Fielded Sequential Dependence Model for Ad-Hoc Entity Retrieval in the Web of Data

Nikita Zhiltsov*
Kazan Federal University
Kazan, 18 Kremlyovskaya Str, 420008, Russia
nzhilcov@kpfu.ru

Textocat
Kazan, 52 Peterburgskaya Str, 420074, Russia
nzhiltsov@textocat.com

Alexander Kotov
Department of Computer Science
Wayne State University
Detroit, MI 48202, USA
kotov@wayne.edu

Fedor Nikolaev
Department of Computer Science
Wayne State University
Detroit, MI 48202, USA
fedor@wayne.edu

ABSTRACT

Previously proposed approaches to ad-hoc entity retrieval in the Web of Data (ERWD) used multi-fielded representation of entities and relied on standard unigram bag-of-words retrieval models. Although retrieval models incorporating term dependencies have been shown to be significantly more effective than the unigram bag-of-words ones for ad hoc document retrieval, it is not known whether accounting for term dependencies can improve retrieval from the Web of Data. In this work, we propose a novel retrieval model that incorporates term dependencies into structured document retrieval and apply it to the task of ERWD. In the proposed model, the document field weights and the relative importance of unigrams and bigrams are optimized with respect to the target retrieval metric using a learning-to-rank method. Experiments on a publicly available benchmark indicate significant improvement of the accuracy of retrieval results by the proposed model over state-of-the-art retrieval models for ERWD.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Term Dependence; Entity Retrieval; Knowledge Graphs

1. INTRODUCTION

The past decade has witnessed the emergence of numerous large-scale publicly available knowledge bases, such as Freebase[1], DBpedia[2], Wikidata[3] and YAGO[4]. Open source knowledge bases typically adopt Resource Description Framework (RDF) data model and are published as part of the Linked Open Data (LOD)[5] cloud commonly referred to as the Web of Data. A similar trend exists in the industry as well (e.g. Google's Knowledge Graph, Facebook's Open Graph and Microsoft's Satori). Individual RDF datasets in the Web of Data can be considered as massive graphs, in which the nodes are the entities (resources) and the edges are semantic relations between the entities. Each resource describes an object in the Web of Data, which can either be a real entity (e.g. *Albert Einstein* or *Apple, Inc.*) or an abstract concept (*special relativity*). The relations between the entities are represented as subject-predicate-object triples (e.g. <Albert Einstein, *knownFor*, special relativity>).

Graph structured knowledge in general and RDF graphs in particular are well-suited for addressing the information needs that aim at finding specific entities, for which the top results returned by the search systems should consist of structured objects rather than individual documents. The analysis of Web search engine logs reported in [26] revealed that such information needs constitute more than half of search engine queries. They also proposed to classify entity-oriented queries into five classes, each of which requires different treatment by the search systems. The demand for efficient access to knowledge graphs gives rise to the task of Ad-hoc Entity Retrieval from the Web of Data (ERWD). As opposed to the traditional information retrieval task, in which search systems return documents in response to keyword queries, the goal of ERWD is to return an entity or a list of entities based on their unconstrained (and often fairly long) keyword descriptions. This task is focused on retrieval from knowledge bases, as opposed to utilization of knowledge bases in retrieval [7, 15]. Although entities have been studied in textual data mining (e.g. [16]), ERWD is a fairly recent trend in IR. It is related to entity retrieval in the context of expert finding, for which different unigram language model based approaches have been proposed (e.g. [1]). ERWD,

*work performed while the first author was visiting the Textual Data Analytics laboratory at Wayne State University

SIGIR'15, August 09 - 13, 2015, Santiago, Chile.

ACM 978-1-4503-3621-5/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767756 .

[1] http://freebase.com
[2] http://dbpedia.org
[3] http://wikidata.org
[4] http://www.mpi-inf.mpg.de/yago-naga/yago/
[5] http://lod-cloud.net/

however, provides its own unique set of challenges, such as designing effective entity representation methods and novel retrieval models.

While several methods [4, 21, 24] to address the first challenge have been previously proposed, only standard bag-of-words retrieval models, such as BM25 [2, 31] and language modeling (LM) based ones [8, 9, 22], as well as their multi-fielded extensions [4, 5, 20, 21, 24] have been applied to the task of entity retrieval from the Web of Data. It is known, however, that retrieval models incorporating term dependencies (in the form of ordered and unordered n-grams) are substantially more accurate than standard bag-of-words models in case of ad-hoc *document* retrieval [12], particularly for longer, verbose queries [3]. Markov Random Field (MRF) retrieval model [18, 19] provided a theoretical foundation to account for bigrams, as well as ordered and unordered phrases in IR, by representing a query as a graph of dependencies between the query terms. MRF calculates the score of each document with respect to a query as a linear combination of potential functions, each of which is computed based on a document and a clique in the query graph. Sequential Dependence Model (SDM) is a variant of the Markov Random Field model that assumes sequential dependence for the query terms and uses three potential functions: one that is based on unigrams and the other two that are based on bigrams, either as ordered sequences of terms or as terms co-occurring within a window of the pre-defined size.

To address the second challenge, the problem of designing effective retrieval models for ERWD, we propose Fielded Sequential Dependence Model (FSDM), which unifies and generalizes sequential dependence [18] and the mixture of language models (MLM) [23] retrieval models and allows to account for term dependencies in structured document retrieval. Although, in this work, we only experimented with entity retrieval from the knowledge graphs, our proposed model can be utilized for retrieval from collections of any structured documents.

The key contributions of this work are two-fold:

1. novel retrieval model for collections of structured documents and the algorithm to learn its parameters with respect to the target retrieval metric. Retrieval performance of the proposed model has been comprehensively evaluated with respect to the state-of-the-art baselines for ERWD using a publicly available benchmark covering different types of information needs;
2. novel multi-fielded entity representation scheme for ERWD that better captures the semantics of both the entities and relations between them.

The remainder of the paper is organized as follows. Section 2 provides a brief overview of previous related work. In Section 3, we present FSDM, an algorithm to optimize its parameters, and a method to construct multi-fielded entity representations. Experimental results are reported in Section 4 and Section 5 concludes the paper.

2. RELATED WORK

The retrieval model proposed in this paper builds upon the previous work along the following research directions.

Entity retrieval. The task of ERWD as answering arbitrary information needs expressed as keyword queries that aim at finding nodes or aspects of nodes in RDF graphs was first introduced by Pound et al. [26]. They also proposed to classify such queries into five categories: entity queries, type queries (i.e. list search queries or telegraphic queries), attribute queries, relation queries, and other queries. Although multiple open entity retrieval evaluation campaigns providing test collections and query sets covering a variety of information needs and relevance judgments have been conducted [10], most of the previous work along this direction focused on one particular query type. For example, the SemSearch Entity Search challenge[6] focused on finding one particular entity described by a keyword query. Previous successful retrieval approaches for queries of this type involved construction of multi-fielded entity representations by grouping entity attributes together by type [24], into title and content [21], according to manually determined importance [4] or into a two-level hierarchy [20]. Several methods adopted a two-stage approach, in which the initial retrieval results obtained using standard bag-of-words retrieval models were first expanded using relations in the knowledge graph and then the expanded results were re-ranked based on entity similarity. Tonon et al. [31] used Jaro-Winkler similarity between the entity names, while Herzig et al. [11] used Jenson-Shannon divergence between the language models of entities. Zhiltsov and Agichtein [33] proposed a learning-to-rank based approach, which combines explicit entity information with semantic similarity between the entities in latent space determined using a modified algorithm for tensor factorization.

The INEX 2009 Entity Ranking (INEX-XER)[7] track introduced an entity list completion task and provided an evaluation platform for type queries. The TREC 2009 Entity track focused on two related retrieval tasks "related entity finding" (i.e. finding all entities related to a given entity query) and "entity list completion" (i.e. finding entities with common properties given some examples). The List Search track from the SemSearch challenge targets a group of entities that match a keyword query. Bron et al. [5] proposed a hybrid approach that linearly combines the scores of the mixture of language models and a structure-based method, which captures the statistics of predicate-object pairs in triples shared by entity candidates. Elbassouni et al. [9] focused on the case of structured queries consisting of RDF triples and proposed a method that constructs LMs (as multinomial distributions over RDF triples) for both the queries and each possible result sub-graph. The ranking is based on the Kullback-Leibler divergence between the query LM and the LMs of each result sub-graph. The SemSets model [6] utilizes the relevance of entities to automatically constructed categories (semantic sets, SemSets) measured according to structural and textual similarity. Their approach combines a retrieval model with the methods for spreading activation over the link structure of a knowledge graph and evaluation of membership in semantic sets.

Question Answering over Linked Data (QALD)[8] evaluation campaigns aim at developing retrieval methods to answer sophisticated question-like queries. The majority of queries are natural language questions that are focused on finding one particular entity (or several entities) as exact answers to these questions. A method based on integer linear

[6] `http://km.aifb.kit.edu/ws/semsearch\{10|11\}`
[7] `http://www.l3s.de/~demartini/XER09/`
[8] `http://www.sc.cit-ec.uni-bielefeld.de/qald/`

programming for joint segmentation and disambiguation of question-like queries was proposed in [32]. Shekarpour et al. [29] applied the Hidden Markov Model to map question terms into entities and relations in the knowledge graph and translated keyword queries into structured SPARQL queries. For comprehensive overview of approaches to entity retrieval, we refer an interested reader to the recent surveys [17] [27].

Multi-fielded retrieval models. Multi-fielded extensions of different bag-of-words retrieval models have been proposed for structured document retrieval. BM25F [28] allows to either combine the BM25 retrieval scores of individual document fields with different weights into the final document retrieval score or calculate the aggregated basic retrieval statistics, such as TF and field lengths across multiple fields and use them in the original retrieval formula. Mixture of Language Models (MLM) [23] is a multi-fielded extension of the query likelihood retrieval model [25], a standard language modeling based retrieval model. In MLM, the retrieval score of a structured document is a linear combination of probabilities of query terms in the language models calculated for each document field. Although individual field weights in BM25F and MLM can be tuned for a particular collection, they are fixed across different query terms. To overcome this limitation, Probabilistic Retrieval Model for Semistructured Data (PRMS) [14] maps each query term into document fields using probabilistic classification based on collection statistics. Although PRMS was originally proposed for XML retrieval, it was later applied to ERWD [2]. Field Relevance Model (FRM) [13] is an extension of PRMS to the case of relevance feedback.

Incorporating term dependencies into retrieval models. Metzler and Croft proposed the MRF retrieval model, which based on the assumption about the dependencies between the query terms, has three variants: full independence (FI), sequential dependence (SD) and full dependence (FD) models. Bendersky et al. [3] extended SDM to allow learning optimal weights for unigrams and bigrams as a linear combination of features that are based on internal and external collection statistics. Huston and Croft [12] systematically evaluated a large number of bigram dependence models across short and long queries and concluded that retrieval models incorporating term dependencies consistently improve retrieval accuracy over the standard bag-of-words retrieval models. They also experimentally demonstrated that the performance of term dependence models can be significantly improved through parameter tuning.

3. APPROACH

In this section, we introduce the Fielded Sequential Dependence Model (FSDM), a novel model for structured document retrieval, and describe our strategy for building multi-fielded entity description documents.

3.1 FSDM

One of the limitations of standard SDM for structured document retrieval is that it considers term matches in different parts of a document as equally important (i.e. having the same contribution to the final retrieval score of a document), thus disregarding the document structure. For example, in case of unigrams, the potential function looks as follows:

$$f_T(q_i, D) = \log P(q_i|\theta_D) = \log \frac{tf_{q_i,D} + \mu \frac{cf_{q_i}}{|C|}}{|D| + \mu}$$

where q_i is a query term, D is a document, $tf_{q_i,D}$ is the frequency of q_i in D, $|D|$ is the document length, μ is a Dirichlet prior, that is usually set to the average document length in the collection, cf_{q_i} is the collection frequency of q_i and $|C|$ is the total number of terms in the collection. To adapt the MRF framework to multi-fielded entity descriptions, we propose to replace a single document language model $P(q_i|\theta_D)$ with a mixture of language models (MLM) for each document field. Hence, our model is an extension of sequential dependence model, although the same transformation can be applied to the full dependence model.

Consequently, the potential function for unigrams in case of FSDM is:

$$\tilde{f}_T(q_i, D) = \log \sum_j w_j^T P(q_i|\theta_D^j) = \log \sum_j w_j^T \frac{tf_{q_i,D^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|D^j| + \mu_j}$$

where $j = \overline{1, F}$, F is the number of fields, θ_D^j is a language model of field j smoothed using its own Dirichlet prior μ_j and w_j are the field weights with the following constraints: $\sum_j w_j = 1, w_j \geqslant 0$; tf_{q_i,D^j} is the term frequency of q_i in field j of document D; $cf_{q_i}^j$ is the collection frequency of q_i in field j; $|C_j|$ is the total number of terms in field j across all the documents in the collection and $|D^j|$ is the length of field j in D.

The potential functions for ordered and unordered bigrams $q_{i,i+1} = (q_i, q_{i+1})$ in the query are defined as follows:

$$\tilde{f}_O(q_{i,i+1}, D) = \log \sum_j w_j^O \frac{tf_{\#1(q_{i,i+1}),D^j} + \mu_j \frac{cf_{\#1(q_{i,i+1})}^j}{|C_j|}}{|D^j| + \mu_j}$$

$$\tilde{f}_U(q_{i,i+1}, D) = \log \sum_j w_j^U \frac{tf_{\#uw8(q_{i,i+1}),D^j} + \mu_j \frac{cf_{\#uw8(q_{i,i+1})}^j}{|C_j|}}{|D^j| + \mu_j}$$

where $tf_{\#1(q_{i,i+1}),D^j}$ is the frequency of exact phrase q_iq_{i+1} in field j of document D, $cf_{\#1(q_{i,i+1})}^j$ is the collection frequency of exact phrase q_iq_{i+1} in field j, $tf_{\#uw8(q_{i,i+1}),D^j}$ is the number of times terms q_i and q_{i+1} occur together within a window of 8 word positions in field j of document D, regardless of the order of these terms.

Having separate mixtures of language models with different weights for unigrams as well as ordered and unordered bigrams gives FSDM the flexibility to adjust the multi-fielded document scoring strategy for matching query terms and phrases depending on the query type (entity, type, relation, etc.). Intuitively, an FSDM field weighting scheme, in which unordered bigram matches in the descriptive fields of entity documents are given higher weight than the matches in the name field, should be more effective for informational queries, while giving higher weight to ordered bigram matches in the name field can be beneficial for navigational queries. For example, precision of retrieval results for an informational query SemSearch_LS-1 *"apollo astronauts who walked on the moon"* is likely to increase when more importance is given to the matches of unordered bigrams *apollo astronauts* and *walked moon* within a window inside the descriptive fields of entity documents, rather than the name field, while giving higher weights to the matches of the same unordered bigrams in the name field is likely to have the opposite effect.

Substituting the potential functions $\psi(q_i, D; \Lambda)$ and $\psi(q_i, q_{i+1}, D; \Lambda)$ in the formula for the joint distribution over

the document and query terms in the MRF model completes the transformation of SDM into FSDM:

$$P_{G,\Lambda}(Q,D) = \frac{1}{Z_\Lambda}\prod_{c\in Cliques(G)}\psi(c;\lambda_c)$$
$$= \frac{1}{Z_\Lambda}\times exp[\lambda_T\tilde{f}_T(q_i,D)]\times$$
$$exp[\lambda_O\tilde{f}_O(q_i,q_{i+1},D)+\lambda_U\tilde{f}_U(q_i,q_{i+1},D)]$$

Note that for the purpose of ranking, it is sufficient to compute the following posterior probability:

$$P_\Lambda(D|Q)=\frac{P_{G,\Lambda}(Q,D)}{P_\Lambda(Q)}$$

Consequently, FSDM is a term dependence retrieval model with the following ranking function:

$$P_\Lambda(D|Q)\stackrel{rank}{=}\lambda_T\sum_{q\in Q}\tilde{f}_T(q_i,D)+$$
$$\lambda_O\sum_{q\in Q}\tilde{f}_O(q_i,q_{i+1},D)+$$
$$\lambda_U\sum_{q\in Q}\tilde{f}_U(q_i,q_{i+1},D) \quad (1)$$

It is easy to see from Equation 1 that MLM is a special case of FSDM, when $\lambda_T=1,\lambda_O=0,\lambda_U=0$.

3.2 Parameter Estimation

Overall, FSDM has $3*F+3$ free parameters: $\langle \boldsymbol{w}^T,\boldsymbol{w}^O,\boldsymbol{w}^U,\boldsymbol{\lambda}\rangle$. Before introducing the parameter estimation procedure, we would like to emphasize two properties of the ranking function of FSDM. The first property is linearity with respect to $\boldsymbol{\lambda}$. We therefore can apply any linear learning-to-rank algorithm to optimize the ranking function with respect to $\boldsymbol{\lambda}$. The second property is linearity with respect to $\boldsymbol{w}$ of the arguments of monotonic $\tilde{f}(\cdot)$ functions. Therefore, optimization of the arguments as linear functions with respect to $\boldsymbol{w}$, leads to optimization of each function $\tilde{f}(\cdot)$.

Based on the above properties, we propose a two-stage Algorithm 1 to optimize the free parameters of FSDM with respect to the target retrieval metric, which in our case is Mean Average Precision (MAP).

Algorithm 1 An algorithm for training FSDM parameters.

1: $Q\leftarrow$ Train queries
2: **for** $s\in\{T,O,U\}$ **do**
3: $\quad\boldsymbol{\lambda}=\boldsymbol{e}_s$
4: $\quad\hat{\boldsymbol{w}}^s\leftarrow CA(Q,\boldsymbol{\lambda})$
5: **end for**
6: $\hat{\boldsymbol{\lambda}}\leftarrow CA(Q,\hat{\boldsymbol{w}}_T,\hat{\boldsymbol{w}}_O,\hat{\boldsymbol{w}}_U)$

In the first stage (lines 2-5), the algorithm independently estimates the optimal values of MLM parameters for unigrams, ordered and unordered bigrams. The unit vectors $\boldsymbol{e}_T=(1,0,0)$, $\boldsymbol{e}_O=(0,1,0)$, $\boldsymbol{e}_U=(0,0,1)$ are the corresponding settings of the parameters $\boldsymbol{\lambda}$ in the formula of FSDM ranking function (Equation 1). Because of linearity of MLM ranking function with respect to $\boldsymbol{w}$, we make use of the coordinate ascent (CA) algorithm, proposed in [19], to directly optimize MAP. The starting values for each $\boldsymbol{w}_s$ are uniform, i.e., equal to $\frac{1}{F}$ each. Therefore, CA iteratively optimizes MAP by performing a series of line searches along one coordinate direction under the constraint of non-negativity of $\boldsymbol{w}_s$. It repeatedly optimizes each of the parameters $w_1^s,\ldots,w_F^s$, while holding all other parameters fixed. Since MLMs are optimized independently, this stage can be easily parallelized to speed-up the training process.

In the second stage (line 6), the algorithm optimizes the parameters $\boldsymbol{\lambda}$ in Equation 1 on the same query set, given the optimal values of $\hat{\boldsymbol{w}}_T$, $\hat{\boldsymbol{w}}_O$, $\hat{\boldsymbol{w}}_U$. Again, since the right-hand side of Equation 1 is a linear function with respect to $\boldsymbol{\lambda}$, the CA algorithm can be applied to maximize the target metric directly. It starts with $\boldsymbol{\lambda}=(1,0,0)$, which is equivalent to MLM ranking model, and iterates until the gain in MAP is less than a given threshold. To avoid the changes of signs of the document ranking scores that may distort the ranking, non-negativity constraints on $\boldsymbol{\lambda}$ are enforced during CA.

Table 1: **Proposed scheme for multi-fielded representation of entity e.**

Field	Condition
names	$o:\exists(e,p,o)\&$ $p\in P_{names}=regex(*[name\|label]\$)$
attributes	$o:\exists(e,p,o)\&p\in P_{datatypes}\&p\notin P_{names}$
categories	$o:\exists(e_1,p_1,e_2)\&(e_2,p_2,o)\&$ $\&p\in P_{categories}\&p_2\in P_{names}$
similar entity names	$o:\exists(e_1,p_1,e_2)\&(e_2,p_2,o)\&$ $e_2\in E_{sim}(e_1)\&p_2\in P_{names}$
related entity names	$o:\exists(e_1,p_1,e_2)\&(e_2,p_2,o)\&$ $e_2\notin E_{sim}(e_1)\&p_2\in P_{names}$

3.3 Modeling Entity Descriptions

Since multi-fielded entity representation proved to be beneficial for ERWD [21, 24], we propose a novel five-field entity representation scheme (Table 1). Attribute values, that satisfy the provided condition, are aggregated into the corresponding field.

The first two fields are the properties of the entities themselves. Other fields capture information from different entities that are related to the given entity. The *names* field contains conventional names of the entities, such as the name of a person or the name of an organization. Some of the predicates that can be used for this purpose are *label* from RDF Schema[9], *name* from FOAF Ontology[10], or */type/object/name* from Freebase[11]. The *attributes* field includes all datatype properties, other than names. The examples are values of *abstract* and *foundingDate* predicates from DBpedia. We found it to be helpful to include predicate names along with the predicate values, e.g. "founding date 1964".

The next three fields are extracted from the *names* field of related entities. The *categories* field combines classes or groups, to which the entity can be assigned. In DBpedia, the membership of entities in classes is represented using the *subject* property, which corresponds to the categories of a Wikipedia article. The *profession* and *ethnicity* attributes for people or the *industry* attribute for companies in Freebase have close semantics. The *similar entity names* field aggregates the names of the entities that are very similar or identical to a given entity, since it is often the case that

[9] `http://www.w3.org/TR/rdf-schema/`
[10] `http://www.foaf-project.org/`
[11] `http://freebase.com`

Table 2: **Multi-fielded entity document for the U.S. president *Barack Obama* (the terms are preprocessed).**

Field	Content
names	barack obama barack hussein obama ii
attributes	44th current president united states birth place honolulu hawaii
categories	democratic party united states senator nobel peace prize laureate christian
similar entity names	barack obama jr barak hussein obama barack h obama ii
related entity names	spouse michelle obama illinois state predecessor george walker bush

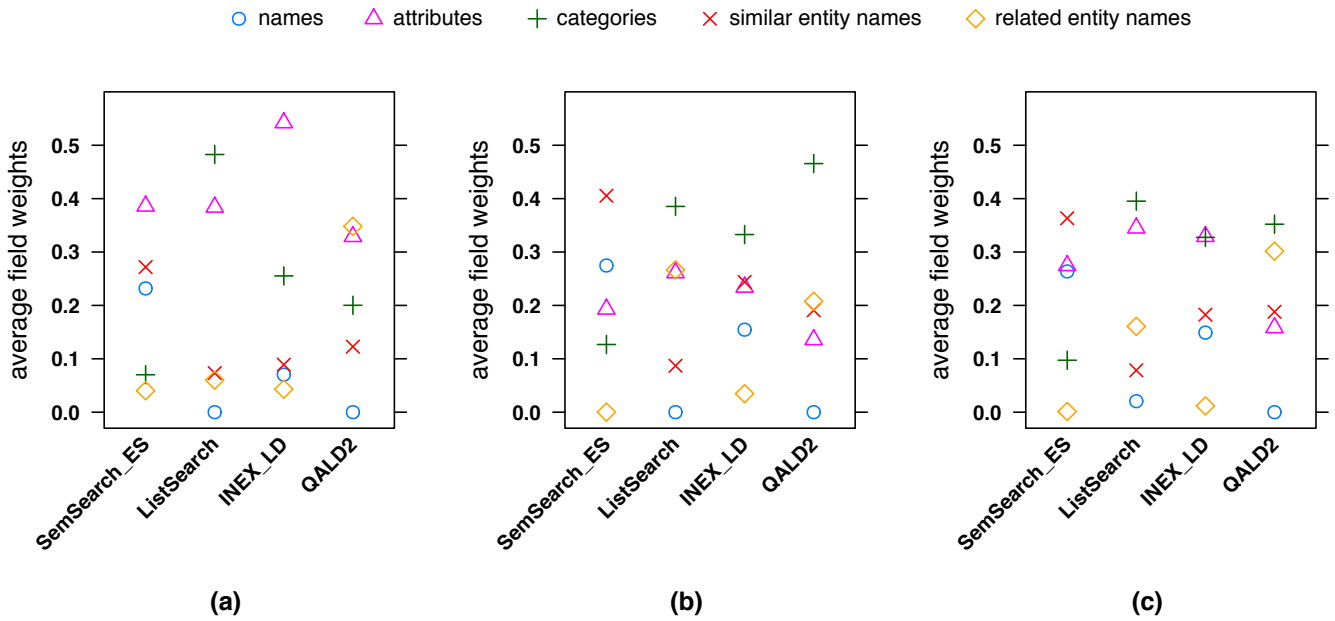


Figure 1: **Field weights in mixtures of language models for (a) unigrams (b) ordered bigrams and (c) unordered bigrams averaged over 5 folds for each query set.**

different knowledge bases (or even the same one) can refer to the same entity with different resource identifiers or describe it according to different schemata. In DBpedia, these values can be obtained with $S1\_1$ SPARQL query from [31] that takes into account *sameAs* property from OWL schema along with *redirect* and *disambiguates* from DBpedia. Finally, the *related entity names* field includes the names of the entities that are part of the same RDF triple with a given entity along with the predicate names, e.g. "spouse michelle obama". Table 2 provides an example of the entity document created using the above approach.

4. EXPERIMENTS

4.1 Experimental setup

We compare the effectiveness of FSDM with existing entity retrieval models based on a publicly available benchmark[12] [2], which combines 485 queries and the corresponding relevance judgments from the past entity retrieval evaluation campaigns. The knowledge graph used in our evaluation is DBPedia 3.7[13]. DBPedia is a structured version of on-line encyclopedia Wikipedia, which provides the descriptions of over 3.5 million entities belonging to 320 classes.

Table 3: **Statistics of the query sets used for evaluation.**

Query set	Amount	Query types
SemSearch ES	130	Entity
ListSearch	115	Type
INEX-LD	100	Entity, Type, Attribute, Relation
QALD-2	140	Entity, Type, Attribute, Relation

We report the retrieval results obtained on the query sets in Table 3:

- **SemSearch ES**: this query set primarily contains named entity queries (e.g. *"charles darwin"*, *"orlando florida"*);
- **ListSearch**: this query set combines three query sets (INEX-XER, SemSearch LS, TREC Entity) from [2], since each one of them primarily consists of type queries (e.g. *"continents in the world"*, *"products of medimmune, inc."*);
- **INEX-LD**: this query set covers different types of queries – named entity queries, type queries, relation queries, and attribute queries (e.g. *"einstein relativ-*

[12] `http://bit.ly/dbpedia-entity`
[13] `http://wiki.dbpedia.org/Downloads37`

ity theory", *"tango music composers"*, *"prima ballerina bolshoi theatre 1960"*);

- **QALD-2**: the Question Answering over Linked Data query set contains natural language questions of 4 different types: e.g., *"who created wikipedia?"* (entity); *"give me all soccer clubs in Spain"* (type); *"what is the currency of the czech republic"* (attribute); *"which books by kerouac were published by viking press?"* (relation).

In total, there are 13,090 available positive relevance judgments. Although some query sets (e.g. SemSearch) contain graded relevance judgments, for the purpose of consistency, we treat all relevance judgments as binary and report only the corresponding retrieval metrics.

For all experiments in this work, we used the multi-fielded entity descriptions constructed from DBpedia RDF graph according to the method presented in Section 3.3. Indexed terms were lower-cased, filtered using the INQUERY stoplist, and stemmed using the Krovetz stemmer. All retrieval models used in the experiments reported in this work were implemented using the Galago Search Engine[14][15]. Parameters of retrieval models were optimized with respect to the Mean Average Precision (MAP) using the Galago's implementation of the coordinate ascent learner based on 5-fold cross validation. All reported evaluation metrics were macro-averaged over 5 folds.

4.2 Parameter tuning

In this section, we discuss the details of optimization procedure for the field weights $\boldsymbol{w}$ and SDM/FSDM parameters $\boldsymbol{\lambda}$, and provide our interpretation of the learned values.

To optimize the field weights in FSDM, we uniformly initialize them and run the CA algorithm under the sum normalization and non-negativity constraints with 5 random restarts. We do not optimize the Dirichlet priors μ_j in language models and set them equal to the average (document or field) lengths, respectively. For optimizing the vectors of SDM parameters $\lambda_T, \lambda_O, \lambda_U$, we initialize them to $(1, 0, 0)$ and run the CA algorithm with 3 random restarts. The unordered window size was set to 8 in all cases, as suggested in previous work [3, 18].

We begin our analysis with the weights for unigram MLMs. Figure 1a depicts the distribution of the learned field weights averaged over all folds for each query set. Interestingly, while fields have similar relative importance across most query sets, some query sets (i.e. query types) are clear outliers. We observe that, compared to other query types, the optimized model strongly favors *names* and *similar entity names* fields for named entity queries (SemSearch ES query set). The *categories* field is important for type queries (ListSearch), while on QALD-2 query set, FSDM puts particularly strong emphasis on *related entity names*, which can be explained by the fact that this query set mostly consists of relation queries, which require capturing the related entity context. INEX-LD query set includes a large number of attribute queries that aim at finding entities primarily through their attributes instead of names. Thus, we observe that the model assigns higher weights to the *attributes* field.

[14] `http://www.lemurproject.org/galago.php`

[15] source code for all retrieval models used in this work as well as run files is available at `https://github.com/teanalab/FieldedSDM`

Since FSDM includes bigram language models, next we investigate the field weights in MLMs for ordered (Figure 1b) and unordered (Figure 1c) bigrams. In particular, we observe that higher weights are assigned to *names* and *similar entity names* fields for entity queries, for which the ordered bigram matchings in these fields is an important relevance factor. Another difference of bigram from unigram MLMs is the increased importance of *categories* field for other (non-entity) types of queries. This can be explained by the fact that bigrams are more effective in matching class names of entities than unigrams. For example, for the query QALD2_tr-89 *"give me all soccer clubs in the premier league"*, an English football club *Bolton Wanderers F.C.*, which is a relevant entity, is assigned to a few DBpedia categories, including *Premier League clubs*. Given this query intent, scoring entities based on matching bigrams *club premier* and *premier league* in the *categories* field is much more effective than scoring based on matching unigrams, since *premier league* is an important specification for the concept *soccer club* and should be considered holistically, as a phrase.

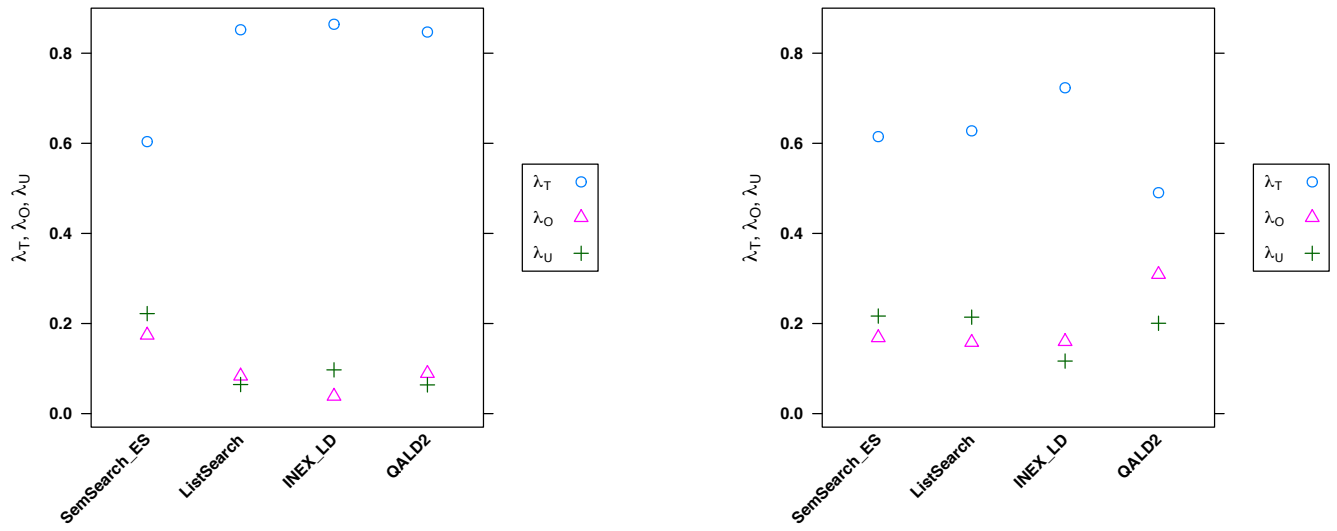
From the perspective of modeling entity descriptions, we can conclude that the *attributes* field is consistently considered to be a very valuable context for queries of any type. The *names* field as well as the *similar entity names* field are highly important for queries aiming at finding named entities, while distinguishing *categories* from *related entity names* is particularly important for type queries.

Overall, these findings support our initial hypothesis that the field weights are dependent on query types. Next, we focus on the analysis of the learned weights $\boldsymbol{\lambda}$ for SDM and FSDM, the distribution of which is depicted in Figures 2a and 2b. The optimal weights of SDM on all query sets, except SemSearch ES, are close to the standard scheme of (0.8, 0.1, 0.1), which was shown to be optimal in several previous works [3, 12, 18]. This discrepancy with SemSearch ES illustrates the significance of bigram matches for named entity queries. The optimal weights of FSDM indicate increased importance of bigram matches on every query set, especially on QALD-2. It follows that transformation of SDM into FSDM increases the importance of bigram matches, which ultimately improves the retrieval performance, as we will demonstrate next.

4.3 Comparison with the baselines

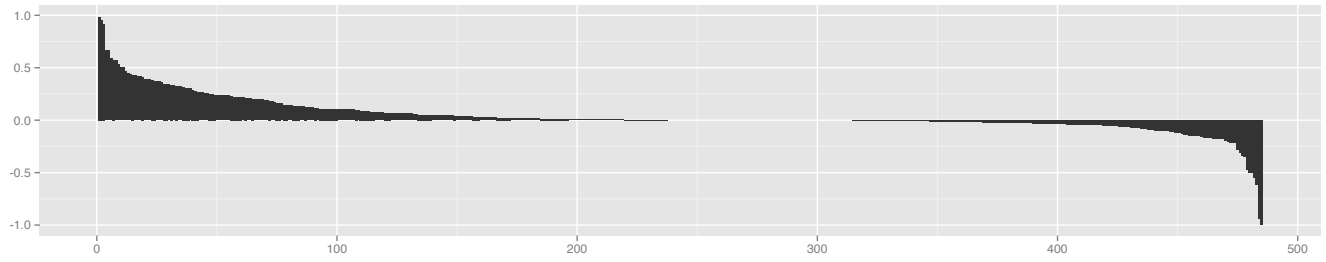
In this section, we present the results of comparing FSDM with the state-of-the-art models for ERWD. In particular, we use MLM-CA (unigram MLM optimized with respect to the field weights w_j by CA) and SDM-CA (SDM optimized with respect to the λ_T, λ_O, λ_U weights by CA) as the baselines. We also used MLM as a baseline to test whether incorporating term dependencies leads to improvement of entity retrieval. SDM leverages only unstructured entity descriptions, in which all fields are merged into a single document. The difference with SDM is measured to show the importance of fielded document representation for ERWD. Additionally, we report the results of the PRMS model using our entity descriptions, since it has been previously shown to have good performance in case of two-fielded entity representations. Finally, we include the results recomputed from the run files of the methods used for evaluation in [2].

Table 4 summarizes the retrieval results of all models on SemSearch ES, ListSearch, INEX-LD, QALD-2 query sets, and the entire query set. MAP and a preference-based mea-

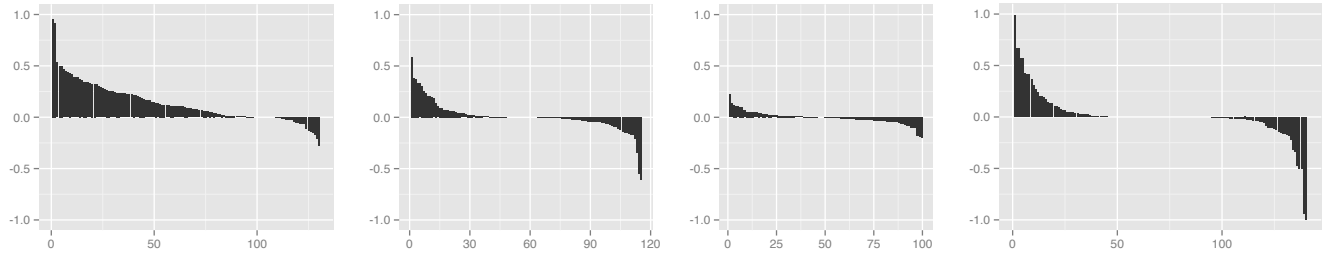


(a) (b)

Figure 2: **Values of λ in (a) SDM; (b) FSDM averaged over 5 folds for each query set.**



(a) All queries



(b) SemSearch ES (c) ListSearch (d) INEX-LD (e) QALD-2

Figure 3: **Topic-level differences in average precision between FSDM and SDM. over a) All queries; b) SemSearch ES; c) ListSearch; d) INEX-LD; e) QALD-2 query sets. Positive values indicate FSDM is better.**

sure (b-pref) are calculated at the top 100 results. As follows from Table 4, the SDM-CA and MLM-CA baselines (optimized SDM and MLM) both outperform previously proposed models on the entire query set, most significantly on QALD-2 and ListSearch query sets. However, the performance of SDM remarkably drops on SemSearch ES query set. To make sure that SDM-CA is not overfit, we run SDM using a standard weighting scheme (0.8, 0.1, 0.1) and got very close results with respect to MAP – 0.258 on SemSearch ES, 0.196 on ListSearch, 0.114 on INEX-LD, 0.186 on QALD-2, and 0.193 on the query set including all queries.

The results of PRMS are significantly worse compared to MLM in our settings, which indicates that the performance of this model degrades in case of a large number of fields in entity descriptions. Bag-of-words models, such as BM25 and LM, achieve comparable results with structured document retrieval models on the more heterogeneous INEX-LD query set, which includes the queries of different types.

Table 4: **Comparison of retrieval models on SemSearch ES, ListSearch, INEX-LD, QALD-2 query sets using the benchmark in [2]. "*" and "†" indicate statistically significant improvement over MLM-CA and SDM-CA baselines, respectively, as measured by the Fisher's randomization test ($\alpha = 0.05$) [30]. Relative improvement over MLM-CA/SDM-CA is shown in parenthesis.**

	SemSearch ES			
	MAP	**P@10**	**P@20**	**b-pref**
LM [2]	$0.314_{\dagger}$ (−1.9%/+23.6%)	$0.251_{\dagger}$ (+0.4%/+24.3%)	$0.179_{\dagger}$ (0.0%/+20.1%)	0.655 (−2.8%/−2.4%)
BM25 [2]	$0.326_{\dagger}$ (+1.9%/+28.3%)	$0.256_{\dagger}$ (+2.4%/+26.7%)	$0.177_{\dagger}$ (−1.1%/+18.8%)	0.658 (−2.4%/−1.9%)
MLM-tc [2]	$0.354_{\dagger}$ (+10.6%/+39.4%)	$0.284^{*}_{\dagger}$ (+13.6%/+40.6%)	$0.200^{*}_{\dagger}$ (+11.7%/+34.2%)	0.694 (+3.0%/+3.4%)
BM25F-tc [2]	$0.334_{\dagger}$ (+4.4%/+31.5%)	$0.263_{\dagger}$ (+5.2%/+30.2%)	$0.180_{\dagger}$ (+0.6%/+20.8%)	0.666 (−1.2%/−0.7%)
PRMS [2]	$0.323_{\dagger}$ (+0.9%/+27.2%)	$0.251_{\dagger}$ (+0.4%/+24.3%)	$0.175_{\dagger}$ (−2.2%/+17.4%)	0.657 (−2.5%/−2.1%)
PRMS	0.230^{*} (−28.1%/−9.4%)	$0.177^{*}_{\dagger}$ (−29.2%/−12.4%)	$0.125^{*}_{\dagger}$ (−30.2%/−16.1%)	$0.569^{*}_{\dagger}$ (−15.6%/−15.2%)
MLM-CA	0.320	0.250	0.179	0.674
SDM-CA	0.254^{*} (−20.6%)	0.202^{*} (−19.2%)	0.149^{*} (−16.8%)	0.671 (−0.4%)
FSDM	$\mathbf{0.386}^{*}_{\dagger}$ (+20.6%/+52.0%)	$\mathbf{0.286}^{*}_{\dagger}$ (+14.4%/+41.6%)	$\mathbf{0.204}^{*}_{\dagger}$ (+14.0%/+36.9%)	$\mathbf{0.750}^{*}_{\dagger}$ (+11.3%/+11.8%)
	ListSearch			
	MAP	**P@10**	**P@20**	**b-pref**
LM [2]	$0.161_{\dagger}$ (−15.3%/−18.3%)	$0.216^{*}_{\dagger}$ (−14.3%/−14.3%)	$0.170^{*}_{\dagger}$ (−11.5%/−15.8%)	$0.403_{\dagger}$ (−5.8%/−14.4%)
BM25 [2]	$0.167_{\dagger}$ (−12.1%/−15.2%)	0.232 (−7.9%/−7.9%)	0.186 (−3.1%/−7.9%)	$0.414_{\dagger}$ (−3.3%/−12.1%)
MLM-tc [2]	$0.153^{*}_{\dagger}$ (−19.5%/−22.3%)	$0.195^{*}_{\dagger}$ (−22.6%/−22.6%)	$0.16^{*}_{\dagger}$ (−16.7%/−20.8%)	$0.398_{\dagger}$ (−7.0%/−15.5%)
BM25F-tc [2]	$0.159^{*}_{\dagger}$ (−16.3%/−19.3%)	$0.221^{*}_{\dagger}$ (−12.3%/−12.3%)	$0.174_{\dagger}$ (−9.4%/−13.9%)	$0.402_{\dagger}$ (−6.1%/−14.6%)
PRMS [2]	0.178 (−6.3%/−9.6%)	0.240 (−4.8%/−4.8%)	0.200 (+4.2%/−1.0%)	$0.407_{\dagger}$ (−4.9%/−13.6%)
PRMS	$0.111^{*}_{\dagger}$ (−41.6%/−43.7%)	$0.154^{*}_{\dagger}$ (−38.9%/−38.9%)	$0.121^{*}_{\dagger}$ (−37.0%/−40.1%)	$0.310^{*}_{\dagger}$ (−27.6%/−34.2%)
MLM-CA	0.190	0.252	0.192	0.428
SDM-CA	0.197 (+3.7%)	0.252 (0.0%)	0.202 (+5.2%)	$\mathbf{0.471}^{*}$ (+10.0%)
FSDM	**0.203** (+6.8%/+3.0%)	**0.256** (+1.6%/+1.6%)	**0.203** (+5.7%/+0.5%)	0.466^{*} (+8.9%/−1.1%)
	INEX-LD			
	MAP	**P@10**	**P@20**	**b-pref**
LM [2]	0.106 (+3.9%/−9.4%)	0.236 (−0.8%/−8.5%)	0.188 (−1.1%/−5.5%)	$0.290_{\dagger}$ (−8.8%/−13.4%)
BM25 [2]	**0.118** (+15.7%/+0.9%)	0.247 (+3.8%/−4.3%)	0.204 (+7.4%/+2.5%)	$0.309_{\dagger}$ (−2.8%/−7.8%)
MLM-tc [2]	$0.104_{\dagger}$ (+2.0%/−11.1%)	$0.232_{\dagger}$ (−2.5%/−10.1%)	$0.197_{\dagger}$ (+3.7%/−1.0%)	$0.288_{\dagger}$ (−9.4%/−14.0%)
BM25F-tc [2]	0.117 (+14.7%/0.0%)	0.249 (+4.6%/−3.5%)	0.200 (+5.3%/+0.5%)	$0.304_{\dagger}$ (−4.4%/−9.3%)
PRMS [2]	$0.084^{*}_{\dagger}$ (−17.6%/−28.2%)	$0.203^{*}_{\dagger}$ (−14.7%/−21.3%)	$0.163^{*}_{\dagger}$ (−14.2%/−18.1%)	$0.256^{*}_{\dagger}$ (−19.5%/−23.6%)
PRMS	$0.064^{*}_{\dagger}$ (−37.3%/−45.3%)	$0.145^{*}_{\dagger}$ (−39.1%/−43.8%)	$0.123^{*}_{\dagger}$ (−35.3%/−38.2%)	$0.216^{*}_{\dagger}$ (−32.1%/−35.5%)
MLM-CA	0.102	0.238	0.190	0.318
SDM-CA	0.117^{*} (+14.7%)	0.258 (+8.4%)	0.199 (+4.7%)	0.335 (+5.3%)
FSDM	0.111^{*} (+8.8%/−5.1%)	$\mathbf{0.263}^{*}$ (+10.5%/+1.9%)	$\mathbf{0.215}^{*}_{\dagger}$ (+13.2%/+8.0%)	$\mathbf{0.341}^{*}$ (+7.2%/+1.8%)
	QALD-2			
	MAP	**P@10**	**P@20**	**b-pref**
LM [2]	$0.107^{*}_{\dagger}$ (−29.6%/−41.8%)	$0.051^{*}_{\dagger}$ (−50.5%/−51.9%)	$0.042^{*}_{\dagger}$ (−50.0%/−53.3%)	$0.233^{*}_{\dagger}$ (−37.5%/−49.9%)
BM25 [2]	$0.118_{\dagger}$ (−22.4%/−35.9%)	$0.066^{*}_{\dagger}$ (−35.9%/−37.7%)	$0.053^{*}_{\dagger}$ (−36.9%/−41.1%)	$0.311^{*}_{\dagger}$ (−16.6%/−33.1%)
MLM-tc [2]	$0.099^{*}_{\dagger}$ (−34.9%/−46.2%)	$0.051^{*}_{\dagger}$ (−50.5%/−51.9%)	$0.037^{*}_{\dagger}$ (−56.0%/−58.9%)	$0.239^{*}_{\dagger}$ (−35.9%/−48.6%)
BM25F-tc [2]	$0.107^{*}_{\dagger}$ (−29.6%/−41.8%)	$0.062^{*}_{\dagger}$ (−39.8%/−41.5%)	$0.049^{*}_{\dagger}$ (−41.7%/−45.6%)	$0.292^{*}_{\dagger}$ (−21.7%/−37.2%)
PRMS [2]	$0.105^{*}_{\dagger}$ (−30.9%/−42.9%)	$0.069^{*}_{\dagger}$ (−33.0%/−34.9%)	$0.051^{*}_{\dagger}$ (−39.3%/−43.3%)	$0.260^{*}_{\dagger}$ (−30.3%/−44.1%)
PRMS	$0.120^{*}_{\dagger}$ (−21.1%/−34.8%)	$0.079^{*}_{\dagger}$ (−23.3%/−25.5%)	$0.067^{*}_{\dagger}$ (−20.2%/−25.6%)	$0.328_{\dagger}$ (−12.1%/−29.5%)
MLM-CA	0.152	0.103	0.084	0.373
SDM-CA	0.184 (+21.1%)	0.106 (+2.9%)	0.090 (+7.1%)	0.465^{*} (+24.7%)
FSDM	$\mathbf{0.195}^{*}$ (+28.3%/+6.0%)	$\mathbf{0.136}^{*}_{\dagger}$ (+32.0%/+28.3%)	$\mathbf{0.111}^{*}$ (+32.1%/+23.3%)	$\mathbf{0.466}^{*}$ (+24.9%/+0.2%)
	All queries			
	MAP	**P@10**	**P@20**	**b-pref**
LM [2]	$0.175^{*}_{\dagger}$ (−10.7%/−8.9%)	$0.182^{*}_{\dagger}$ (−11.7%/−8.1%)	$0.139^{*}_{\dagger}$ (−11.5%/−10.3%)	$0.398^{*}_{\dagger}$ (−12.5%/−19.6%)
BM25 [2]	0.186 (−5.1%/−3.1%)	0.194 (−5.8%/−2.0%)	0.149 (−5.1%/−3.9%)	$0.428^{*}_{\dagger}$ (−5.9%/−13.5%)
MLM-tc [2]	0.181 (−7.7%/−5.7%)	$0.185^{*}_{\dagger}$ (−10.2%/−6.6%)	$0.143^{*}_{\dagger}$ (−8.9%/−7.7%)	$0.409^{*}_{\dagger}$ (−10.1%/−17.4%)
BM25F-tc [2]	0.182 (−7.1%/−5.2%)	0.192^{*} (−6.8%/−3.0%)	$0.145^{*}_{\dagger}$ (−7.6%/−6.5%)	$0.421^{*}_{\dagger}$ (−7.5%/−14.9%)
PRMS [2]	$0.176^{*}_{\dagger}$ (−10.2%/−8.3%)	0.186^{*} (−9.7%/−6.1%)	$0.143^{*}_{\dagger}$ (−8.9%/−7.7%)	$0.400^{*}_{\dagger}$ (−12.1%/−19.2%)
PRMS	$0.136^{*}_{\dagger}$ (−30.6%/−29.2%)	$0.136^{*}_{\dagger}$ (−34.0%/−31.3%)	$0.107^{*}_{\dagger}$ (−31.8%/−31.0%)	$0.365^{*}_{\dagger}$ (−19.8%/−26.3%)
MLM-CA	0.196	0.206	0.157	0.455
SDM-CA	0.192 (−2.0%)	0.198 (−3.9%)	0.155 (−1.3%)	0.495^{*} (+8.8%)
FSDM	$\mathbf{0.231}^{*}_{\dagger}$ (+17.9%/+20.3%)	$\mathbf{0.231}^{*}_{\dagger}$ (+12.1%/+16.7%)	$\mathbf{0.179}^{*}_{\dagger}$ (+14.0%/+15.5%)	$\mathbf{0.517}^{*}_{\dagger}$ (+13.6%/+4.4%)

We also observe that FSDM significantly outperforms the MLM-CA baseline on all query sets and all metrics, except ListSearch. FSDM also significantly outperforms SDM-CA on SemSearch ES and the entire query set with respect to all evaluation metrics. On the remaining query sets, FSDM is more effective than SDM in all but two cases (on INEX-LD with respect of MAP and on ListSearch with respect to b-pref; however, the difference in both cases is not significant), including statistically significant improvement on INEX-LD with respect to P@20 and on QALD-2 with respect to P@10.

4.4 Success/Failure Analysis

In this section, we present the results of quantitative and qualitative analysis of errors in the search results. First, Figure 3 illustrates the per-topic differences in average precision between FSDM and SDM. From Figure 3, it follows that, on the entire query set, FSDM performs better than SDM on a larger number of topics than vice versa, with the most significant difference on SemSearch ES query set. QALD-2 has the largest number of queries with no performance differences, since both FSDM and SDM fail to find any relevant results for 28 out of 140 queries from this fairly difficult query set.

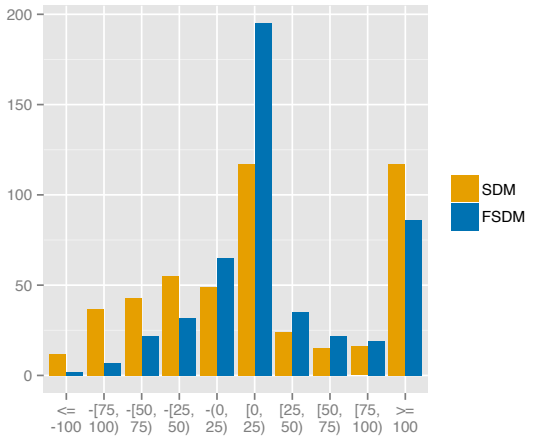


Figure 4: **Robustness of SDM and FSDM methods with respect to the MLM-CA baseline over the entire query set.**

As the next experiment, we analyze the robustness of SDM and FSDM compared to the MLM-CA baseline on the entire query set. Figure 4 shows the histogram for various ranges of relative decreases and increases in MAP with respect to MLM-CA and indicates that FSDM is more robust compared to SDM. In particular, it improves the performance of 50% of the queries with respect to MLM-CA, compared to 45% of the queries improved by SDM. At the same time, FSDM decreases the performance of only 26% of the queries, while SDM degrades the performance of 40% of the queries.

Next, we focus on a more detailed qualitative analysis of queries showing the highest relative gain in terms of MAP. In particular, we observe that a common mistake of SDM on named entity queries is overestimation of importance of matches in the fields other than *names*. For example, for the query SemSearch_ES-22 *"city of charlotte"*, SDM mistakenly promotes the entities, such as *Anthony Foxx*, a former mayor of Charlotte, or *Clayton Heafner*, an American golfer, who lived in this city. Therefore, using SDM for type queries may result in a topic drift and decreased precision. For example, for the query QALD2_tr-89 *"give me all soccer clubs in the premier league"*, SDM ranks higher the soccer clubs from the premier leagues in Tasmania (Northern Rangers) and New Zealand (Metro F.C.). Similar effect is observed for the query INEX_XER-121*"us presidents since 1960"*, for which SDM promotes the entities with matched bigrams and unigrams in the fields of minor importance, such as *List of people on stamps of Liberia*, whereas FSDM correctly emphasizes term matches in categories and ranks the correct results, such as *Gerald Ford*, *Theodore Roosevelt*, and *Ronald Reagan* at the top.

We also observed that a common cause of many FSDM failures is neglecting the important query terms. For example, for the TREC_Entity-9 query *"members of the beaux arts trio"*, FSDM mistakenly promotes the entities *Emmanuel Pontremoli* and *Pierre Carron*, the members of the Académie des Beaux Arts, which is caused by matching bigrams *members beaux* and *beaux arts* in the *categories* field. However, the query term *trio* provides an important clue that the query intent is about Beaux Arts Trio, a famous piano trio, which FSDM is unable to pick up. Similar reasoning applies to the QALD2_tr-15 query *"who created goofy"*, for which FSDM drifts to cartoons rather than information about the actual character creator, and the QALD2_te-90 query *"where is the residence of the prime minister of spain?"*, for which FSDM promotes the Spanish prime ministers in retrieval results, instead of the exact answer to the question.

The reason of FSDM failure on another difficult query SemSearch_LS-10 *"did nicole kidman have any siblings"* is slightly different. The most precise answer, *Antonia Kidman*, who is the younger sister of the actress Nicole Kidman, does not contain any occurrences of the query term *sibling*. In this case, SDM ranks higher a DBpedia disambiguation page that mentions the right entity. At the same time, FSDM tends to return the movies starring Nicole Kidman as the top results. We hypothesize that query expansion with synonyms can potentially resolve these issues, however we leave verification of this hypothesis to future work.

Table 5: **Comparison of SDM and FSDM on queries of various difficulty (according to the number of positive relevance judgments). "†" indicates statistical significance with respect to the Fisher's randomization test ($\alpha = 0.05$) [30].**

	Difficult queries			
	MAP	**P@10**	**P@20**	**b-pref**
SDM	0.213	**0.067**	0.042	0.599
FSDM	**0.239**	0.065	**0.043**	**0.621**
	Medium queries			
	MAP	**P@10**	**P@20**	**b-pref**
SDM	0.209	0.224	0.165	0.532
FSDM	**0.264**$_\dagger$	**0.272**$_\dagger$	**0.191**$_\dagger$	**0.559**$_\dagger$
	Easy queries			
	MAP	**P@10**	**P@20**	**b-pref**
SDM	0.139	0.298	0.262	0.316
FSDM	**0.166**$_\dagger$	**0.345**$_\dagger$	**0.309**$_\dagger$	**0.330**

To complete the analysis of retrieval performance of FSDM, Table 5 compares the performance of SDM and FSDM on queries of various levels of difficulty. To obtain the results

in Table 5, we grouped all the queries into three categories: difficult queries (that have 3 or less positive relevance judgments), medium queries (with 4 to 20, i.e., potential first two SERPs) and easy queries (that have more than 20 positive relevance judgments associated with them). As a result, we obtained 141, 216, and 128 queries in each category, respectively. From Table 5 it follows that FSDM outperforms SDM in each query category with respect to all metrics in all the cases but one (P@10 for difficult queries), with the most significant difference in performance on medium and easy queries. We, therefore, conclude that creating sophisticated entity descriptions is not sufficient for answering *difficult queries* in entity retrieval scenario and better capturing the semantics of query terms is required to further improve the precision of FSDM for difficult queries.

5. CONCLUSION

This paper proposed Fielded Sequential Dependence Model, a novel retrieval model, which incorporates term dependencies into structured document retrieval, and a two-stage algorithm to directly optimize the parameters of this model with respect to the target retrieval metric. Although we only experimented with ERWD, FSDM can be applied to retrieval from collections of structured documents of any type.

We demonstrated that having different field weighting schemes for unigrams and bigrams is effective for different types of queries in ad-hoc entity retrieval scenario. Experimental evaluation of FSDM on a standard publicly available benchmark showed that it consistently and, in most cases, statistically significantly outperforms state-of-the-art structured and unstructured retrieval models for ERWD.

Acknowledgments

This work was partially supported by the subsidy from the government of the Russian Federation to support the program of competitive growth of Kazan Federal University among world class academic centers and universities and by the Russian Foundation for Basic Research (grants # 15-07-08522, 15-47-02472).

6. REFERENCES

[1] K. Balog, M. Bron, and M. D. Rijke. Query Modeling for Entity Search based on Terms, Categories, and Examples. *ACM TOIS*, 29:22, 2011.

[2] K. Balog and R. Neumayer. A Test Collection for Entity Search in DBpedia. In *Proceedings of the 36th ACM SIGIR*, pages 737–740, 2013.

[3] M. Bendersky, D. Metzler, and W. B. Croft. Learning Concept Importance Using a Weighted Dependence Model. In *Proceedings of the 3rd ACM WSDM*, pages 31–40, 2010.

[4] R. Blanco, P. Mika, and S. Vigna. Effective and Efficient Entity Search in RDF Data. In *Proceedings of the 10th ISWC*, pages 83–97, 2011.

[5] M. Bron, K. Balog, and M. de Rijke. Example Based Entity Search in the Web of Data. In *Proceedings of the 35th ECIR*, pages 392–403, 2013.

[6] M. Ciglan, K. Nørvåg, and L. Hluchý. The SemSets Model for Ad-hoc Semantic List Search. In *Proceedings of the 21st WWW*, pages 131–140, 2012.

[7] J. Dalton, L. Dietz, and J. Allan. Entity Query Feature Expansion Using Knowledge Base Links. In *Proceedings of the 37th ACM SIGIR*, pages 365–374, 2014.

[8] S. Elbassuoni and R. Blanco. Keyword Search over RDF Graphs. In *Proceedings of the 20th ACM CIKM*, pages 237–242, 2011.

[9] S. Elbassuoni, M. Ramanath, R. Schenkel, M. Sydow, and G. Weikum. Language-model-based Ranking for Queries on RDF-graphs. In *Proceedings of the 18th ACM CIKM*, pages 977–986, 2009.

[10] K. M. Elbedweihy, S. N. Wrigley, P. Clough, and F. Ciravegna. An Overview of Semantic Search Evaluation Initiatives. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2014.

[11] D. M. Herzig, P. Mika, R. Blanco, and T. Tran. Federated Entity Search Using On-the-Fly Consolidation. In *Proceedings of the 12th ISWC*, pages 167–183. 2013.

[12] S. Huston and W. B. Croft. A Comparison of Retrieval Models using Term Dependencies. In *Proceedings of the 23rd ACM CIKM*, pages 111–120, 2014.

[13] J. Y. Kim and W. B. Croft. A Field Relevance Model for Structured Document Retrieval. In *Proceedings of the 34th ECIR*, pages 97–108, 2012.

[14] J. Y. Kim, X. Xue, and W. B. Croft. A Probabilistic Retrieval Model for Semistructured Data. In *Proceedings of the 31st ECIR*, pages 228–239, 2009.

[15] A. Kotov and C. Zhai. Tapping into Knowledge Base for Concept Feedback: Leveraging ConceptNet to Improve Search Results for Difficult Queries. In *Proceedings of the 5th WSDM*, pages 403–412, 2012.

[16] A. Kotov, C. Zhai, and R. Sproat. Mining Named Entities with Temporally Correlated Bursts from Multilingual Web News Streams. In *Proceedings of the 4th ACM WSDM*, pages 237–246, 2011.

[17] C. L. Koumenides and N. R. Shadbolt. Ranking Methods for Entity-Oriented Semantic Web Search. *JASIST*, 65(6):1091–1106, 2014.

[18] D. Metzler and W. B. Croft. A Markov Random Field Model for Term Dependencies. In *Proceedings of the 28th ACM SIGIR*, pages 472–479, 2005.

[19] D. Metzler and W. B. Croft. Linear Feature-based Models for Information Retrieval. *Information Retrieval*, 10:257–274, 2007.

[20] R. Neumayer, K. Balog, and K. Nørvåg. On the Modeling of Entities for Ad-hoc Entity Search in the Web of Data. In *Proceedings of the 34th ECIR*, pages 133–145, 2012.

[21] R. Neumayer, K. Balog, and K. Nørvåg. When Simple is (more than) Good Enough: Effective Semantic Search with (almost) no Semantics. In *Proceedings of the 34th ECIR*, pages 540–543, 2012.

[22] Z. Nie, Y. Ma, S. Shi, J.-R. Wen, and W.-Y. Ma. Web Object Retrieval. In *Proceedings of the 16th WWW*, pages 81–90, 2007.

[23] P. Ogilvie and J. Callan. Combining Document Representations for Known-item Search. In *Proceedings of the 26th ACM SIGIR*, pages 143–150, 2003.

[24] J. R. Pérez-Aguera, J. Arroyo, J. Greenberg, J. P. Iglesias, and V. Fresno. Using BM25F for Semantic Search. In *Proceedings of the 3rd SemSearch Workshop*, 2010.

[25] J. M. Ponte and W. B. Croft. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st ACM SIGIR*, pages 275–281, 1998.

[26] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc Object Retrieval in the Web of Data. In *Proceedings of the 19th WWW*, pages 771–780, 2010.

[27] A. J. Roa-Valverde and S. Miguel-Angel. A Survey of Approaches for Ranking on the Web of Data. *Information Retrieval*, 17(4):295–325, 2014.

[28] S. Robertson, H. Zaragoza, and M. Taylor. Simple BM25 Extension to Multiple Weighted Fields. In *Proceedings of the 13th ACM CIKM*, pages 42–49, 2004.

[29] S. Shakarpour, A.-C. N. Ngomo, and S. Auer. Question Answering on Interlinked Data. In *Proceedings of the 22nd WWW*, pages 1145–1156, 2013.

[30] M. D. Smucker, J. Allan, and B. Carterette. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *Proceedings of the 16th ACM CIKM*, pages 623–632, 2007.

[31] A. Tonon, G. Demartini, and P. Cudré-Mauroux. Combining Inverted Indices and Structured Search for Ad-hoc Object Retrieval. In *Proceedings of the 35th ACM SIGIR*, pages 125–134, 2012.

[32] M. Yahya, K. Berberich, S. Elbassuoni, and G. Weikum. Robust Question Answering over the Web of Linked Data. In *Proceedings of the 22nd ACM CIKM*, pages 1107–1116, 2013.

[33] N. Zhiltsov and E. Agichtein. Improving Entity Search over Linked Data by Modeling Latent Semantics. In *Proceedings of the 22nd ACM CIKM*, pages 1253–1256, 2013.