Expert Systems With Applications 176 (2021) 114909

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Query expansion based on term distribution and DBpedia features

Sarah Dahir[a,*], Abderrahim El Qadi[b], Hamid Bennis[a]

[a] IMAGE Laboratory, SCIAM Team, Graduate School of Technology, Moulay Ismail University of Meknes, Morocco
[b] ENSAM, Mohammed V University in Rabat, Morocco

ARTICLE INFO

Keywords:
Information retrieval
Query expansion
DBpedia spotlight
Term distribution
Language model

ABSTRACT

Query Expansion (QE) approaches that involve the reformulation of queries by adding new terms to the initial user query, are intended to ameliorate the vocabulary mismatch between the query keywords and the documents' in Information Retrieval Systems (IRS). One big issue in QE is the selection of the right candidate terms for expansion. For this purpose Linked Data can be used, as a valuable resource, for providing additional expansion features such as the values of sub- and super classes of resources. The underlying research question is whether interlinked data and vocabulary items provide features which can be taken into account for query expansion. In this paper, we introduced a new QE approach that aimed at improving IRS by using the well-known distribution based method Bose-Einstein statistics (Bo1) as well as Linked Data from the knowledge base DBpedia using different numbers of expansion terms. We evaluated the effectiveness of each method individually as well as their combinations using two Text REtrieval Conference (TREC) test collections. Our approach has lead to significant improvement in terms of precision, recall, Mean Average Precision (MAP) at rank 10, and normalized Discounted Cumulative Gain (nDCG) at different ranks compared to Pseudo Relevance Feedback (PRF) that we used as a baseline. The results show that the inclusion of semantic annotations clearly improves the retrieval performance over the baseline method.

1. Introduction

Information Retrieval (IR) aims at returning relevant documents to the user's query. Yet, this domain of study suffers from many challenging problems. Some of the problems correspond to evaluation issues such as: the difficulty and impossibility of building larger test collections along with complete relevance judgment. In fact, it requires assessor time and many diverse retrieval runs (Goharian, 2020) to evaluate an Information Retrieval System. Also, there is the problem of the determination of adequate dataset to use for evaluation as well as the adequate number of queries to test and the evaluation measures to use. Moreover, it is difficult to know the number of query runs to opt for. Other IR problems are related to users' queries like: vocabulary gap between query keywords and terms used by indexers. The vocabulary gap prevents relevant documents from being retrieved and results in low recall values. And there is the problem of the small average length of a query which is 2.4 terms (Spink et al., 2001); especially that 4% of Web queries and 16% of the most frequently used queries are ambiguous (Di Marco & Navigli, 2013).

Linked Data are becoming more and more widely used and exploited for different purposes using from the one hand, Resource Description Framework (RDF) that is a data format for describing things as well as their interrelations. And using from the other hand, Uniform Resource Identifier (URI) through typed links. Those characteristics of linked data, allow the measurement of entities' semantic relatedness and semantic similarity (Ruback et al., 2017, 2018). Consequently, interlinked data can help improving recommendation systems by suggesting for example similar activities to users based on the activities they tend to choose or attend, etc.

QE methods improve the retrieval effectiveness, by adding relevant terms to the initial user query. There are several query expansion methods. A well-known and effective technique, in ad-hoc IR, to address the vocabulary mismatch problem is PRF (Rocchio, 1971) (Carpineto & Romano, 2012). It represents a small set of top-retrieved documents which are relevant to the query. Therefore, in this set of documents, a number of relevant terms can be selected, and added to the original query.

This article presents a novel query expansion approach that benefits from a semantic external source, called DBpedia (Dahir et al., Personal communication, 2018a), to further expand relevant feedback terms

[*] Corresponding author.
E-mail addresses: sarah.dahir2012@gmail.com (S. Dahir), abderrahim.elqadi@um5.ac.ma (A. El Qadi).

https://doi.org/10.1016/j.eswa.2021.114909
Received 28 January 2020; Received in revised form 5 August 2020; Accepted 9 March 2021
Available online 16 March 2021
0957-4174/© 2021 Elsevier Ltd. All rights reserved.

obtained using the distribution based technique Bo1 (Amati, 2003). In other words, we want to check if the combination of a local target-based approach (Bo1) and a global semantic approach (DBpedia) would positively affect query expansion. Especially that, on the one hand, Bo1 ranks top feedback documents' terms based on the whole corpus. Consequently, it is more advantageous than PRF. And on the other hand, DBpedia extracts linked data from Wikipedia (i.e. it is large enough, domain independent, and often updated) and allows Named Entity Recognition through annotation. One more novelty of our work is checking how low and high numbers of expansion terms affect the retrieval result.

Through this work, we are looking forward to determine the ideal number of expansion terms to add to the initial query by comparing different numbers of expansion terms. Moreover, we would like to check whether the increasement of expansion terms leads to higher results. Also, we would like to verify if the Bo1 improves the PRF results and if the use of DBpedia features to further expand the Bo1 expanded queries will give better retrieval results.

The remainder of this paper is organized as follows. Section 2 provides related work. Section 3 accounts for our proposed approach. Section 4 addresses their evaluation, and discusses the achieved results. Finally, section 5 concludes the work.

2. Related work

Over the last years, Automatic Query Expansion techniques have been widely applied for improving text retrieval performance. And different approaches have been proposed for selecting expansion terms (Carpineto & Romano, 2012). Traditional expansion methods can be roughly divided into QE based on local analysis and QE through global analysis:

(1) Local methods which are target corpus based (Pal et al., 2013). Such methods can be done through: Relevance Feedback (RF) (Salton & McGill, 1983) in which it is up to the user to mark as relevant or irrelevant a list of answers to his or her query. Or expansion through PRF which is a variation of RF, also known as blind RF, that automates the manual part of RF by assuming that the top *k* documents are relevant (Buckley et al., 1995; Manning et al., 2008). Also, through term distribution methods which allow the determination of candidate expansion terms that are more likely to occur in a highly ranked feedback document than to occur in a randomly selected document. And term association methods that consider a term that tends to co-occur with all or many of the query terms as a good expansion term (Pal et al., 2013).

In (Keikha et al., 2018) authors exploit Wikipedia feedback articles of query n-grams to expand their queries. In (Pal et al., 2013), authors suggest improving retrieval results by modifying distribution based and association based methods then combining them. In (Dahir et al., Personal communication, 2018a) a query's DBpedia entity is expanded based on the similarity between its features' vector and the features' vector of the indexed terms from feedback documents. And in (Xu et al., 2020), authors rely on the social annotations of a user for labeling resources within the feedback documents. The socially extended feedback documents are then used in topic models to expand queries.

(2) Global methods that include query expansion using: thesauri that do not label the semantic relations between terms (Jain et al., 2014). Or methods using ontology like the lexical database WordNet that groups nouns, verbs, etc. into sets of synonyms called "synsets". Or even using Linked Open Data that contrary to the classic Web which uses documents, the Web of data exploits resources identified by URIs (Abbes et al., 2013). As for DBpedia, the URI of a certain entity has the following form: http://dbpedia.com/resource/entity_name. Yet, the huge number of available resources' utility (e.g. type, subject, etc.) and ontology attributes that are most of the time multi-valued (Abbes et al., 2013); makes it difficult to select the right attribute to use. But in our work we overcame this problem by using an attribute that usually has only one corresponding value.

In (Mendes et al., 2011), authors suggest using DBpedia Spotlight to annotate DBpedia's mentioned resources in a text through: First, "spotting" to determine, in a phrase, the expressions that indicate a DBpedia's resource. Second, "candidates' selection" that maps the previously determined resources to candidate disambiguation. Third, "disambiguation": the candidate resources are ranked depending on the similarity score between their context vectors and the context surrounding the annotated resource using cosine similarity. And fourth, "configuration" of some resources' annotation parameters such as: annotating only resources of a certain type/period/place. Authors also define the "resource prominence" that allows specifying the minimum number of "inlinks", which a resource should have in order to be annotated. And they define the "disambiguation confidence" that can be set to a high value to avoid incorrect annotations.

In (Jain et al., 2014) authors use a graph based method to expand queries. They select all synonyms, hypernyms, etc. of each query term to obtain graph *G*. And they create a sub-graph *G'* that include only the shortest path between a pair of query terms. Then, they calculate graph connectivity measures between pairs of terms that have a less or an equal length to a calculated minimum distance between query terms from graph *G'*. A term is then chosen for expansion based on whether it has at least three measures higher than the average calculated one.

In (Dahir et al., Personal communication, 2018b), queries are expanded by using SPARQL Protocol and RDF Query Language (SPARQL) to interrogate DBpedia and select all the resources with a certain feature (value). Then, cosine similarity measure is used to compare the found lists of resources and expand the queries.

In (Shekarpour et al., 2013), authors from the one hand select synonyms, hyponyms, and hypernyms of each query keyword *k*. And from the other hand, they match every *k* against the "rdfs:label" of all Linked Open Data (LOD) resources in order to extract classes, instances, and properties whose labels contain *k* or are equal to it. Then, they derive all semantically related resources to the previously extracted ones (r_i) using some semantic relations like "sameAs" and "subclass" depending on the type of r_i. For instance, "subclass" is applicable just for recourses of type: "class". After this, they do tokenization, lemmatization and stop-words' removal before doing feature weighting to keep only relevant features for expansion.

In (Augenstein et al., 2013), authors determine dataset triples that use a query keyword as the label of a resource. Then, they select the neighbors of this resource, i.e. the objects of any relationship from the subject resource, as expansion candidates.

In (Zong et al., 2015), authors first use RDF documents published using the N-quads (subject, predicate, object, and context) format. Second, the page rank of these documents is calculated. Third, an RDF document is represented by a vector of terms, and the weight of a term in the vector is calculated using Term Frequency-Inverse RDF Frequency. Similarly a query is transformed to a vector in order to compute its similarity with the RDF document's vector. And fourth, all entities contained in the top RDF documents are extracted and the entity that contains any query term is considered as an "anchor entity". An entity is expansion candidate if it has a link to the "anchor entity".

In (Balaneshinkordan & Kotov, 2016) authors exploit the corresponding DBpedia feature "dbo:abstract" of entities within the query. Then, they create term association graphs using the values of that feature. In order to construct the graphs; they calculated a co-occurrence based information theoretic measure called Mutual Information (MI) which uses the whole document to determine the strength of association between a pair of terms; by counting the occurrence number of each term individually and their joint occurrence's number.

And in (Todor et al., 2016), authors downsize text documents to keep only entities and expand those entities using DBpedia features. After that, they use the enriched entities in topic models for query expansion.

Table 1 analyzes our work with respect to the existing approaches and highlights the differences.

3. Proposed approach

In order to improve IR results, we suggested a query expansion approach that not only expanded the initial user query (q_I) using both relevant feedback terms and their DBpedia features, but also allowed testing the approach using 8 different numbers of expansion terms; ranging from $k = 1$ to $k = 100$. The proposed process, depicted in the Figs. 1 and 2, is divided into two main steps:

Step 1: Distribution QE method (exp_Bo1) (Fig. 1)

1. Documents retrieval using KL language model: Assignment of a relevance score to a document based on whether this document may be generated by the initial query's model or not using the Kullback-Leibler (KL) Language Model (LM).

In fact, language modeling approaches have already shown great promise for multiple retrieval tasks with very good empirical performance (El Ghali & El Qadi, 2017; Zhai, 2008; Lafferty & Zhai, 2001). They consist on considering that a document represents a sub-language, for which we try to construct the LM. In fact, this standard approach is the main idea of the query-likelihood (1); where the score of a document, given a query, can be determined by the probability that the document's model generates the query.

$$\mathrm{P}(\mathrm{q}_1 \cdots \mathrm{q}_k | \mathrm{M}_\mathrm{D}) = \prod_{i=1}^{k} \mathrm{P}(\mathrm{q}_i | \mathrm{M}_\mathrm{D}) \tag{1}$$

Or, we create a LM for the query and give a score to a document based on whether this document may be generated by the query's model which is called document-likelihood.

$$\mathrm{P}(\mathrm{D} | \mathrm{M}_\mathrm{Q}) = \prod_{t \in \mathrm{D}} \mathrm{P}(\mathrm{t} | \mathrm{M}_\mathrm{Q}) \tag{2}$$

Another way to score a document is model comparison which is the basic idea of KL-divergence: it consists on comparing the document's model with the query's model.

$$\mathrm{D}_{\mathrm{KL}}(\mathrm{P}||\mathrm{Q}) = \sum_{x \in \mathrm{X}} \mathrm{P}(x) \log(\frac{\mathrm{P}(x)}{\mathrm{Q}(x)}) \tag{3}$$

Where P and Q are discrete probability distributions defined on the same probability space.

Moreover, smoothing through Dirichlet (4) (used in this work) for example may be used to avoid getting a null result when a term is absent in the constructed Language Model.

$$\mathrm{P}_{\mathrm{Dir}}(\mathrm{t}_i | \mathrm{D}) = \frac{\mathrm{tf}(\mathrm{t}_i, \mathrm{D}) + \mu \mathrm{P}_{\mathrm{ML}}(\mathrm{t}_i | \mathrm{C})}{|\mathrm{D}| + \mu} \tag{4}$$

Where the frequency of a term $\mathrm{t_i}$ in document D is incremented with $\mu \mathrm{P}_{\mathrm{ML}}(\mathrm{t}_i|\mathrm{C})$, μ is a pseudo frequency parameter, $|D|$ is the number of words' occurrences in the document, and $tf(t_i, D)$ is the frequency of the term t_i in document D.

2. Expansion of q_I using Bo1: all words closely related to the original query keywords were extracted from the top n ranked documents, using the Distribution Method Bo1 (Pal et al., 2013). In fact, Bo1 consists on comparing the distribution of a candidate indexed term in the top feedback documents with its distribution in the whole corpus as shown in (5) (Pal et al., 2013).

Table 1
Comparison between related work and our approach in terms of significant improvements.

Related work		Significant improvements of our approach
Reference	Limits	
Keikha et al., 2018	- A feedback document can be dependent on the others. - Testing only low numbers of expansion terms; with 35 being the highest number.	- Each feedback document is independent from the others. - Testing both low and very high numbers of expansion terms; with 100 being the highest number.
Pal et al., 2013	- Not trying different numbers of expansion terms. - Not using external sources.	- Trying different numbers of expansion terms. - Improving the results of Bo1 distribution method by using linked data.
Dahir et al., Personal communication, 2018a	Relying on the top feedback documents that may be irrelevant.	- Using term distribution that considers the whole corpus when giving a score to a term from a top feedback document.
Xu et al., 2020	- Depending on the availability of user annotations and their quality (e.g. spelling errors, use of expressions that are different from those used by domain experts because of the user's lack to knowledge about the domain of his/her search).	- Using "rdfs:label" that gives a sequence of terms that is most likely to be used by domain experts and is thus important for retrieval of relevant documents.
Mendes et al., 2011	- DBpedia Spotlight does not annotate term sequences that are misspelled. Also, it is case sensitive i.e. it annotates differently lower and upper case term sequences. If an acronym is written in lower case and/or its letters are separated by dots, DBpedia Spotlight does not annotate it. And it fails to annotate very long texts.	- To make sure that all entities will be annotated by DBpedia Spotlight; we tried annotation using lower case, upper case and upper case for the first letter from a term. - We used the default value of "disambiguation confidence" (0.5).
Jain et al., 2014	Using WordNet that may not have ontology for a certain domain. Moreover, WordNet has a low coverage of concepts and phrases (Sinha & Mihalcea, 2007)	Using DBpedia which annotates term sequence entities.
Dahir et al., Personal communication, 2018b	Using an association based approach.	Using a distribution based approach.
Shekarpour et al., 2013	- Each keyword from the query is processed separately from the others which means that the context of the query is not taken into consideration. - Leaving preprocessing methods to the end.	- Taking the context into consideration by using DBpedia Spotlight for annotation. - Preprocessing is used in the beginning.
Augenstein et al., 2013	Mapping query keywords to labels from linked data requires queries that are long enough which is usually not the case.	Using labels from linked data on expanded Bo1 queries and not on initial queries.
Zong et al., 2015	Using "anchor entity" to determine candidate entities for expansion is not always efficient. For instance, the "anchor	Using annotation through DBpedia Spotlight.

(continued on next page)

Table 1 (*continued*)

Related work		Significant improvements of our approach
Reference	Limits	
	entity" may contain a query term but does not have any relation with the intention of the user.	
Balaneshinkordan & Kotov, 2016	Calculating semantic term relatedness based on co-occurrence solely; requires large data. Yet, "dbo: abstract" has often a short to medium value.	- Using distribution instead of association since distribution approaches tend to outperform association approaches (Pal et al., 2013). - Using "rdfs:label" that is often mono-valued, just like "dbo:abstract", but concise and gives information that can solve the vocabulary mismatch problem. In fact, the feature that we used gives a term sequence that is usually the most used by indexers and experts of the domain.
Todor et al., 2016	- Mining topic models using ontology concepts; requires identifying the right features to use, because of the enormous loss of context from documents after reducing their size. - Using a very small number of features. - Using either features that are not always available even if they are domain independent (e.g. hypernyms) or features that can be multi-valued e.g. category. These problems may lead to non accurate results. - Testing as many features' combinations as possible is very demanding computationally, because DBpedia is extremely rich in terms of features.	- Instead of keeping all entities within documents, we keep them based on their term distribution score. - Choosing "rdfs:label" in particular because it is always available for all resources and it is almost always mono-valued.

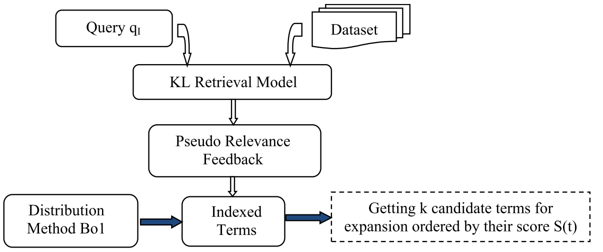
Query q_I
Dataset
KL Retrieval Model
Pseudo Relevance Feedback
Distribution Method Bo1
Indexed Terms
Getting k candidate terms for expansion ordered by their score S(t)

Fig. 1. Distribution query expansion method (exp_Bo1).

$$S(t) = \left(\sum_{d\in RD} tf(t,d)\right)^{*}\log_2\left(\frac{1+f_{avg}(t,C)}{f_{avg}(t,C)}\right) + \log_2(1+f_{avg}(t,C)) \tag{5}$$

Where: *RD* and *C* (in this work *C* = 200) represent the relevant documents and the whole corpus respectively, and *tf*(*t*,*d*) represents the term frequency of a candidate term *t* in document *d*,

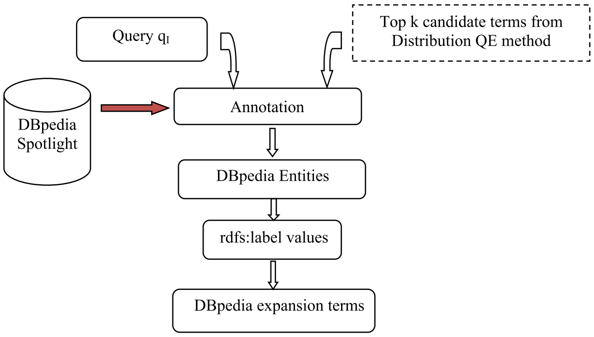

Fig. 2. Flowchart of our DBpedia-Distribution Expansion Approach (exp_Bo1 + DBpedia).

$$f_{avg}(t,C) = \sum_{d\in C}\frac{tf(t,d)}{N} \tag{6}$$

$f_{avg}(t,C)$ is the average term frequency of *t* in the whole corpus, and *N* is the number of documents in the corpus.

The obtained *S(t)* of each indexed term was considered as the weight of the term, and *k* candidate terms with highest scores: $k = 1, 3, 5, 7, 10, 20, 50, 100$ were used to obtain the expanded queries;

Step 2: DBpedia-Distribution Expansion Approach (exp_Bo1 + DBpedia) (Fig. 2)

3. Annotation of the previously expanded query using, the web platform, *DBpedia Spotlight*[1]: The DBpedia mentions (i.e. entities) in the Bo1 expanded query are recognized by DBpedia and transformed into links, allowing us to find all their associated attributes including "rdfs:label" values,

Further expansion of the previously expanded query using the found DBpedia labels (7):

$$q_{exp} = q_I + q_{I_{Labels}} + Bo1_{Labels} \tag{7}$$

q_I: the initial query,

$q_{I_{Labels}}$: the DBpedia labels of DBpedia entities within the initial query, and $Bo1_{Labels}$: the DBpedia labels of the determined expansion indexed terms after using Bo1 method.

The reasons why we opted for "rdfs:label" was that from the one hand, contrary to so many other attributes this one usually had only one value which means we would not find difficulties in determining the right value to use, and from the other hand this attribute was available for any resource because it was not an ontology attribute but an utility

Table 2
Examples of initial queries, top Bo1 words, and their semantic expansion words.

Initial Query	Highest 15 distribution expansion words	Semantic expansion words
controlling type ii diabetes	texas, recruiting, texans, michelle, tdh, al, workgroup, med, metformin, quinn, diabetes, restriction, fiber, carbohydrate, caloric	diabetes mellitus type 2, diabetes mellitus, dietary fiber
antitrust cases pending	justice, department, lackadaisical, defense, procurement, fraud, gao, criminal, dairy, milk, rule, farmers, fines, nfo, abbott	competition law, united states department of justice, cadbury dairy milk, abbott laboratories

[1] https://www.dbpedia-spotlight.org/demo/

one.

Table 2 shows an example of the application of our query expansion method on two queries: “controlling type ii diabetes” from GOV2 collection and “antitrust cases pending” from AP collection. We first run the query using KL Retrieval Model to determine the top *n* feedback documents ($n = 10$). Then, we applied Bo1 distribution method using the indexed terms of pre-determined *n* documents. After that, we used *k* indexed terms that had the highest Bo1 scores as expansion terms to obtain our exp_Bo1. Then using DBpedia Spotlight, we added terms and expressions that corresponded to the DBpedia attribute “rdfs:label” of the DBpedia mentions in the expanded query to obtain our semantic expansion words.

The expansion expression “diabetes mellitus type 2”, in the table above, is the DBpedia label of the initial query’s DBpedia resource “type ii diabetes”, “diabetes mellitus” is the label of the expansion term “diabetes” from the exp_Bo1, and “dietary fiber” is a label that corresponds to the expansion term “fiber” from the exp_Bo1.

The practical evaluation outcomes of this approach are:

- Identifying the most accurate number of feedback terms, with the highest distribution scores, to use for expansion;
- Providing semantic term sequences from linked data that are more used by domain experts to refer to relevant entities from top feedback documents and the initial query. Consequently, our approach increases the chances of retrieving documents that do not contain any term from the user query but are relevant to it. And this fulfillment is one of the main objectives of query expansion.

4. Experiments and results

To evaluate the proposed methods, we used the search engine *Indri*[2], and two standard TREC collections: GOV2[3] and AP88-90[4]. The statistics of these collections are reported in Table 3. The standard stop words list was used, and no stemming was performed. We opted for KL as a Retrieval Model with a smoothing parameter$\mu = 1000$ (Dahir et al., Personal communication, 2018b).

1) Evaluation measures

In this work we used four evaluation measures:

Precision. Shows to which level a system is capable of returning only relevant documents (Zuva & Zuva, 2012):

$$\text{Precision} = \frac{\text{Numberofrelevantretrieveddocuments}}{\text{Numberofretrieveddocuments}} \tag{8}$$

Table 3
Test collections.

Dataset		Value
GOV2	Number of documents	25,205,179
	Average document size	17.7 kb
	Document relevancy	0 (non-relevant), 1 (relevant), 2 (highly relevant)
	Topics (queries) numbers	701–775 and 776–850
AP	Number of documents	158,240
	Average document size	261
	Document relevancy	0 (non-relevant), 1 (relevant)
	Topics (queries) numbers	1–50, 51–100, and 101–150

Recall. Is also called the true positive rate and it shows how capable a system is of returning all relevant documents (Zuva & Zuva, 2012):

$$\text{Recall} = \frac{\text{Numberofrelevantretrieveddocuments}}{\text{Numberofrelevantdocuments}} \tag{9}$$

MAP. The MAP for or a set of queries is the mean of the Average Precision (AP) scores for each query (Wikipedia contributors, 2019, December 31).

$$\text{MAP} = \frac{\sum_{q=1}^{Q}\text{AveP(q)}}{Q} \tag{10}$$

Where Q is the number of queries.

$$\text{AveP} = \frac{\sum_{k=1}^{n}(P(k) \times \text{rel}(k))}{\text{Numberofrelevantdocuments}} \tag{11}$$

Where *k* is the rank in the sequence of retrieved documents, *n* is the number of retrieved documents, *P(k)* is the precision at cut-off *k* in the list, and *rel(k)* is an indicator function equaling 1 if the item at rank *k* is a relevant document, zero otherwise.

NDCG. Uses the Discounted Cumulative Gain (DCG) which measures gain or usefulness by considering the top ranked retrieved documents as well as the position of documents in the result set. For this measure; the lower the position of a relevant document is, the less useful it is for the user. And the higher the relevance of a document is (e.g. relevance 2 is better than 1), the better the document is (compared to a marginally relevant one) (Goharian, 2020).

$$\text{DCG}_{P=}\text{rel}_1 + \sum_{i=1}^{p}\frac{\text{rel}_i}{\log_2 i} \tag{12}$$

Where: *rel$_i$* is the graded relevance of the result at position *i.*

DCG is normalized (13) using Ideal Discounted Cumulative Gain (IDCG) which is the ordered documents in the decreasing order of relevance.

$$\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p} \tag{13}$$

Precision, MAP, etc. return a score which is not negative and that is ranging from 0 to 1 for the top *n* documents (Ruback et al., 2018) e.g. P@n.

2) Results and discussion

To compare our suggested approach, we used PRF as a baseline for many reasons: First, unlike RF; PRF is automatic and can be done without the need for the user. And second, we used PRF because we wanted to compare our expansion approach with another expansion approach which is based on the same principal of the distribution approach i.e. using top feedback documents’ terms for expansion. Moreover, PRF is believed to decrease the relevancy of results when the query is difficult i.e. the top results are irrelevant. And since our approach takes into consideration the distribution of the terms in the top documents and in the whole corpus; we would like to check if the results will be better in this case compared to the other local approach (the PRF approach).

The results achieved by the proposed methods are reported in Tables 4–8 as well as in Figs. 3–6. We switched from low *k* values in Table 4 to high values in the other tables because we wanted to focus more on the higher values that are neglected by earlier studies in the domain.

The values in Table 4 that are marked in bold represent the highest improvement obtained for a certain number of expansion terms and a certain evaluation measure over PRF. According to the GOV2 results in Table 4; the exp_Bo1 increased the PRF MAP@10 for all values of *k* except for $k = 1$ where the MAP@10 decreased slightly. And the most

[2] http://www.lemurproject.org/indri.php
[3] http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html
[4] https://trec.nist.gov/data/docs_eng.html

Table 4
PRF, exp_Bo1, and exp_Bo1 + DBpedia performance using MAP@10 on two test collections. k is the number of expansion terms.

Collection	Model	*k*				
		1	3	5	7	10
GOV2	PRF	0.597	0.538	0.538	0.528	0.529
	Exp_Bo1	0.514	**0.555**	**0.643**	**0.582**	**0.557**
	Exp_Bo1 + DBpedia	0.397	0.445	0.436	0.414	0.514
AP	PRF	0.687	0.687	0.684	0.680	0.695
	Exp_Bo1	0.420	0.418	0.420	0.420	0.612
	Exp_Bo1 + DBpedia	0.420	0.407	0.420	0.420	0.662

Table 5
Accuracy retrieval on GOV2 dataset.

Measure	Model	*k*			
		10	20	50	100
P@10	PRF	0.399	0.389	0.419	0.403
	Exp_Bo1	0.417	0.447	0.451	0.482
	Exp_Bo1 + DBpedia	0.348	0.408	0.397	0.473
R@10	PRF	0.495	0.501	0.513	0.512
	Exp_Bo1	0.567	0.511	0.530	0.521
	Exp_Bo1 + DBpedia	0.563	0.510	0.546	**0.585**

Table 6
Accuracy retrieval on AP dataset.

Measure	Model	*k*			
		10	20	50	100
P@10	PRF	0.497	0.250	0.521	0.520
	Exp_Bo1	0.487	0.582	0.453	0.438
	Exp_Bo1 + DBpedia	0.561	0.584	0.451	0.425
R@10	PRF	0.705	0.336	0.627	0.660
	Exp_Bo1	0.517	0.607	0.526	0.550
	Exp_Bo1 + DBpedia	0.500	**0.615**	0.602	0.492

Table 7
Comparison of our approach using $k = 20$ with related works' approaches (AP dataset).

Approach	P@10
Sim_{LOD} approach (Dahir et al., Personal communication, 2018a)	0.460
Linked data COS-SIM + expansion labels (Dahir et al., Personal communication, 2018b)	0.230
DB-MI (Balaneshinkordan & Kotov, 2016)	0.233
Exp_Bo1 + DBpedia	0.584

Table 8
Comparison of our approach using $k = 100$ with related work approach (GOV dataset).

Approach	P@20
DB-MI (Balaneshinkordan & Kotov, 2016)	0.047
Exp_Bo1 + DBpedia	0.320

significant improvement of MAP for exp_Bo1 compared to baseline was that of $k = 5$. This improvement started to decrease whenever we used a higher value for k and became even lower when the k was smaller than 5. For the exp_Bo1 + DBpedia, the obtained results were lower than both baseline and exp_Bo1. We believe that this is due to the aim of the exp_Bo1 + DBpedia; which is retrieving relevant documents that do not necessarily contain query terms, but contain terms from DBpedia that are more commonly used by experts of the domain of search for example. This objective contributes to the improvement of the recall. And since the retrieved documents do not contain query terms they are

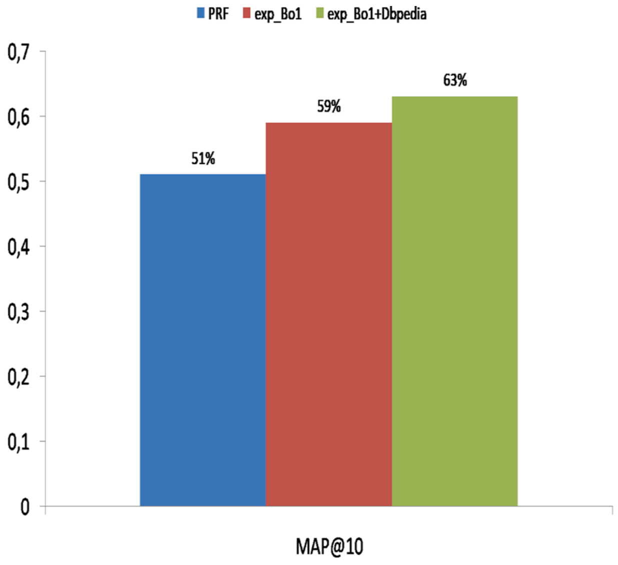

Fig. 3. Performance comparison using MAP@10 on the test collection GOV2 ($k = 100$).

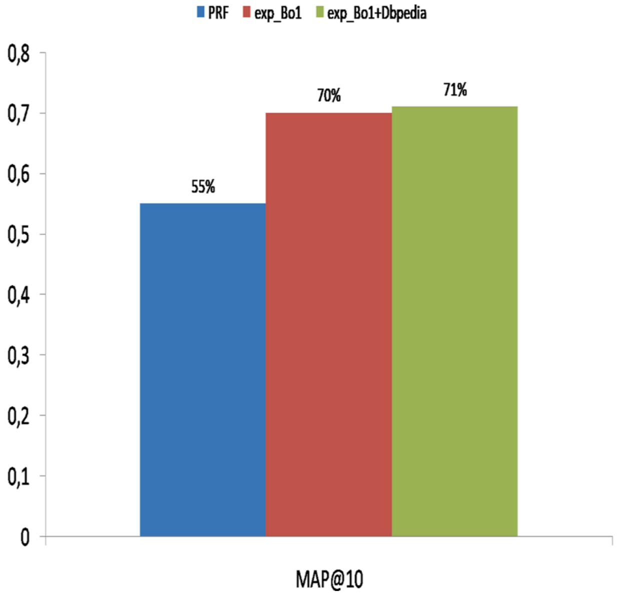

Fig. 4. Performance comparison using MAP@10 on the test collection AP ($k = 20$).

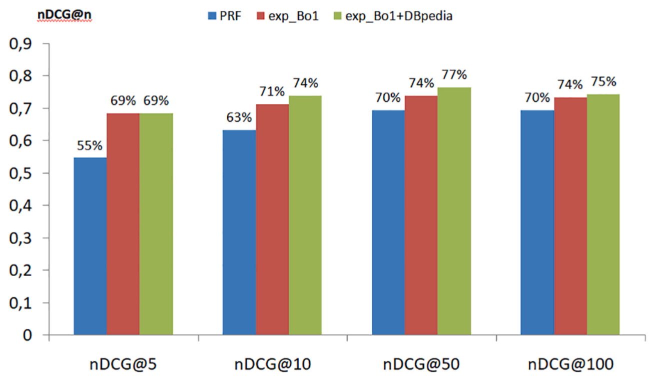

Fig. 5. Performance comparison using nDCG@n on GOV2 dataset ($k = 100$).

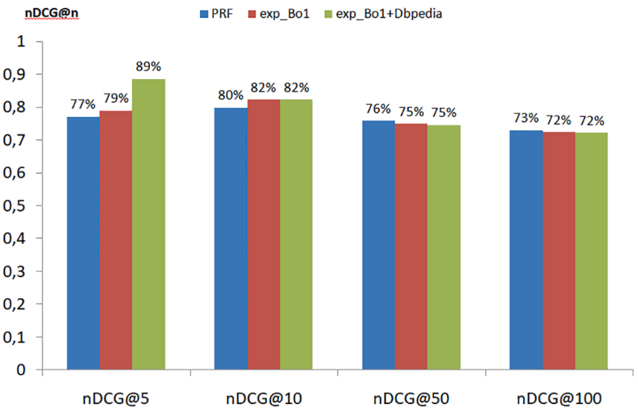

Fig. 6. Performance comparison using nDCG@n on AP dataset ($k = 20$).

ranked lower which gives low results of MAP@10.

For the AP results (Table 4); the MAP@10 for all exp_Bo1 cases of k was lower than that of the PRF. And the exp_Bo1 + DBpedia did not change, compared to exp_Bo1, for most of the used numbers of expansion terms except from the case of $k = 3$ where the MAP@10 decreased very slightly and the case of $k = 10$ where the MAP@10 increased. The decrease in exp_Bo1, compared to PRF, shows that the distribution technique can be damaging when using a very low number of expansion terms, especially that AP is not as big as GOV2. And exp_Bo1 requires a large collection to compare, efficiently, the distribution of a term in a top document with its distribution in the whole corpus.

From the GOV2 results in Table 5 and Fig. 3: we noticed that for the PRF the highest precision and recall at rank 10 were obtained when k was equal to 50. Moreover, the recall at rank 10 of $k = 50$ (0.513) was very close to that of $k = 100$ (0.512). The exp_Bo1′s highest improvements, over the PRF approach for both precision at 10 and MAP@10, occurred in the case of $k = 100$ (terms) where P@10 increased from 0.403 for PRF approach to 0.482 for exp_Bo1, and MAP@10 increased from 51% to 59%. But for the recall, the highest improvement was obtained in the case of $k = 10$ where it moved from 0.495 to 0.567. Also, we noticed that for MAP@10 and P@10 the best number of expansion terms was $k = 100$ followed by $k = 20$, then $k = 50$, and finally $k = 10$. Contrary to, recall at 10 for which the best number of expansion terms was $k = 10$, then $k = 50$, followed by $k = 20$, and finally $k = 100$. And concerning the exp_Bo1 + DBpedia approach; the highest improvements were again in the case of $k = 100$ with high improvements in both Recall, that moved from 0.521 for the exp_Bo1 to 0.585 for the exp_Bo1 + DBpedia, and MAP which increased from 59% for exp_Bo1 to 63% for exp_Bo1 + DBpedia. Also, the exp_Bo1 + DBpedia achieved significant improvements over PRF approach especially in the case of $k = 100$ where the MAP increase from 51% to 63%.

And from the AP results in Table 6 and Fig. 4: we found that the improvements in all measures were in the case of $k = 20$ for both the exp_Bo1 and the exp_Bo1 + DBpedia. But compared to the exp_Bo1; the improvements were not as high as the improvement that we noticed in the case of GOV2 test collection except for P@10 in the case of $k = 10$. We believe that the differences in the results may be related to the used collections of documents, their sizes, and the used queries.

In the next set of experiments, we used nDCG to measure the overall ranking accuracy of a retrieval system in case of multiple relevancy levels (which is the case for GOV2 documents relevancy). As shown in Fig. 5, the exp_Bo1 + DBpedia increased the nDCG@50 from 70% (PRF) and 74% (exp_Bo1) to 77% for our DBpedia based expansion approach. Also, we noticed that the distribution based methods improved significantly the ranking of the top 5 documents: nDCG@5 from 55% (PRF) to 69% (exp_Bo1 + DBpedia). And from Fig. 6, we noticed that the exp_Bo1 + DBpedia increased significantly the nDCG@5 from 77% (PRF) and 79% (exp_Bo1) to 89% for the exp_Bo1 + DBpedia and that nDCG@n gave lower results whenever we increased the number of retrieved documents.

According to the results in Tables 7 and 8, our best performing DBpedia approaches, i.e. the ones using $k = 20$ for AP dataset and $k = 100$ for GOV dataset, clearly outperformed association based approaches using DBpedia. Consequently, we can say that linked data perform better when applied on distribution approaches rather than association ones.

The reason why exp_Bo1 performs better than PRF is that in PRF the expansion terms are chosen depending on their frequency in the selected documents. Whereas, in Bo1 and distribution based methods in general the candidate expansion terms are chosen after comparing their distribution in the top documents with that of the whole corpus which leads to better results. Yet, neither PRF nor distribution methods solve the problem of difficult queries for which the IRS retrieves 0 relevant documents, especially for the top results. In fact, both PRF and distribution based techniques use terms form the top documents. Consequently, if the top documents are irrelevant, the use of such methods will not contribute to the improvement of the results and may instead damage the performance of the retrieval system and decrease the results.

Additionally, the use of DBpedia to further expand the queries helps improving the exp_Bo1 results especially in terms of recall. In fact, thanks to this knowledge base we find labels i.e. expressions that are more frequently used than the query's terms. For example, the label of "type ii diabetes" is "diabetes mellitus type 2". Also, thanks to DBpedia we find labels that are a full version of what an acronym (in the query) stands for e.g. the label of "IMF" is "International Monetary Fund". As a result, documents that do not contain the query terms but contain related or similar terms to them will be retrieved which will boost the recall.

Furthermore, query expansion approaches are more sensitive to very low numbers of expansion terms ($k < 5$) than to high numbers of expansion terms $k > 5$. For instance, the use of a very low number of expansion terms $k < 5$ (Table 4); either damages the MAP@10 results or increases them slightly. While, the use of numbers of terms that are higher than 5 ($k > 5$) for expansion; contributes generally to the improvement of MAP@10 (Table 4, Figs. 3, and 4). In other words, unlike (keikha et al., 2018) that encourage selecting expansion terms from the range of 5 to 15 words, we prove that even a (very) high number of expansion terms can be extremely beneficial for the improvement of retrieval results.

As for the applicability of our approach on other document collections, we believe that it can be used on any dataset no matter if it is domain specific or not; because the semantic source we used (i.e. DBpedia) is based on Wikipedia. Thus, DBpedia covers a broad range of subjects. Similarly, our approach can be applied on any kind of IRS; because they all aim at returning relevant results to the user that may not be knowledgeable about the domain of his or her search so as to use the right keywords. In other words whether the system's objective is to suggest activities to tourists in a tourist application or to answer health queries in a search engine or health application, our approach can always be applicable and beneficial.

5. Conclusion

Unlike previous studies, where the number of expansion terms is either chosen randomly or chosen after testing few possibilities by considering only low numbers of terms. We are, to our best knowledge, the first to test and consider very high numbers of terms from feedback documents that we further expand using semantic terms.

Furthermore, we prove that the adequate number of expansion terms to use may vary depending on the test collections because we get great results in the case of $k = 100$ terms for GOV2 test collection while we get high results in the case of $k = 20$ for AP collection. This difference may be caused by the fact that AP collection is small compared to GOV2 which is a large collection. But, in general the adequate number of expansion terms should not be very low ($k < 5$) to avoid damaging the results.

Also, we show that the exp_Bo1 boosts the results of the PRF approach. Moreover, we find that our exp_Bo1 + DBpedia approach does improve significantly: the PRF results and the exp_Bo1 results, in terms of R@10 and NDCG@5, and earlier studies using DBpedia in terms of P@n for two reasons. First, the attribute "rdfs:label" that we use is usually mono-valued. Second, it helps in dealing with the vocabulary mismatch problem by adding semantically related terms to the query which helps in the improvement of the recall.

Moreover, a system should be judged based on a certain measure and not all measures. In fact, there is not a system that improves results of all measures. Consequently, when judging a system we should consider that an IRS with a high precision may not have a high recall, etc. As a result, we choose the evaluation measure based on our objectives from the IRS.

CRediT authorship contribution statement

Sarah Dahir: Data curation, Methodology, Formal analysis, Writing - original draft. **Abderrahim El Qadi:** Conceptualization, Writing - review & editing. **Hamid Bennis:** Supervision, Validation.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abbes, R., Kopliku, A., Pinel-Sauvagnat, K., Hernandez, N., & Boughanem, M. (2013). Apport du Web et du Web de Données pour la recherche d'attributs.

Amati, G. (2003). *Probabilistic Models for Information Retrieval based on Divergence from Randomness* (Doctoral dissertation, PhD Thesis). UK: University of Glasgow.

Augenstein, I., Gentile, A. L., Norton, B., Zhang, Z., & Ciravegna, F. (2013, May). Mapping keywords to linked data resources for automatic query expansion. In Extended semantic web conference (pp. 101–112). Springer, Berlin, Heidelberg.

Balaneshinkordan, S., & Kotov, A. (2016, March). An empirical comparison of term association and knowledge graphs for query expansion. In European conference on information retrieval (pp. 761–767). Springer, Cham.

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. NIST special publication sp, pp. 69–69.

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR), 44*(1), 1–50.

Dahir, S., El Qadi, A., & Bennis, H. (Personal communication, 2018a). Enriching user queries using Dbpedia features and relevance feedback. Procedia Computer Science, 127, 499–504.

Dahir, S., El Qadi, A., & Bennis, H. (Personal communication, 2018b). An Association Based Query Expansion Approach Using Linked Data. In 2018 9th international symposium on signal, image, video and communications (ISIVC) (pp. 340–344). IEEE.

Di Marco, Antonio, & Navigli, Roberto (2013). Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics, 39*(3), 709–754.

El Ghali, Btihal, & El Qadi, Abderrahim (2017). Context-aware query expansion method using Language Models and Latent Semantic Analyses. *Knowledge and Information Systems, 50*(3), 751–762.

Goharian, N. (2020, January 27). Information Retrieval Evaluation, COSC 488: https://www.coursehero.com/file/8847955/Evaluation/.

Jain, Amita, Mittal, Kanika, & Tayal, Devendra K. (2014). Automatically incorporating context meaning for query expansion using graph connectivity measures. *Progress in Artificial Intelligence, 2*(2-3), 129–139.

Keikha, Andisheh, Ensan, Faezeh, & Bagheri, Ebrahim (2018). Query expansion using pseudo relevance feedback on wikipedia. *Journal of Intelligent Information Systems, 50* (3), 455–478.

Lafferty, J., & Zhai, C. (2001). September). Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 111–119).

Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011). September). DBpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems* (pp. 1–8).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Pal, D., Mitra, M., & Datta, K. (2013). Query expansion using term distribution and term association. arXiv preprint arXiv:1303.0667.

Rocchio, J. (1971). Relevance feedback in information retrieval. The Smart retrieval system-experiments in automatic document processing, 313–323.

Ruback, L., Casanova, M. A., Renso, C., & Lucchese, C. (2017). SELEcTor: Discovering similar entities on LinkEd DaTa by ranking their features. In 2017 IEEE 11th international conference on semantic computing (ICSC) (pp. 117–124). IEEE.

Ruback, L., Lucchese, C., Caraballo, A. A. M., García, G. M., Casanova, M. A., & Renso, C. (2018). Computing entity semantic similarity by features ranking. arXiv preprint arXiv:1811.02516.

Salton, G., & McGill, M. J. (1983). Introduction to modern information retrieval. New York: McGraw Hill Book Company.

Shekarpour, S., Höffner, K., Lehmann, J., & Auer, S. (2013, September). Keyword query expansion on linked data using linguistic and semantic features. In 2013 IEEE seventh international conference on semantic computing (pp. 191–197). IEEE.

Sinha, R., & Mihalcea, R. (2007, September). Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In International conference on semantic computing (ICSC 2007) (pp. 363–369). IEEE.

Spink, Amanda, Wolfram, Dietmar, Jansen, Major B. J., & Saracevic, Tefko (2001). Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology, 52*(3), 226–234.

Todor, A., Lukasiewicz, W., Athan, T., & Paschke, A. (2016, October). Enriching topic models with DBpedia. In OTM confederated international conferences "On the Move to Meaningful Internet Systems" (pp. 735–751). Springer, Cham.

Wikipedia contributors. (2019, December 31). Evaluation measures (information retrieval). In Wikipedia, The Free Encyclopedia. Retrieved 18:40, January 27, 2020, from https://en.wikipedia.org/w/index.php?title=Evaluation_measures_(information_retrieval)&oldid=933290621.

Xu, Bo, Lin, Hongfei, Lin, Yuan, & Guan, Yizhou (2020). Integrating social annotations into topic models for personalized document retrieval. *Soft Computing, 24*(3), 1707–1716.

Zhai, ChengXiang (2008). Statistical language models for information retrieval. *Synthesis Lectures on Human Language Technologies, 1*(1), 1–141.

Zong, Nansu, Lee, Sungin, & Kim, Hong-Gee (2015). Discovering expansion entities for keyword-based entity search in linked data. *Journal of Information Science, 41*(2), 209–227.

Zuva, Keneilwe (2012). Evaluation of information retrieval systems. *International Journal of Computer Science & Information Technology, 4*(3), 35–43.