Design Patterns for Fusion-Based Object Retrieval

Shuo Zhang and Krisztian Balog[(✉)]

University of Stavanger, Stavanger, Norway
{shuo.zhang,krisztian.balog}@uis.no

Abstract. We address the task of ranking objects (such as people, blogs, or verticals) that, unlike documents, do not have direct term-based representations. To be able to match them against keyword queries, evidence needs to be amassed from documents that are associated with the given object. We present two design patterns, i.e., general reusable retrieval strategies, which are able to encompass most existing approaches from the past. One strategy combines evidence on the term level (early fusion), while the other does it on the document level (late fusion). We demonstrate the generality of these patterns by applying them to three different object retrieval tasks: expert finding, blog distillation, and vertical ranking.

1 Introduction

Viewed broadly, information retrieval is about matching information objects against information needs. In the classical ad hoc document retrieval task, information objects are documents and information needs are expressed as keyword queries. This task has been a main focal point since the inception of the field. The past decade, however, has seen a move beyond documents as units of retrieval to other types of objects. Examples of object retrieval tasks studied at the Text REtrieval Conference (TREC) include ranking people (experts) [1,4], blogs [10, 11], and verticals [5,6]. Common to these tasks is that objects do not have direct representations that could be matched against the search query. Instead, they are associated with documents, which are used as a proxy to connect objects and queries. See Fig. 1 for an illustration. The main question, then, is how to combine evidence from documents that are associated with a given object.

Most approaches that have been proposed for object retrieval can be categorized into two main groups of retrieval strategies: (1) *object-centric* methods build a term-based representation of objects by aggregating term counts across the set of documents associated with the objects; (2) *document-centric* methods first retrieve documents relevant to the query, then consider the objects associated with these documents. Viewed abstractly, the object retrieval task is about fusing or blending information about a given object. This fusion may happen early on in the retrieval process, on the term level (i.e., object-centric methods), or later, on the document level (i.e., document-centric methods). Using either of the two strategies, two main shared components can be distilled:
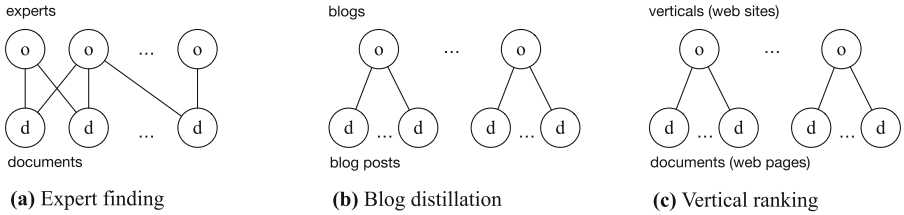
Fig. 1. Illustration of various object retrieval tasks.

the underlying term-based retrieval model (e.g., language models, BM25, DFR, etc.) and the document-object association method. Various instantiations (i.e., choice of retrieval strategy, retrieval model, and document-object associations) have been studied, but always in the context of a particular object retrieval task, see, e.g., [2,7,9,13].

    We show in this paper, as our main contribution, that further generalizations are possible. We present two *design patterns* for object retrieval, that is, general repeatable solutions that can easily emulate most previously proposed approaches. We call these design patterns to emphasize that they can be used in many different situations. The second contribution of this work is an experimental evaluation performed for three different object retrieval tasks: expert finding, blog distillation, and vertical ranking. Using standard TREC collections, we demonstrate that the early and late fusion patterns are indeed widely applicable and deliver competitive performance without resorting to any task-specific tailoring. The implementation of our models is available at http://bit.ly/ecir2017-fusion.

2   Fusion-Based Object Retrieval Methods

Object retrieval is the task of returning a ranked list of objects in response to a keyword query. We assume a scenario where objects do not have direct term-based representations, but each object is associated with one or more documents. These documents are used as a bridge between queries and objects. We present two design patterns, i.e., general retrieval strategies, in the following two subsections. Both strategies consider the relationship between a document and an object; we detail this element in Sect. 2.3

2.1   Early Fusion

According to the early fusion (or object-centric) strategy a term-based representation is created for each object. That is, the fusion happens on the term level. One can think of this approach as creating a pseudo document for each object; once those object description documents are created, they can be ranked using standard document retrieval models. We define the (pseudo) frequency of a term t for an object o as follows:

$$\tilde{f}(t, o) = \sum_d f(t, d)w(d, o), \tag{1}$$

where $f(t, d)$ is the frequency of the term in document d and $w(d, o)$ denotes the document-object association weight. The relevance score of an object for a given query q is then calculated by summing the individual scores of the individual query terms:

$$score(o, q) = \sum_{i=1}^{|q|} score(q_i, o) = \sum_{i=1}^{|q|} score(q_i, \tilde{f}, \varphi),$$

where φ holds all parameters of the underlying retrieval model (e.g., k_1 and b for BM25). For computing $score(t, o)$, any existing retrieval model can be used. Specifically, using language models with Jelinek-Mercer smoothing it is:

$$score_{LM}(t, o) = \log \left((1 - \lambda)\frac{\tilde{f}(t, o)}{|o|} + \lambda P(t)\right),$$

where $|o|$ is the length of the object ($|o| = \sum_t \tilde{f}(t, o)$), $P(t)$ is the background language model, and λ is the smoothing parameter. Using BM25, the term score is computed as:

$$score_{BM25}(t, o) = \frac{\tilde{f}(t, o)(k_1 + 1)}{\tilde{f}(t, o) + k_1(1 - b + b\frac{|o|}{\text{avg}(o)})}IDF(t),$$

where $IDF(t)$ is computed as $\log \frac{N}{|\{o:\tilde{f}(t,o)>0\}|}$ and $\text{avg}(o)$ is the average object length.

Table 1 lists exiting approaches for different search tasks, which can be classified as early fusion. Due to space constraints, we only highlight one specific method for each of the object ranking tasks we consider.

Table 1. Examples of early fusion approaches. Notice that the aggregation happens on the term level. (Computing the log probabilities turns the product into a summation over query terms.)

Task	Model	Equation
Expert finding	Profile-based [8]	$P(q|\theta_c, R = 1) = \prod_{t \in q} p(t|\theta_c, R = 1)^{n(t,q)}$
Blog distillation	Blogger model [3]	$P(q|\theta_{blog}) = \prod_{t \in q} P(t|\theta_{blog})^{n(t,q)}$
Vertical ranking	CVV [12]	$Goodness(c, q) = \sum_{i=1}^{|q|} CVV_i \times df_{i,c}$

Table 2. Examples of late fusion approaches. Notice that aggregation happens on the document level; each formula contains a term that expresses the document's relevance.

Task	Model	Equation
Expert finding	Voting model [9]	$score\_cand\_RR(e,q) = \sum_{d \epsilon R(q) \cap profile(e)} \frac{1}{rank(d,q)}$
Blog distillation	Posting model [3]	$P(q|blog) = \sum_{post \epsilon blog} P(q|\theta_{post})P(post|blog)$
Vertical ranking	ReDDE [12]	$R(c,q) = \sum_{d \epsilon c} P(R|d)P(d|c)|c|$

2.2 Late Fusion

Instead of creating a direct term-based representation for objects, the late fusion (or document-centric) strategy models and queries individual documents, then aggregates their relevance estimates. Formally:

$$score(o,q) = \sum_d score(d,q)w(d,o), \qquad (2)$$

where $score(d,q)$ expresses the document's relevance to the query and can be computed using any existing document retrieval method, such as language models or BM25. As before, $w(d,o)$ is the weight of document d for the given object. The efficiency of this approach can be further improved by restricting the summation to the top-K relevant documents. Table 2 shows three exiting models for different search tasks, which can be catalogued as late fusion strategies.

2.3 Document-Object Associations

Using either the early or the late fusion strategy, they share the component $w(d,o)$, cf. Eqs. (1) and (2). This document-object association score determines the weight with which a particular document contributes to the relevance score of a given object. In this paper, we consider two simple ways for setting this weight. We introduce the shorthand notation $d \in o$ to indicate that document d is associated with object o (i.e., there is an edge between d and o in Fig. 1). According to the *binary* method, $w(d,o)$ can take only two values: it is 1 if $d \in o$ and 0 otherwise. Alternatively, the *uniform* method assigns the value $\frac{1}{len(o)}$ if $d \in o$, where $len(o)$ is the total number of documents associated with o, and 0 otherwise.

3 Experimental Setup

We consider three object retrieval tasks, with corresponding TREC collections. *Expert finding* uses the test suites of the TREC 2007 and 2008 Enterprise track [1,4]. Objects are experts and each of them is typically associated with multiple documents. *Blog distillation* is based on the TREC 2007 and 2008 Blog track [10,11]. Objects are blogs and documents are posts; each document (post)

Table 3. Object retrieval tasks and collections used in this paper.

Task	Collection (#docs)	Queries
Expert finding	CSIRO (370K)	50 (2007), 77 (2008)
Blog distillation	Blogs06 (3.2M)	50 (2007), 50 (2008)
Vertical ranking	FedWeb13 (1.9M), FedWeb14 (3.6M)	50 (2013), 50 (2014)

belongs to exactly on object (blog). *Vertical ranking* corresponds to the resource selection task of the TREC 2013 and 2014 Federated Search track [5,6]. Objects are verticals (i.e., web sites) and documents are web pages. Table 3 summarizes the data sets used for each task.

For each task, we consider two retrieval models: language models (using Jelinek Mercer Smoothing, $\lambda = 0.1$) and BM25 (with $k_1 = 1.2$ and $b = 0.75$). We further compare two models of document-object associations: binary and uniform.

4   Experimental Results

The results for the expert finding, blog distillation, and vertical ranking tasks are presented in Tables 4, 5, and 6, respectively. Our main observations are the following. First, there is no preferred fusion strategy; early and late fusion both emerge as overall bests in 3–3 cases. While early fusion is clearly preferred for vertical ranking and late fusion is clearly favorable for blog distillation, a mixed picture unfolds for expert finding: early fusion performs better on one query set (2007) while late fusion wins on another (2008). The differences between the corresponding early and late fusion configurations can be substantial. Second, the main difference between binary and uniform associations is that the latter takes into account the number of different documents associated with the object, while the former does not. For expert finding and vertical ranking the binary method is clearly superior. For blog distillation, on the other hand, it is nearly always the uniform method that performs better. The difference between vertical ranking and blog distillation is especially interesting given that these two tasks have essentially identical structure, i.e., each document is associated with exactly one object (see Fig. 1). Third, concerning the choice of retrieval model (LM vs. BM25), we again find that it depends on the task and fusion strategy. BM25 is superior to LM on blog distillation. For expert finding and vertical ranking, LM performs better in case of early fusion, while BM25 is preferable for late fusion.

We also include the TREC best and median results for reference comparison. In most cases, our fusion-based methods perform better than the TREC median, and on one occasion (vertical ranking, 2013) we outperform the best TREC run. Let us emphasize that we did not resort to any task-specific treatment. In the light of this, our results can be considered more than satisfactory and signify the generality of our fusion strategies.

Table 4. Results on the expert finding task. Highest scores are in boldface.

Fusion strategy	Retr. model	Doc-obj. assoc.	2007			2008		
			MAP	MRR	P@10	MAP	MRR	P@10
Early fusion	LM	binary	**0.3607**	**0.4809**	0.1229	0.1927	0.3741	0.1863
	LM	uniform	0.2902	0.3650	0.1083	0.1760	0.3843	0.1725
	BM25	binary	0.2887	0.3654	0.0900	0.1203	0.2599	0.1148
	BM25	uniform	0.1688	0.2159	0.0780	0.0646	0.1517	0.0741
Late fusion	LM	binary	0.3283	0.4730	0.1420	0.2036	0.4342	0.2167
	LM	uniform	0.1978	0.2561	0.0940	0.1146	0.2948	0.1296
	BM25	binary	0.3495	0.4949	**0.1480**	**0.2623**	**0.5048**	**0.2648**
	BM25	uniform	0.2492	0.3065	0.1040	0.1787	0.3988	0.1759
TREC best			**0.4632**			**0.2987**	0.4951	
TREC median			0.3090			0.2606	0.3843	

Table 5. Results on the blog distillation task. Highest scores are in boldface.

Fusion strategy	Retr. model	Doc-obj. assoc.	2007			2008		
			MAP	MRR	P@10	MAP	MRR	P@10
Early fusion	LM	binary	0.2055	0.4660	0.3432	0.1883	0.6996	0.3684
	LM	uniform	0.2479	0.5313	0.3932	0.1897	0.6228	**0.3740**
	BM25	binary	0.2374	0.4773	0.3844	0.1789	0.5731	0.3460
	BM25	uniform	0.2088	0.6316	0.3578	0.1936	0.6180	0.3460
Late fusion	LM	binary	0.1845	0.5349	0.3111	0.1556	0.4755	0.2800
	LM	uniform	0.2605	0.6140	0.4222	0.2040	0.7241	0.3360
	BM25	binary	0.2202	0.5892	0.3489	0.1731	0.5478	0.3140
	BM25	uniform	**0.2987**	**0.7303**	**0.4822**	**0.2245**	**0.7482**	0.3600
TREC best			**0.3695**	**0.8093**	**0.5356**	**0.3015**	**0.8051**	**0.4480**
TREC median			0.2353	0.7425	0.4567	0.2416	0.7167	0.3580

Table 6. Results on the vertical ranking task. Highest scores are in boldface.

Fusion strategy	Retr. model	Doc-obj. assoc.	2013			2014		
			nDCG@20	MAP	P@5	nDCG@20	MAP	P@5
Early fusion	LM	binary	**0.3382**	**0.3656**	**0.4000**	**0.2782**	**0.3052**	**0.4857**
	LM	uniform	0.2271	0.2293	0.3306	0.2184	0.2612	0.3633
	BM25	binary	0.2588	0.2704	0.2500	0.2354	0.2758	0.3920
	BM25	uniform	0.1689	0.1960	0.2612	0.1669	0.2204	0.2960
Late fusion	LM	binary	0.1950	0.1991	0.2163	0.1961	0.2439	0.3000
	LM	uniform	0.1370	0.1641	0.1755	0.1408	0.2094	0.2400
	BM25	binary	0.2373	0.2163	0.2490	0.2220	0.2576	0.3400
	BM25	uniform	0.1548	0.1755	0.1918	0.1658	0.2208	0.3000
TREC best			0.2990		0.3200	**0.7120**		**0.6040**
TREC median			0.1410		0.1850	0.3450		0.2125

5   Conclusions

In this paper we have presented two design patterns, early and late fusion, to the commonly occurring problem of object retrieval. We have demonstrated the generality and reusability of these solutions on three different tasks: expert finding, blog distillation, and vertical ranking. Specifically, we have considered various instantiations of these patterns using (i) language models and BM25 as the underlying retrieval model and (ii) binary and uniform document-object associations. We have found that these strategies are indeed robust and deliver competitive performance using default parameter settings and without resorting to any task-specific treatment. We have also observed that there is no single best configuration; it depends on the task and sometimes even on the particular test query set used for the task. One interesting question for future work, therefore, is how to automatically determine the configuration that should be used for a given task.

References

1. Bailey, P., Craswell, N., de Vries, A.P., Soboroff, I.: Overview of the TREC: enterprise track. In: Proceedings of TREC 2007 (2008)
2. Balog, K., Azzopardi, L., de Rijke, M.: Formal models for expert finding in enterprise corpora. In: Proceedings of SIGIR, pp. 43–50 (2006)
3. Balog, K., de Rijke, M., Weerkamp, W.: Bloggers as experts: feed distillation using expert retrieval models. In: Proceedings of SIGIR, pp. 753–754 (2008)
4. Balog, K., Soboroff, I., Thomas, P., Craswell, N., de Vries, A.P., Bailey, P.: Overview of the TREC: enterprise track. In: Proceedings of TREC 2008 (2009)
5. Demeester, T., Trieschnigg, D., Nguyen, D., Hiemstra, D.: Overview of the TREC: federated web search track. In: Proceedings of TREC 2013 (2014)
6. Demeester, T., Trieschnigg, D., Nguyen, D., Hiemstra, D., Zhou, K.: Overview of the TREC: federated web search track. In: Proceedings of TREC 2014 (2015)
7. Elsas, J.L., Arguello, J., Callan, J., Carbonell, J.G.: Retrieval and feedback models for blog feed search. In: Proceedigs of SIGIR, pp. 347–354 (2008)
8. Fang, H., Zhai, C.X.: Probabilistic models for expert finding. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 418–430. Springer, Heidelberg (2007). doi:10.1007/978-3-540-71496-5_38
9. Macdonald, C., Ounis, I.: Voting techniques for expert search. Knowl. Inf. Syst. **16**, 259–280 (2008)
10. Macdonald, C., Ounis, I., Soboroff, I.: Overview of the TREC: blog track. In: Proceedings of TREC 2007 (2008)
11. Ounis, I., Macdonald, C., Soboroff, I.: Overview of the TREC-2008 blog track. In: Proceedings of TREC 2008 (2009)
12. Shokouhi, M., Si, L.: Federated search. Found. Trends Inf. Retr. **5**, 1–102 (2011)
13. Weerkamp, W., Balog, K., de Rijke, M.: Blog feed search with a post index. Inf. Retr. **14**, 515–545 (2011)