaction.

Although the evaluation of aggregated search is not a solved question, some preliminary works or evaluation frameworks can however be mentioned in the context of relational aggregated search and cross-vertical aggregated search: they are discussed in the following two sections.

### 7.1 Cross-vertical aggregated search

Although the process of cross-vertical aggregated search is clearly composed of vertical selection, item selection and results presentation, evaluation does not concern so clearly these different parts. Vertical selection and result presentation are in fact strongly linked [Zhou et al. 2012].

Among approaches aiming at mainly evaluating vertical selection, one can cite [Arguello et al. 2009, Li et al. 2008, Diaz 2009a] or [Kopliku et al. 2011]. Evaluation

[12]http://maps.google.com

protocols are based either on manual assessments or on user interaction data (click through).

Aggregated search interfaces (blended vs unblended) are evaluated in [Sushmita et al. 2010, Arguello et al. 2012] using user studies. A methodology for evaluating results presentation is also defined in [Arguello et al. 2011] : authors propose a metric-based evaluation of any possible presentation of the query. The idea is to use preference judgments on pairs of blocks to construct a reference presentation for the query. Approaches can then be evaluated by calculating their distance to the reference presentation. Although the proposed approach only considered ranking by block, the methodology can be applied to every approach.

At last in [Zhou et al. 2012], authors propose to evaluate the aggregated search process as a whole, by proposing an evaluation framework using a test collection [Zhou et al. 2011] as well as evaluation metrics that are showed to be more suitable for evaluated cross-vertical aggregated search than traditional IR metrics.

To summarize, cross-vertical aggregated search is today mainly evaluated by studying the aggregated search behavior, either using laboratory studies ([Arguello et al. 2012], [Arguello and Capra 2012], [Sushmita et al. 2010]), or search logs-data ([Diaz 2009a]). The community is however working on the crucial point of reusable test collection, as shown by the first efforts published in [Arguello et al. 2011, Zhou et al. 2012].

### 7.2 Relational aggregated search

In the context of relational aggregated search, existing frameworks for evaluation mainly concern the evaluation of entity search. Among them one can cite:

—the INEX Entity Ranking track (2007-2009): the track proposed a framework to evaluate engines that return lists of entities. Two main tasks were considered on structured documents collections: entity ranking (ER) and entity list completion (LC) [Demartini et al. 2010]

—The TREC Entity track (2009-2011): the aim of the track was similar to the INEX one (i.e. to perform entity-oriented search tasks) but on Web data (web pages or Linked Open Data) [Balog et al. 2009a, Balog et al. 2010, 2011]. In 2011 for example, the ClueWeb 2009 web corpus and the Sindice-2011 data set [Campinas et al. 2011] were used as corpora. Two main tasks were evaluated (Related Entity Finding task and Entity List Completion task), whose aim was to find related entities of a given entity regarding a given relation.

—the SemSearch challenge (2010-2011): this challenge aimed at evaluating semantic search systems for entity search on a collection which is a sample of Linked Data (RDF triples) crawled from publicly available sources [Halpin et al. 2010, sem a, b]

—the TAC KBP (Knowledge Base Population) track, which first run in 2009 : this track evaluates the discovering of information about named entities in large corpus and its integration into a knowledge base [TAC 2011]. The number of tasks in the track increase with time: in 2013 for example, 7 tasks are proposed, among them one can cite the Entity Linking track, the Slot Filling task or the Cold Start KBP.

—the TREC KBA track, which first run in 2012: as the TAC KBP track, TREC KBA aims at helping the construction and maintenance of large Knowledge Bases [Frank et al. 2012]. In 2013, two tasks are proposed: (i) the Cumulative Citation Recommendation -CCR) task, in which systems should filter in a time-ordered corpus documents that are centrally relevant to a given entity; and (ii) the Streaming Slot Filling (SSF) task in which given a slot for a target entity, systems should detect changes to the slot value.

The preceding evaluation campaigns allow thus to evaluate approaches targeting the extraction of instance-class or instance-instance relation. The evaluation of attribute extraction do not fit in these campaigns (except maybe for the TAC KBP and TREC KBA SSF tasks which are interested in attribute values), and existing approaches for attribute retrieval were mainly evaluated on their own test set (see [Kopliku et al. 2011b] for instance). To conclude, the utility of RAS is not just in the relevance of information, it is also in the focus and organisation. Few studies address the utility of relational information as a whole or as a complement to traditional search results.

## 8. CONCLUSION AND FUTURE DIRECTIONS

### 8.1 Conclusion

This survey shows that there is a myriad of approaches that go beyond query matching with an additional effort on result aggregation. We have analyzed many of these approaches from multiple research domains in the light of aggregated search. Some approaches have strong considerations on the data they process (the sources in the database community should own a schema; RDF triples are queried in semantic search, vertical searches act on very specific data), whereas others are more concerned with textual information (that can or not be structured): this is the case of federated search, meta-search, information extraction, question answering or natural language generation.

Three processes are shown to decompose well the broad aggregated search process namely: *query dispatching*, *nugget retrieval* and *result aggregation*. They encapsulate well the different existing query matching approaches with an additional attention at query time and result aggregation time.

Among the listed approaches, there are many that rely on a multiple-sources (search engines) architecture. Among them, *cross-vertical aggregated search* seems to be the most successful; it has met commercial success in Web search and it is capturing an increasing interest in research. Here, the integration of diverse vertical search engines (image search, video search, ...) has been shown beneficial i.e. it increases the amount of relevant results and it often provides complementary results.

In addition, we have also analyzed research on *relational aggregated search* another promising research direction. The latter makes use of relations to assemble and retrieve search results. This enables retrieving at finer granularity and composing new aggregated objects of related content.

## 8.2 Future directions

We can expect aggregated search to attract research direction for the next decades. In this section, we list some challenging (and promising) research directions.

Starting from the broad aggregated search framework, we believe the most challenging issues concern query dispatching and result aggregation. Indeed, nugget retrieval has been the research target for more than 50 years now. Identifying queries that benefit from aggregation as well as new ways to aggregate content remain novel and promising.

Another important issue is evaluation. Evaluation in the context of cvAS and RAS concerns mainly the effectiveness of each sub-task that composes the different approaches. In other terms, relevance is mainly assessed on single information nuggets taken separately (even if a noticeable exception can be recently found in the context of cross-vertical aggregated search [Zhou et al. 2012]). The quality of the aggregated results should however be measured by considering the complementarity of nuggets, coherence, completeness, or coverage of the information need, and appropriate metrics are still needed. Moreover, finding a way to measure the quality of any possible aggregate for a given query is a problem that is far from being solved.

At last and as stated in [Arguello et al. 2012], a promising perspective concerns the personalization of aggregated results, in order to make users be the "the core content".

If we keep to specific research directions, we believe that cross-vertical aggregated search and relational aggregated search remain the most promising in the short term.

Despite the success of *cross-vertical aggregated search* in the context of Web search, we belive its potential is far from being completely exploited. There is evidence that we can combine sources such as Wikipedia, DBPedia, images, maps to form aggregate unexisting objects with quite some visualization flexibility. As well, we can target new applications other than Web search. We can foresee enriched map search results, enriches Wikipedia results and so on.

In the context of *relational aggregated search*, there is a lot of work to be done. We need to improve relation retrieval techniques in both terms of precision and recall. Depending on the targeted application, a good precision/recall is fundamental. We believe that two research directions are more promising at this moment. First, there is an increasing effort towards open-domain and large-scale information extraction techniques. Second, we can observe a fast growth of quality linked data in the Web. We believe that the combination of automatic extraction and user-generated content will produce enough relational and semantic data for multiple commercial uses. Moreover, a substantial effort is needed to define coherent ways to explore and aggregate results (tables, lists, graphs, ...).

To conclude, we believe that this survey in conjunction with other ongoing research indicate that future IR can integrate more focus, structure, and semantics in search results.

## REFERENCES


Semantic search challenge 2010. http://km.aifb.kit.edu/ws/semsearch10/, Retrieved: April 2013.

Semantic search challenge 2011. http://semsearch.yahoo.com/, Retrieved: April 2013.

Aditya, P. and Jaya, K. 2008. Leveraging query association in federated search. In *SIGIR 2008 Workshop on aggregated search*.

Agichtein, E. and Gravano, L. 2000. Snowball: extracting relations from large plain-text collections. In *DL '00: Proc. of the fifth ACM conference on Digital libraries*. 85–94.

Alfonseca, E., Pasca, M., and Robledo-Arnuncio, E. 2010. Acquisition of instance attributes via labeled and related instances. In *Proc. of SIGIR '10*. 58–65.

Almuhareb, A. and Poesio, M. 2004. Attribute-based and value-based clustering: An evaluation. In *In EMNLP '04, ACL*. 158–165.

Arguello, J. and Capra, R. 2012. The effect of aggregated search coherence on search behavior. In *Proc. of CIKM'12, Maui, HI, USA*. 1293–1302.

Arguello, J., Diaz, F., and Callan, J. 2011. Learning to aggregate vertical results into web search results. In *Proc. of CIKM 2011*. 201–210.

Arguello, J., Diaz, F., Callan, J., and Carterette, B. 2011. A methodology for evaluating aggregated search results. In *Proc. of ECIR 2011, Dublin, Ireland*. 141–152.

Arguello, J., Diaz, F., Callan, J., and Crespo, J.-F. 2009. Sources of evidence for vertical selection. In *SIGIR '09: Proc. of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 315–322.

Arguello, J., Diaz, F., and Paiement, J.-F. 2010. Vertical selection in the presence of unlabeled verticals. In *Proc. of SIGIR '10, Geneva, Switzerland*. 691–698.

Arguello, J., Diaz, F., and Shokouhi, M. 2012. Integrating and ranking aggregated content on the web. In *WWW 2012, Tutorial, Lyon, http://ils.unc.edu/~jarguell/www12_content_agg/*.

Arguello, J., Wu, W.-C., Kelly, D., and Edwards, A. 2012. Task complexity, vertical display and user interaction in aggregated search. In *Proc. of SIGIR '12, Portland, OR, USA,*. 435–444.

Aumann, Y., Feldman, R., Liberzon, Y., Rosenfeld, B., and Schler, J. 2006. Visual information extraction. *Knowl. Inf. Syst. 10*, 1–15.

Avrahami, T. T., Yau, L., Si, L., and Callan, J. 2006. The fedlemur project: Federated search in the real world. *JASIST 57,* 3, 347–358.

Balog, K., de Vries, A., Serdyukov, P., Thomas, P., and Westerveld, T. 2009a. Overview of the trec 2009 entity track. In *Proc. of TREC 2009*.

Balog, K., de Vries, A. P., Serdyukov, P., Thomas, P., and Westerveld, T. 2009b. Overview of the trec 2009 entity track. In *TREC 2009 Working Notes*. NIST.

Balog, K., Serdyukov, P., and de Vries, A. 2010. Overview of the trec 2010 entity track. In *Proc. of TREC 2010*.

Balog, K., Serdyukov, P., and de Vries, A. 2011. Overview of the trec 2010 entity track. In *Proc. of TREC 2011*.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. 2007. Open information extraction from the web. In *Proc. of IJCAI 2007*. 2670–2676.

Basu Roy, S., Amer-Yahia, S., Chawla, A., Das, G., and Yu, C. 2010. Constructing and exploring composite items. In *Proc. of SIGMOD 2010, Indianapolis, Indiana, USA*. 843–854.

Baziz, M., Boughanem, M., Loiseau, Y., and Prade, H. 2007. Fuzzy Logic and Ontology-based Information Retrieval. In *Studies in Fuzziness and Soft Computing*. Vol. 215/2007. 193–218.

Bellare, K., Talukdar, P. P., Kumaran, G., Pereira, O., Liberman, M., Mccallum, A., and Dredze, M. 2007. Lightly-supervised attribute extraction for web search. In *Proc. of Machine Learning for Web Search Workshop, NIPS 2007*.

Ben-Yitzhak, O., Golbandi, N., Har'El, N., Lempel, R., Neumann, A., Ofek-Koifman, S., Sheinwald, D., Shekita, E., Sznajder, B., and Yogev, S. 2008. Beyond basic faceted search. In *Proc. of WSDM '08, Palo Alto, California, USA*. 33–44.

Bizer, C., Heath, T., and Berners-Lee, T. 2009. Linked data - the story so far. *International Journal Semantic Web and Information Systems 5,* 3, 1–22.

Bizer, C., Heath, T., Idehen, K., and Berners-Lee, T. 2008. Linked data on the web (ldow2008). In *Proc. of WWW 2008, Beijing, China*. WWW '08. 1265–1266.

Bouidghaghen, O., Tamine, L., and Boughanem, M. 2009. Dynamically Personalizing Search Results for Mobile Users. In *Flexible Query Answering (FQAS)*. 99–110.

Boyce, B. R. 1982. Beyond topicality : A two stage view of relevance and the retrieval process. *Inf. Process. Manage. 18,* 3, 105–109.

Cafarella, M. J., Banko, M., and Etzioni, O. 2006. Relational web search. Tech. rep., University of Washington.

Cafarella, M. J., Halevy, A., Wang, D. Z., Wu, E., and Zhang, Y. 2008. Webtables: exploring the power of tables on the web. *Proc. VLDB Endow. 1,* 1, 538–549.

Cafarella, M. J., Halevy, A. Y., and Khoussainova, N. 2009. Data integration for the relational web. *PVLDB 2,* 1, 1090–1101.

Callan, J. 2000. Distributed information retrieval. In *Advances in Information Retrieval*, W. B. Croft, Ed. Kluwer Academic Publishers, Dordrecht, 235–266.

Campinas, S., Ceccarelli, D., Perry, T. E., Delbru, R., Balog, K., and Tummarello, G. 2011. The sindice-2011 dataset for entity-oriented search in the web of data. In *1st International Workshop on Entity-Oriented Search (EOS)*. 26–32.

Chang, C.-C. and Lin, C.-J. 2001. LIBSVM: a library for support vector machines. Tech. rep.

Chen, H.-H., Tsai, S.-C., and Tsai, J.-H. 2000. Mining tables from large scale html texts. In *Proc. of COLING 2000*. 166–172.

Church, K. W. and Hanks, P. 1989. Word association norms, mutual information, and lexicography. In *Proc. of ACL 98, Vancouver, British Columbia, Canada*. 76–83.

Clarke, C. L. A., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. 2008. Novelty and diversity in information retrieval evaluation. In *Proc. of SIGIR 2008*. 659–666.

Crescenzi, V., Mecca, G., and Merialdo, P. 2001. Roadrunner: Towards automatic data extraction from large web sites. In *Proc. of VLDB 2001*. 109–118.

Dang, H., Kelly, D., and Lin, J. 2007. Overview of the trec 2007 question answering track. In *TREC 2007 Proc.*

Dang, H. T. 2006. Overview of DUC 2006. In *Proc. of the 2006 Document Understanding Conference*.

Demartini, G., Iofciu, T., and Vries, A. 2010. Overview of the inex 2009 entity ranking track. In *Focused Retrieval and Evaluation*. 254–264.

Diaz, F. 2009a. Integration of news content into web results. In *Proc. of WSDM 2009, Barcelona, Spain*. 182–191.

Diaz, F. 2009b. Integration of news content into web results. In *WSDM*. 182–191.

Diaz, F. and Arguello, J. 2009. Adaptation of offline vertical selection predictions in the presence of user feedback. In *Proc. of SIGIR '09, Boston, MA, USA*. 323–330.

Dong, A., Chang, Y., Zheng, Z., Mishne, G., Bai, J., Zhang, R., Buchner, K., Liao, C., and Diaz, F. 2010. Towards recency ranking in web search. In *Proc. of WSDM '10, New York, New York, USA*. 11–20.

Downey, D., Etzioni, O., and Soderland, S. 2005. A probabilistic model of redundancy in information extraction. In *Proceedigns of IJCAI*. 1034–1041.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. 2008. Open information extraction from the web. *Commun. ACM 51*, 68–74.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell. 165,* 1, 91–134.

Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam. 2011. Open information extraction: The second generation. In *Proc. of IJCAI 2011, Barcelona, Spain*. 3–10.

Frank, J. R., Kleiman-Weiner, M., Roberts, D. A., Niu, F., Zhang, C., Re, C., and Soboroff, I. 2012. Building an entity-centric stream filtering test collection for trec 2012. In *Proc. of TREC 2012*.

Geva, S., Kamps, J., and Trotman, A., Eds. 2009. *Proc. of INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008*.

Goecks, J. 2002. Nuggetmine: Intelligent groupware for opportunistically sharing information nuggets. In *Proc. of IUI '02*. 87–94.

Gravano, L., Chang, C.-C. K., García-Molina, H., and Paepcke, A. 1997. Starts: Stanford proposal for internet meta-searching. *SIGMOD Rec. 26*, 207–218.

Greenshpan, O., Milo, T., and Polyzotis, N. 2009. Autocompletion for mashups. *Proc. VLDB Endow. 2,* 1, 538–549.

Grice, H. P. 1975. Logic and conversation. In *Syntax and Semantics: Vol. 3: Speech Acts*, P. Cole and J. L. Morgan, Eds. Academic Press, San Diego, CA, 41–58.

Grishman, R. and Sundheim, B. 1996. Message understanding conference-6: a brief history. In *Proc. of the 16th conference on Computational linguistics*. 466–471.

Gulli, A. and Signorini, A. 2005. Building an open source meta-search engine. In *Proc. of WWW '05: Special interest tracks and posters*. 1004–1005.

Guo, J., Xu, G., Cheng, X., and Li, H. 2009. Named entity recognition in query. In *Proc. of SIGIR 2009*. 267–274.

Halpin, H., Herzig, D. M., Mika, P., Blanco, R., Pound, J., Thompson, H. S., and Tran, D. T. 2010. Evaluating ad-hoc object retrieval. In *Proc. of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010)*.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th conference on Computational linguistics*. 539–545.

Hearst, M. A. 1998. Automated discovery of wordnet relations. In *C. Fellbaum, WordNet: An Electronic Lexical Database*. MIT Press, 131–153.

Hearst, M. A. and Pedersen, J. O. 1996. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *SIGIR*. 76–84.

Hennig, S. and Wurst, M. 2006. Incremental clustering of newsgroup articles. In *IEA/AIE*. 332–341.

Hirschman, L. and Gaizauskas, R. 2001. Natural language question answering: the view from here. *Natural Language Engineering 7*, 275–300.

Ji, L., Yan, J., Liu, N., Zhang, W., Fan, W., and Chen, Z. 2009. Exsearch: a novel vertical search engine for online barter business. In *Proc. of CIKM '09*. 1357–1366.

Jones, C. B. and Purves, R. S. 2009. Geographical information retrieval. In *Encyclopedia of Database Systems*. 1227–1231.

Kamps, J., Geva, S., and Trotman, A. 2008. Report on the SIGIR 2008 workshop on focused retrieval. *SIGIR Forum 42,* 2, 59–65.

Kaptein, R. and Marx, M. 2010. Focused retrieval and result aggregation with political data. *Inf. Retr. 13*, 412–433.

Kato, M. P., Ohshima, H., Oyama, S., and Tanaka, K. 2009. Query by analogical example: relational search using web search engine indices. In *Proc. of CIKM '09*. 27–36.

Katz, B., Borchardt, G., and Felshin, S. 2005. Syntactic and semantic decomposition strategies for question answering from multiple resources. In *Proc. of the AAAI 2005 workshop on inference for textual question answering. Pittsburgh, Pennsylvania, USA*.

Kelly, D. and Lin, J. 2006. Overview of the TREC 2006 question answering task. In *In Text REtrieval Conference 2006*.

Kelly, D. and Lin, J. 2007. Overview of the TREC 2007 question answering task. In *In Text REtrieval Conference 2007*.

Kennedy, L. S. and Naaman, M. 2008. Generating diverse and representative image search results for landmarks. In *Proc. of WWW '08, Beijing, China*. 297–306.

Kopliku, A. 2011. Thèse de doctorat. Ph.D. thesis, Université Paul Sabatier, Toulouse, France.

Kopliku, A., Boughanem, M., and Pinel-Sauvagnat, K. 2011a. Mining the Web for lists of Named Entities. In *Proc. of CORIA 2011, Avignon, France*. 113–120.

Kopliku, A., Boughanem, M., and Pinel-Sauvagnat, K. 2011b. Towards a framework for attribute retrieval. In *Proc. of CIKM 2011*. 515–524.

Kopliku, A., Damak, F., Pinel-Sauvagnat, K., and Boughanem, M. 2011. Interest and Evaluation of Aggregated Search. In *Proc. of IEEE/WIC/ACM International Conference on Web Intelligence, Lyon, France*. 154–161.

Kopliku, A., Pinel-Sauvagnat, K., and Boughanem, M. 2009. Aggregated search: Potential, issues and evaluation. Tech. rep., Institut de Recherche en Informatique de Toulouse, France.

Kopliku, A., Pinel-Sauvagnat, K., and Boughanem, M. 2011a. Attribute retrieval from relational web tables. In *Proc. of SPIRE 2011*. 117–128.

Kopliku, A., Pinel-Sauvagnat, K., and Boughanem, M. 2011b. Retrieving attributes using web tables. In *Proc. of JDCL 2011*. 397–398.

Kopliku, A., Thomas, P., Wan, S., and Paris, C. 2013. Filtering and ranking for social media monitoring. In *Proc. of CORIA 2013, Neuchâtel, Switzerland*.

Krichen, I., Kopliku, A., Pinel-Sauvagnat, K., and Boughanem, M. 2011. Une approche de recherche d'attributs pertinents pour l'agrégation d'information . In *Proc. of INFORSID 2009, Lille, France*. 385–400.

Lalmas, M. 2011. Advanced topics on information retrieval. Springer, Chapter Aggregated search.

Lenzerini, M. 2002. Data integration: a theoretical perspective. In *Proc. of PODS 2002*. 233–246.

Li, X., Wang, Y.-Y., and Acero, A. 2008. Learning query intent from regularized click graphs. In *Proc. of SIGIR 2008, Singapore, Singapore*. 339–346.

Limaye, G., Sarawagi, S., and Chakrabarti, S. 2010. Annotating and searching web tables using entities, types and relationships. *Proc. VLDB Endow. 3*, 1338–1347.

Lin, C.-J. and Liu, R.-R. 2008. An analysis of multi-focus questions. In *SIGIR workshop on focused retrieval, Singapore*.

Lin, T. and Etzioni, O. 2010. Identifying functional relations in web text. In *Proc. of EMNLP 2010*.

Liu, N., Yan, J., and Chen, Z. 2009. A probabilistic model based approach for blended search. In *Proc. of WWW '09, Madrid, Spain*. 1075–1076.

Macdonald, C. 2009. The voting model for people search. *SIGIR Forum 43*, 73–73.

Manning, C., Raghavan, P., and Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

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

McCurley, K. S. 2001. Geospatial mapping and navigation of the web. In *WWW '01: Proc. of the 10th international conference on WWW*. ACM, New York, NY, USA, 221–229.

Meng, X., Wang, H., Hu, D., and Li, C. 2003. A supervised visual wrapper generator for web-data extraction. In *Proc. of the 27th Annual International Conference on Computer Software and Applications*. COMPSAC '03. IEEE Computer Society, Washington, DC, USA, 657–.

Moriceau, V. and Tannier, X. 2010. Fidji: using syntax for validating answers in multiple documents. *Inf. Retr. 13*, 507–533.

Murdock, V. and Lalmas, M. 2008. Workshop on aggregated search. *SIGIR Forum 42,* 2, 80–83.

Naaman, M., Song, Y. J., Paepcke, A., and Garcia-Molina, H. 2006. Assigning textual names to sets of geographic coordinates. *Computers, Environment and Urban Systems 30,* 4, 418–435.

Nie, Z., Ma, Y., Shi, S., Wen, J.-R., and Ma, W.-Y. 2007. Web object retrieval. In *Proc. of WWW '07, Banff, Alberta, Canada*. 81–90.

Nie, Z., Wen, J.-R., and Ma, W.-Y. 2007. Object-level vertical search. In *CIDR*. www.crdrdb.org, 235–246.

Ou, S. and Khoo, C. S. G. 2008. Aggregating search results for social science by extracting and organizing research concepts and relations. In *SIGIR 2008 Workshop on aggregated search*.

Paris, C. and Colineau, N. 2006. Scifly: Tailored corporate brochures on demand. Tech. rep., CSIRO ICT Centre.

Paris, C., Lampert, A., Lu, S., and Wu, M. 2005. Enhancing dynamic knowledge management services - tailored documents. Tech. rep., CSIRO ICT Centre. Technical Report 05/034, Commercial-in-Confidence.

Paris, C., Wan, S., and Thomas, P. 2010. Focused and aggregated search: a perspective from natural language generation. *Information Retrieval Journal 44,* 3.

Paris, C., Wan, S., Wilkinson, R., and Wu, M. 2001. Generating personal travel guides - and who wants them? In *User Modeling 2001, Germany, July 13-17, 2001, Proc.* 251–253.

Paris, C. L. 1988. Tailoring object descriptions to a user's level of expertise. *Comput. Linguist. 14,* 3, 64–78.

Pasca, M. and Durme, B. V. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *ACL*. 19–27.

Ponnuswami, A. K., Pattabiraman, K., Wu, Q., Gilad-Bachrach, R., and Kanungo, T. 2011. On composition of a federated web search result page: using online users to provide pairwise preference for heterogeneous verticals. In *Proc. of WSDM 2011, Hong Kong, China*. 715–724.

Ponte, J. M. and Croft, W. B. 1998. A language modeling approach to information retrieval. In *Proc. of SIGIR 98, Melbourne, Australia*. 275–281.

Popescu, A.-M. and Etzioni, O. 2005. Extracting product features and opinions from reviews. In *Proc. of HLT 2008, Vancouver, British Columbia, Canada*. 339–346.

Pound, J., Hudek, A. K., Ilyas, I. F., and Weddell, G. 2012. Interpreting keyword queries over web knowledge bases. In *Proc. of CIKM 2012, Maui, Hawaii, USA*. 305–314.

Ranganathan, A., Riabov, A., and Udrea, O. 2009. Mashup-based information retrieval for domain experts. In *Proc. of CIKM '09, Hong Kong, China*. 711–720.

Robertson, S. E. and Walker, S. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proc. of SIGIR 1994. Dublin, Ireland (Special Issue of the SIGIR Forum)*, W. B. Croft and C. J. van Rijsbergen, Eds. 232–241.

Rohr, C. and Tjondronegoro, D. 2008. Aggregated cross-media news visualization and personalization. In *Proc. of MIR 2008, Vancouver, British Columbia, Canada*. 371–378.

Sahoo, N., Callan, J., Krishnan, R., Duncan, G., and Padman, R. 2006. Incremental hierarchical clustering of text documents. In *Proc. of CIKM 2006, Arlington, Virginia, USA*. 357–366.

Salton, G., Wong, A., and Yang, C. S. 1975. A vector space model for automatic indexing. *Commun. ACM 18*, 613–620.

Santos, R. L. T., Macdonald, C., and Ounis, I. 2011. Aggregated search result diversification. In *Proc. of the 3rd International Conference on the Theory of Information Retrieval*. Springer, Bertinoro, Italy.

Sauper, C. and Barzilay, R. 2009. Automatically generating wikipedia articles: a structure-aware approach. In *Proc. of ACL-IJCNLP 09, Suntec, Singapore*. 208–216.

Sauvagnat, K., Boughanem, M., and Chrisment, C. 2006. Answering content-and-structure-based queries on XML documents using relevance propagation. *Information Systems, Special Issue SPIRE 2004 31*, 621–635.

Sekine, S., Sudo, K., and Nobata, C. 2002. Extended named entity hierarchy. In *Proc. of LREC 2002*.

Selberg, E. and Etzioni, O. 1995. Multi-service search and comparison using the metacrawler. In *In Proc. of the 4th International World Wide Web Conference*. 195–208.

Shokouhi, M., Zobel, J., Tahaghoghi, S., and Scholer, F. 2007. Using query logs to establish vocabularies in distributed information retrieval. *Inf. Process. Manage. 43*, 169–180.

Singhal, A. 2012. Introducing the knowledge graph: Things, not strings. Official Blog of Google, `http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html`. Retrieved April 2013.

Spärck-Jones, K., Robertson, S. E., and Sanderson, M. 2007. Ambiguous requests: implications for retrieval tests, systems and theories. *SIGIR Forum 41,* 2, 8–17.

Srinivas, K., Srinivas, P. V. S., and Govardhan, A. 2011. A survey on the performance evaluation of various meta search engines. *Int. Journal of Computer Science Issues 8*, 359–364.

Strotmann, A. and Zhao, D. 2008. Bibliometric maps for aggregated visual browsing in digital libraries. In *SIGIR 2008 Workshop on aggregated search*. ACM.

Suchanek, F. M., Kasneci, G., and Weikum, G. 2007. Yago: a core of semantic knowledge. In *Proc. of WWW 2007, Banff, Alberta, Canada*. 697–706.

Suchanek, F. M., Kasneci, G., and Weikum, G. 2008. YAGO: A large ontology from Wikipedia and Wordnet. *Web Semant. 6,* 3, 203–217.

Sushmita, S., Joho, H., and Lalmas, M. 2009. A task-based evaluation of an aggregated search interface. In *Proc. of SPIRE 2009, Saariselkä, Finland*. 322–333.

Sushmita, S., Joho, H., Lalmas, M., and Villa, R. 2010. Factors affecting click-through behavior in aggregated search interfaces. In *Proc. of CIKM 2010, Toronto, Ontario, Canada*. 519–528.

TAC. 2011. Proc. of the fourth text analysis conference. National Institute of Standards and Technology Gaithersburg, Maryland, USA.

Taneva, B., Kacimi, M., and Weikum, G. 2010. Gathering and ranking photos of named entities with high precision, high recall, and diversity. In *Proc. of WSDM 2010*. 431–440.

Tokunaga, K. and Torisawa, K. 2005. Automatic discovery of attribute words from web documents. In *In Proc. of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05), pages 106-118, Jeju Island, Korea*. 106–118.

Tran, T., Wang, H., Rudolph, S., and Cimiano, P. 2009. Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In *Proc. of ICDE 2009*. 405–416.

Trotman, A., Geva, S., Kamps, J., Lalmas, M., and Murdock, V. 2010. Current research in focused retrieval and result aggregation. *Journal of Information Retrieval*.

Turney, P. D. 2001. Mining the Web for synonyms: PMI-IR versus lsa on toefl. In *Proc. of EMCL 2001*. 491–502.

Vaid, S., Jones, C. B., Joho, H., and Sanderson, M. 2005. Spatio-textual indexing for geographical search on the web. In *SSTD*. 218–235.

Vallet, D. and Zaragoza, H. 2008. Inferring the most important types of a query: a semantic approach. In *Proc. of SIGIR 2008, Singapore, Singapore*. 857–858.

Voorhees, E. M. 2003. Evaluating answers to definition questions. In *Proceedigns of NAACL '03, Edmonton, Canada*. 109–111.

Wong, T.-L. and Lam, W. 2004. A probabilistic approach for adapting information extraction wrappers and discovering new attributes. In *Proc. of ICDM 2004*. 257–264.

Wong, T.-L. and Lam, W. 2009. An unsupervised method for joint information extraction and feature mining across different web sites. *Data Knowl. Eng. 68*, 107–125.

Wu, F., Hoffmann, R., and Weld, D. S. 2008. Information extraction from wikipedia: moving down the long tail. In *Proc. of KDD 2008, Las Vegas, Nevada, USA*. 731–739.

Wu, M. and Fuller, M. 1997. Supporting the answering process. In *Proc. of the Second Australian Document Computing Symposium*. 65–73.

Yoshinaga, N. and Torisawa, K. 2007. Open-domain attribute-value acquisition from semi-structured texts. In *Proc. of the workshop on Ontolex*. 55–66.

Zhou, K., Cummins, R., Lalmas, M., and Jose, J. M. 2011. Evaluating large-scale distributed vertical search. In *LSDS-IR workshop in CIKM*.

Zhou, K., Cummins, R., Lalmas, M., and Jose, J. M. 2012. Evaluating aggregated search pages. In *Proc. of SIGIR 2012, Portland, OR, USA*. 115–124.

Zhu, J., Nie, Z., Liu, X., Zhang, B., and Wen, J.-R. 2009. Statsnowball: a statistical approach to extracting entity relationships. In *Proc. of WWW 2009, Madrid, Spain*. 101–110.