Complementary Document Representations for Information Retrieval

Sylvia Melzer  
University of Hamburg  
Centre for the Study of Manuscript Cultures  
Warburgstraße 26  
20354 Hamburg

Simon Schiff  
University of Lübeck  
Institute of Information Systems  
Ratzeburger Allee 160  
23562 Lübeck

Ralf Möller  
University of Lübeck  
Institute of Information Systems  
Ratzeburger Allee 160  
23562 Lübeck

Abstract

In this paper, we present an approach for combining different document representations to support retrieval systems to deliver similar documents from different views.

Introduction

In content management systems, content is stored, organized, and supplemented with high-level content descriptions. Among simple data for authors, characters, publishers, and so on, nowadays, content descriptions contain feature-based (vector-based) as well as symbolic content descriptions, which, for instance, can be represented via logic-based techniques (Kaya 2011). Applications exploit symbolic content descriptions in various ways (see e.g. also systems such as OpenIE, OpenCalais). For example, in the semantic web, content descriptions are used to find documents, images, videos, or people. Search requests are specified by posing queries in query languages typically based on string patterns.

Until now, large-scale information retrieval processes are rarely based on symbolic content descriptions to match queries with content (Voorhees and Harman 2005). Google’s Knowledge Vault (KV) uses symbolic descriptions to support users in creating useful follow-up queries (Dong et al. 2014). It is also possible that so called holistic content descriptions (e.g., TF.IDF matrices) and corresponding similarity measures are used for query answering (Salton, Wong, and Yang 1975; Manning, Raghavan, and Schütze 2008). Matches on holistic content descriptions can be realized efficiently, e.g., by utilizing nearest-neighbor algorithms. These algorithms provide an efficient way to search for information that is similar to the given query, but limits their effectiveness in finding more information from different views (Ma and Tanaka 2005). For example, sports reporters writing an article about the city of “London” may also be interested in obtaining additional information about events in London while writing this article.

Holistic Representations

In the context of statistical relational learning many research contributions present a highly specialized scientific background for combined information retrieval (IR), e.g., for learning from low-dimensional embeddings (Mikolov et al. 2013), to identify a set of plausible formulas from knowledge bases (Wang, Mazaitis, and Cohen 2014), as well as learning latent and distributional representations of Horn clauses to enhance logic-based completion for large datasets (Wang and Cohen 2016) by using a scalable probabilistic logic called ProPPR (Wang, Mazaitis, and Cohen 2013) to built intelligent IR systems, dealing with the uncertainty of content representations. Towards a combined IR a standard boolean model was developed (Salton, Fox, and Wu 1983). Nevertheless the potential of complementarity-based IR is far from exhausted. In (Melzer 2018) an algorithm for combining holistic and symbolic representations is presented, which delivers complement (additional) documents with high recall and precision. Due to high-complexity of the underlying logic, the performance of this system is not yet mature, however.

In this paper, we use standard latent semantic indexing (LSI) (Deerwester et al. 1990) for the holistic representation of documents in order to be able to focus mainly on the new complementary approach. LSI is recapitulated as follows.

In LSI, a document corpus is represented with a term-document matrix. This matrix is analyzed by singular value decomposition (SVD) in order to derive a latent semantic structure. The latent semantic structure is represented by the computed singular vectors and singular values. Each document and query is represented by a vector of $M$ feature values (terms), respectively. The “meaning” of a document or query can be approximated by $k$ values ($k < M$) or, to put it in other words, by the location of vectors in a $k$-dimensional space. This so called latent space is the result of the LSI approach, which computes an approximation of the original space. In general, the SVD for the term-document matrix $C$ is a $M \times N$ matrix and defined as $C = U \Sigma V^T$, where the matrices $U$, $\Sigma$, and $V$ are all of full rank. The idea of

---

*This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC 2176 ‘Understanding Written Artefacts: Material, Interaction and Transmission in Manuscript Cultures’, project no. 390893796.
Copyright © 2021 by the authors. All rights reserved.
LSI is to compute a rank-$k$ approximation of low error. This is accomplished by considering only the highest $k$ singular values from $\Sigma$. The new matrix is called $\Sigma_k$ and we get:

$$C_k = U_k \Sigma_k V_k^T,$$

where $U_k$ is an $M \times k$ matrix, $\Sigma_k$ is a $k \times k$ matrix, and $V_k^T$ is a $k \times N$ matrix (for details see (Deerwester et al. 1990)).

The rank of $C_k$ is at most $k$. This follows from the fact that $\Sigma_k$ has at most $k$ non-zero values. In LSI, singular value decomposition is used to construct a low-rank approximation $C_k$ of the term-document matrix $C$. The value $k$ is smaller than the original rank of $C$. The matrix $C_k$ is the best rank-$k$ approximation of the original matrix $C$ because the distance $\Delta$ between these two matrices by the 2-norm is minimized:

$$\Delta = ||C - C_k||_2.$$

A low-rank approximation of $C$ yields a new representation for the set of documents in a repository. Queries can also be represented using the low-rank approximation. In this context the process of computing query-document similarity scores is known as LSI. From Equation 1 we derive the holistic repository representation with documents as column vectors in latent space as:

$$H := V_k^T$$

(2)

Note, to fold-in a new $M \times 1$ document vector, $d$, into an existing LSI model, a projection, $d$, of $d$ onto the span of the current term vectors (columns of $U_k$) is computed by $d = \sum_k u_k^T d$. Analogously, to fold-in a new $1 \times N$ term vector, $t$, into an existing LSI model, a projection, $t$, of $t$ onto the span of the current document vectors (columns of $H$) is computed by $t = \sum_k v_k^T t^T$. A string query is represented by a query vector $\tilde{q} = (q_1, q_2, \ldots, q_m)^T$, where the values $q_1, q_2, \ldots, q_m$ are either 0 or 1. If a string is equal to a term, the value is 1, and 0 otherwise. The vector $\tilde{q}$ is mapped into its representation in the LSI space via the following equation:

$$\tilde{q}_k = \sum_k u_k^T \tilde{q}$$

(3)

Given the cosines between query $\tilde{q}_k$ and document vectors from $H$, the documents with the $t$ largest cosine values are selected as a query result. Locality-sensitive hashing (LSH) is another approach to compute the similarity more efficiently. Additionally, other embedding approaches could also be suitable to compute $H$ instead of LSI.

**Entity Type Similarity Representation**

The notable feature of this paper is to use complementary document representations for information retrieval to retrieve documents which are not just similar to others, but also provides additional information w.r.t. specific entity types in which a user could be interested in.

Therefore we use the holistic representation $H$ and a so-called entity type similarity representation of documents. Both document representations are in a way complementary to each other.

The entity type similarity matrix $H'$ is an $N \times N$ matrix, where $N$ is the number of documents $d_1, \ldots, d_N$ in the repository. $H'$ is defined as:

$$H' := \begin{pmatrix} h'_{d_1,d_1} & h'_{d_1,d_2} & \cdots & h'_{d_1,d_N} \\ h'_{d_2,d_1} & h'_{d_2,d_2} & \cdots & h'_{d_2,d_N} \\ \vdots & \vdots & \ddots & \vdots \\ h'_{d_N,d_1} & h'_{d_N,d_2} & \cdots & h'_{d_N,d_N} \end{pmatrix},$$

where $h'_{d_i,d_j} := \text{normalized number of the same entity types between } d_i \text{ and } d_j \text{ with } 1 \leq i, j \leq N$. The Jaccard similarity is used to formally define $h'_{d_i,d_j}$, where the function $\text{repr}$ denote the set of entity types associated with a document:

$$h'_{d_i,d_j} = \frac{(\text{repr}(d_i) \cap \text{repr}(d_j))}{(\text{repr}(d_i) \cup \text{repr}(d_j))}$$

(4)

In $H'$ the diagonal values are the maximum because the documents are compared with themselves. Large values represent high similarity/low complementarity and small values represent low similarity/high complementarity. A document contains text. Every word in a text belongs to an entity type. These entity types are hidden and can be obtained through various techniques. In these methods hidden entity types are identified, however, the entity types can be only computed with a set of documents as input.

To identify named entities in a text and classifies them into predefined categories (entity types) automatically, natural language processing (NLP) techniques such as named entity recognition (NER) can be used to automatically extract key information from texts, or merely use it to collect important information to store in a repository. Entity types can be “Person”, “Organization”, and “Location.” Entities can be names of persons, organizations, or locations. The Open Information Extraction (OpenIE) annotator (Manning et al. 2014) can assign the entities, mentioned in the document, to the entity types in which users are mainly interested in. In this paper, OpenIE is used to identify the entity types for each document in an offline setting, and the matrix $H'$ is computed offline as well.

**Complementarity-based Information Retrieval**

Suppose a repository $R$ is represented by a set of documents $Docs = \{d_1, \ldots, d_n\}$, by a holistic representation $H$, by an entity type similarity representation ($H'$) associated with each document $doc$, formally:

$$R := (Docs, H, H').$$

(5)

With respect to a repository of this kind, an online query answering problem $QA$ for retrieving relevant documents is defined as:

$$QA(Q, R, \theta),$$

(6)

where $Q$ is a query vector and $\theta$ is a threshold value. The algorithm for *holistic* IR is defined in Algorithm 1.

A holistic document representation $V_k^H$ is computed as an approximation of a term-document matrix $C$ by one of lower rank $k$ using the SVD. The document representation $H := V_k^H$ with $V_k^H = (doc_1, \ldots, doc_N)$ is the representation for each document in the collection. Queries will also
be cast into the same low-rank representation which are represented by \( \vec{q} \). The documents Docs, their holistic representations presented with \( H \), and the query vector \( \vec{q} \) are input parameters of Algorithm 1. With QA_{hol} the similarity scores between query and document representations are computed, e.g. with the cosine similarity or LSH for better performance. If the query vector \( \vec{q} \) and the document representation \( H[i] \) have a small distance, i.e. the predicate \( \text{sim}(\vec{q}, H[i]) \geq \theta \), where \( \theta \) is a threshold, then the associated document Docs[i] of \( H[i] \) is in the result set docs.

**Algorithm 2** The EntityQuery algorithm.

\[
\text{QA}_{\text{ent}}(\vec{q}, (\text{Docs, } H'), \theta): \\
\text{docs} := \emptyset \\
\text{for } i = 1 \text{ to } N \text{ do} \\
\quad \text{if } \text{sim}(\vec{q}, H'[i]) \geq \theta \text{ then} \\
\quad\quad \text{docs} := docs \cup \{(H'[i], \text{Docs}[i])\} \\
\text{end if} \\
\text{end for} \\
\text{return docs}
\]

The algorithm for entity-type-based IR is defined in Algorithm 2. A query vector \( \vec{q} \), documents Docs, and entity type similarity representations of the documents \( H' \) are input parameters of the Algorithm 2. Algorithm 2 computes the query-document similarity scores. If the query vector \( \vec{q} \) and the document representation \( H'[i] \) have a small distance, i.e. the predicate \( \text{sim}(\vec{q}, H'[i]) \geq \theta \), where \( \theta \) is a threshold, then the associated document Docs[i] of \( H'[i] \) is in the result set docs.

In the following, we present a new algorithm called Compl-IR Algorithm (Algorithm 3) to receive similar documents more efficiently than presented in (Melzer 2018).

**Algorithm 3** Compl-IR Algorithm.

\[
\text{Compl-IR}(\vec{q}_k, (\text{Docs, } H, H'), (\theta_1, \theta_2)): \\
\text{docs'} := \text{QA}_{\text{hol}}(\vec{q}_k, (\text{Docs, } H, \), \theta_1) \\
\text{Queries} := \emptyset \\
\text{for } (\_ , i) \text{ in docs' do} \\
\quad \text{Queries} := \text{Queries} \cup \{H'[i]\} \\
\text{end for} \\
\text{qref} := \text{centroid}(\text{Queries}) \\
\text{docs}'' := \text{QA}_{\text{ent}}(\vec{q}_\text{ref}, (\text{Docs, } H'), \theta_2) \\
\text{return } (\text{docs'}, \text{docs}'')
\]

The Compl-IR Algorithm is a combination of Algorithm 1 and Algorithm 2. Algorithm 3 requires as input a user query \( \vec{q}_k \), a set of documents Docs, the holistic representation of documents \( H \), the entity type similarity matrix \( H' \), and the threshold values \( \theta_1 \) and \( \theta_2 \). Similar documents docs' are computed via QA_{hol}(\vec{q}_k, (Docs, H, \), \theta_1).

In order to execute QA_{ent} for receiving additional documents, a query vector is required. However, in the QA_{ent} algorithm, \( \vec{q}_k \) cannot be used as a query because the matrices \( H \) and \( H' \) of the algorithms 1 and 2 have different semantics. In order to keep the semantics, the LSI results are used as new queries and the required entity type similarity representation \( H' \) is used instead of the holistic representation. Therefore, the entity type similarity representation of each document in docs' are then identified using the entity type similarity matrix \( H' \).

In Algorithm 3, Queries contain the set of entity type similarity representation for each document in docs'. Then the centroid of Queries with \( q_{\text{ref}} := \text{centroid}(\text{Queries}) \) is computed which is used as a reference query \( q_{\text{ref}} \). It is also possible to use all queries as input queries. In this case, almost all documents from the repository will be returned. The retrieval of all documents is not the idea of complementarity-based IR. The reference query is a good choice to get not all, but some more relevant documents. Hence, we propose to use a reference query for receiving additional documents. As a result of Algorithm 3, it returns a tuple of documents (docs', docs'').

**Application and Results**

For visualizing the application results, we use a test repository consisting of about 30 documents \( (d_1 \ldots d_{30}) \) taken from the athletics domain to evaluate the Compl-IR Algorithm. We compute (offline) and use the respective holistic representation \( H \) of the repository in a 2-dimensional space \( (k=2) \). For the purpose to compute the values for the complementarity matrix, the classification of words to an entity type was computed with the OpenIE annotator. Therefore the available entity types “Person”, “Date”, “Organization”, and “Place” were used. The entity type similarity matrix is also computed offline.

In Figure 1 the entity type similarity representations of documents and of queries based on the values from the entity type similarity matrix, are illustrated. The entity type similarity representation of documents are presented as \( \times \), the queries derived from docs' are presented as \( \circ \), and the reference query (centroid of the queries \( X \)) is presented as \( \bullet \).

For the query “London” and a threshold \( \theta_1 = 0.94, \theta_2 = 0.999 \), QA_{hol} delivers six documents. In docs', 4 out of 6 documents contain the term “London.” All these documents are assigned to the same topic because a high similarity to each other was calculated via the other terms used in the document.

QA_{ent} delivers four other documents from the repository. These four documents in docs'' contain the term “London” and provide further information w.r.t. the date and location of an event in London. These additional documents could also be delivered by LSI by decreasing the threshold value, however, the false positive rate is then increased.

Summarized, in this example, the four entity types people, date, organization, and place were used to compute the entity
similarity matrix $H'$. Then, the new Compl-IR Algorithm delivers additional documents with lower false negative rate (67% lower), higher recall (97%) and higher precision (19.4%). The values still need to be verified by extensive testing and a larger data set. These tests should include common topic-based approaches as well as highlight the characteristics of extreme cases. Nevertheless, the results of the Compl-IR Algorithm indicate the potential that an increase in recall and precision can be expected through the usage of complementary document representations.

In (Melzer 2018) it has also already been shown that the applicability of the complementarity-based IR approach leads to the improvement of IR results. However, the approach described there required a huge computational power (the computation of $H$ takes hours). In this paper the entity types similarity matrix is calculated differently, and a better performance could be achieved (the computation of $H'$ takes seconds).

It is conceivable to create or extend the entity type similarity matrix depending on the user query. For performance, however, this means that some entity type similarity matrices are already kept so that the calculation does not have to be performed online at the expense of performance.

Conclusions

In this paper, we present a new algorithm, which is called Compl-IR algorithm. The new algorithm use complementary document representations for IR with higher recall and precision compared to the LSI approach. For the holistic part of the presented methodology we used LSI for computing latent structures of documents to receive similar documents from different viewpoints. Instead of the classical holistic document representation $H = V^T$, we define a new entity type similarity representation of documents $H'$. The query answering problem is solved in a way that documents with high recall and precision are determined.

References


