Toward Optimized Multimodal Concept Indexing

Navid Rekabsaz[1(✉)], Ralf Bierig[2], Mihai Lupu[1], and Allan Hanbury[1]

[1] Information and Software Engineering Group, Vienna University of Technology, 1040 Vienna, Austria
{rekabsaz,lupu,hanbury}@ifs.tuwien.ac.at
[2] School of Computing, National College of Ireland, Dublin 1, Ireland
ralf.bierig@ncirl.ie

Abstract. Information retrieval on the (social) web moves from a pure term-frequency-based approach to an enhanced method that includes conceptual multimodal features on a semantic level. In this paper, we present an approach for semantic-based keyword search and focus especially on its optimization to scale it to real-world sized collections in the social media domain. Furthermore, we present a faceted indexing framework and architecture that relates content to semantic concepts to be indexed and searched semantically. We study the use of textual concepts in a social media domain and observe a significant improvement from using a concept-based solution for keyword searching. We address the problem of time-complexity that is a critical issue for concept-based methods by focusing on optimization to enable larger and more real-world style applications.

Keywords: Semantic indexing · Concept · Social web · Word2Vec

1 Introduction

The past decade has witnessed the massive growth of the social web, the continued impact and expansion of the world wide web and the increasing importance and synergy of content modalities, such as text, images, videos, opinions, and other data. There are currently about 200 active social networks[1] that attract visitors in the range of the 100 s of millions each month. Online visitors spend considerable amounts of time on social network platforms where they constantly contribute, consume, and implicitly evaluate content. The Facebook community alone, with over 1.2 billion members, shares the impressive amount of 30 billion pieces of content every month [17]. The knowledge contained in these massive data networks is unprecedented and, when harvested, can be made useful for many applications. Although research has started to automatically mine information from these rich sources, the problem of knowledge extraction from multimedia content remains difficult. The main challenges are the heterogeneity

[1] http://en.wikipedia.org/wiki/List_of_social_networking_websites.

N.T. Nguyen et al. (Eds.): TCCI XXVI, LNCS 10190, pp. 144–161, 2017.
DOI: 10.1007/978-3-319-59268-8_7

of the data, the scalability of the processing methods and the reliability of their predictions.

In order to address these challenges in the social web domain, recent researches exploit the use of semantics in multimodal information retrieval and specially in image retrieval [12]. However, the focus resided on image processing and, so far, the methods used for text similarity for the purpose of multimodal retrieval are fairly mainstream [24]. In this work, we focus on semantic-based keyword search while specifically considering the optimization of the processing time, thus making our approach manageable in an information system.

This paper has four contributions. The *first* contribution presents a general investigation into semantic similarity matching for three types of information retrieval tasks: sentence paraphrasing, sentence-to-paragraph similarity matching and document matching. We applied a semantic similarity algorithm with a threshold filter to leverage its semantic preciseness. While discovering that the chosen threshold has a large effect on performance, we also found that only certain tasks benefit from such a parameterized approach. As the *second* contribution, we explored the effect of semantic similarity and optimization methods in text-based image retrieval in social media by applying Word2Vec [18] and Random Indexing (RI) [23]. This represents one possible form for a semantic concept index. As the *third* contribution we provide an optimization for these algorithms to allow them to scale to real-world collection sizes for more effective semantic-based keyword search on the (social) web. With an execution time that is about 40 times slower than standard TF-IDF in Solr, especially with longer documents, it is clear that optimization is paramount for allowing semantic search to become applicable and useful. We applied and evaluated two optimization techniques to contribute to this essential and important goal. The *fourth* contribution is a framework for integrating and evaluating algorithms and methods for semantic indexing and keyword search. It is designed as a combined faceted index for multimodal content collections, such as MediaEval Diverse Images [10,11]. The framework is based on a flexible document model and incorporates concepts as a semantic extension toward more generalized forms of information search that exceed the classic bag-of-words approach. The interlinked nature of these parts has the benefit of being flexible with respect to many kinds of multimodal and also multilingual documents. Each of these facets can be transformed into a semantic representation based on a dynamic and exchangable set of algorithms. The index itself is implemented effectively by using flexible facet indices that can be combined within a flexible document format based on the data at hand. The previous contributions additionally serve as an application use-case for this framework.

The following section describes the related work surrounding the domains of faceted, multi-modal and semantic indexing and search. In particular, we cover concept-based information retrieval. We describe our indexing architecture together with an application example of semantic index in Sects. 4 and 3. Focusing on questions of optimization, we explain two methods, followed by

discussion and comparison in Sect. 5. We summarize our findings in Sect. 6, and subsequently elaborate on a range of future plans.

2 Related Work

While different modalities often occur together in the same document (scientific paper, website, blog, etc.), search through these modalities is usually done for each modality in isolation. It is well known that combining information from multiple modalities assists in retrieval tasks. For instance, the results of the ImageCLEF campaign's photographic retrieval task have shown that combining image and text information results in better retrieval performance than text alone [19]. There are two fundamental approaches to fusing information from multiple modalities: early fusion and late fusion [8].

Late fusion is widely used, as it avoids working in a single fused feature space but, instead, fusing results by reordering them based on the scores from the individual systems. Clinchant et al. [4] propose and test a number of late fusion approaches involving the sum or product combination of weighted scores from text and image retrieval systems. Difficulties arise from

- weights that must be fixed in advance or that need to be learned from difficult to obtain training data
- modality weights that might be query dependent and
- weights that are sensitive to the IR system performance for the various modalities [8]

Separate queries are needed for each modality, so that for example to find a picture of a cat in a database of annotated images, one would need to provide a picture of a cat and text about the cat. There are ways of getting around this limitation, such as choosing the images for the top returned text documents as seeds in an image search [8], but these are generally ad-hoc.

With early fusion, a query would not have to contain elements from all modalities in the dataset. To continue the previous example, pictures of a cat could be found only with text input. Early fusion suffers from the problem that text tends to sparsely inhabit a large feature space, while non-text features have denser distributions in a small feature space. It is however possible to represent images sparsely in higher-dimensional feature spaces through the use of bags of 'visual words' [5] that are obtained by clustering local image features. The simplest approach to early fusion is to simply concatenate the feature vectors from different modalities. However, concatenated feature vectors become less distinctive, due to the curse of dimensionality [8], making this approach rather ineffective. A solution proposed by Magalhaes and Rüeger [16] is to transform the feature vectors to reduce the dimension of the text feature vectors and increase the dimension of the image feature vectors using the minimum description length (MDL) principle.

Textual features has been used in many multimodal retrieval systems. For instance, recently, Eskevich et al. [9] considered a wide range of text retrieval

methods in the context of multimodal search for medical data, while Sabetghadam et al. [22] used text features in a graph-based model to retrieve images from Wikipedia. However, these works do not particularly exploit text semantics.

In the text retrieval community, text semantics started with Latent Semantic Analysis/Indexing (LSA/LSI) [7], the pioneer approach that initiated a new trend in surface text analysis. LSA was also used for image retrieval [20], but the method's practicality is limited by efficiency and scalability issues caused by the high-dimensional matrices it operates on. Explicit Semantic Analysis (ESA) is one of the early alternatives, aimed at reducing the computational load [14]. However, unlike LSA, ESA relies on a pre-existing set of concepts, which may not always be available. Random Indexing (RI) [23] is another alternative to LSA/LSI that creates context vectors based on the occurrence of word contexts. It has the benefit of being incremental and operating with significantly less resources while producing similar inductive results as LSA/LSI and not relying on any pre-existing knowledge. Word2Vec [18] further expands this approach while being highly incremental and scalable. When trained on large datasets, it is also possible to capture many linguistic subtleties (e.g., similar relation between Italy and Rome in comparison to France and Paris) that allow basic arithmetic operations within the model. This, in principle, allows exploiting the implicit knowledge within corpora. All of these methods represent the words in vector spaces.

In order to compare the text semantic approaches, Baroni et al. [3] systematically evaluates a set of models with parameter settings across a wide range of lexical semantics tasks. They observe an overall better performance of state-of-the-art context-based models (e.g., Word2Vec) than the classic methods (e.g., LSA).

Approaching the text semantics, Liu et al. [15] introduced the Histogram for Textual Concepts (HTC) method to map tags to a concept dictionary. However, the method is reminiscent of ESA described above, and it was never evaluated for the purpose of text-based image retrieval.

3 Concept-Based Multimedia Retrieval

In this section, first we explain the architecture of our system for semantic indexing and keyword search.

We introduce a framework for multimodal concept and facet-based information retrieval and, in the scope of this paper, focus on the indexing component, particularly the semantic indexing features. The interaction between the components of the indexing framework is depicted in Fig. 1. These components represent the conceptual building blocks of the indexing architecture as part of the general framework. The figure presents the document model, the concept model and the indexing model with its individual document facets, such as text-, tag-, and image-typed content. We additionally depicts the information flow between these parts in a simplified form.

The *document model* defines a document that functions as the basic unit for content that is composed of facets. A facet is either a text, a tag or an

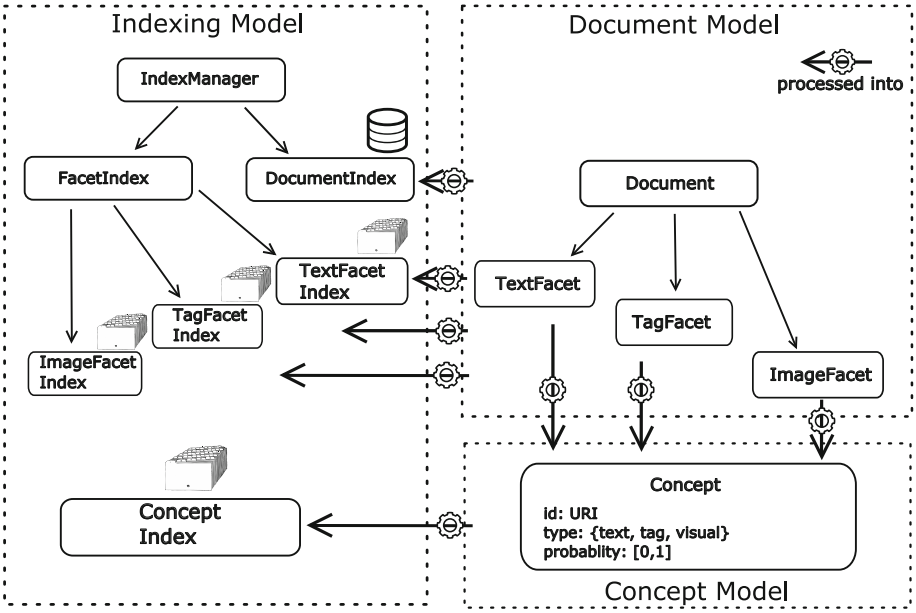

Fig. 1. Interaction between document model, concepts and (semantic) concept index

image. This allows many content structures to be created and organized, such as Wikipedia pages, scientific articles, websites, or blogs that often consist of such text, tag and image facets in various combinations. This structure also covers all unimodal variants, such as pure picture collections, since each document may contain any facet type in any order.

The *concept model* defines the structure of concepts. All concepts share a common identifier (usually a URI) that uniquely represents and differentiates them. A concept can describe either one of the three facet types, expressed as a type. That means, the concept can either be a text concept, a tag concept or a visual concept. Furthermore, a concept has a probability of being true, that allows a learning algorithm to store its confidence.

The *indexing model* is managed by the IndexManager, which controls the creation process of all indices, based on the configuration of the entire system. Facets are processed into respective indices that are all variations of a general FacetIndexer. TextFacets are indexed as a TextFacetIndex and TagFacets as a TagFacetIndex which are both based on Lucene[2] that stores it as separate, for their purpose optimized, inverted index file structures. ImageFacets are transformed into an ImageFacetIndex that is processed based on Lire, a Lucene derivative that is specialized on visual features. The indexing architecture therefore has three types of facet indexers, one per facet type, but maintains an arbitrary number of instances for each of them based on the structure of the

[2] http://lucene.apache.org/core.

content collection that is indexed. The DocumentIndex is a data structure that is implemented as a Database that connects all facets to make them accessible and usable for applications.

The *concept model* provides a definition of concepts for the framework. Concepts are processed into a ConceptIndex that is separate from the DocumentIndex and the FacetIndices. This concept model is used to translate facets into concepts. The ConceptIndex merges both text- and visual concepts into a common concept index space. In the next section, we demonstrate a first step into this direction by applying it solely on text concepts that are represented as an index of word vectors. Future work will expand on this by mapping concepts in an inverted index using Lucene covering both text, tag and visual concepts and representing it by a single index space.

In the following, we describe an application of semantic indexing in a social media domain. We specifically evaluate the effect of semantic-based retrieval on the textual features of multimodal documents.

4 Application of Concept-Based Retrieval

Based on the architecture discussed in the previous section, an application use-case is applied on the MediaEval Diverse Social Images task [10,11], using textual concepts. Our concept-based approach shows a significant improvement for keyword search on the test collection in the social media domain.

We explore the effect of semantic similarity and optimization methods on text-based image retrieval in social media as well as sentence paraphrasing and sentence-to-paragraph similarity. We introduce two semantic similarity methods, namely *Combinatorial* and *Greedy*, and evaluate them on the tasks using Word2Vec and Random Indexing word representations in different dimensions. This represents one possible scenario for a semantic concept index as shown in Fig. 1 and also examines the effectiveness of concept-based retrieval in this domain.

4.1 Experiment Setup

The evaluation on document retrieval was conducted using Flickr data, in particular in the framework of the MediaEval Retrieving Diverse Social Images Task 2013/2014 [10,11]. The task addresses result relevance and diversification in social image retrieval. We merged the datasets of 2013 (Div400) [11] and 2014 (Div150Cred) [10] and denoted it as MediaEval. It consists of about 60k photos of 300 world landmark locations (e.g., museums, monuments, churches, etc.). The provided data for each landmark location include a ranked list of photos together with their representative texts (title, description, and tags), Flickr's metadata, a Wikipedia article of the location and a user's credibility estimation (only for the 2014 edition). The name of each landmark location (e.g., Eiffel Tower) is used as the query for retrieving its related documents. For semantic text similarity, we focus on the relevance of the representative text of the photos containing title,

description, and tags. We removed HTML tags and decompounded the terms using a dictionary obtained from the whole corpus.

We consider the evaluation metric as the precision at a cutoff of 20 documents (P@20) which was also used in the official runs. In order to examine the results, a standard Solr index was used as the baseline (P@20 = 0.760). Statistical significant difference at $p = 0.05$ or lower against the baseline (denoted by† in the tables) was calculated using Fisher's two-sided paired randomization test. The two-sided paired randomization test examines the significance of the difference between two sets of data by calculating the difference of each pair of the datasets and then passing them to a more common significance test such as a one-sample t-test.

As mentioned before, in addition to image retrieval, we tested the methods on broader text-based information seeking tasks, namely sentence paraphrasing and sentence-to-paragraph similarity tasks. For sentence paraphrasing, we use SemEval 2014 Multilingual Semantic Textual Similarity - Task 10 [1] (SemEval Task 10), the English subtask. The goal of this task is to measure the semantic similarity of two sentences. The participating systems are compared by their mean Pearson correlation between the system output and a human-annotated gold standard. For sentence-to-paragraph similarity, we select the collection of SemEval 2014 Cross-Level Semantic Similarity - Task 3 [13] (SemEval Task 3), the paragraph to sentence subtask. The test collection contains 500 sentence-paragraph pairs. Similar to the Task 10, Pearson correlation is used as evaluation metrics. Table 1 summarises the tasks and test collections used in the experiments.

Table 1. Tasks and test-collections.

Task	Test collection	Evaluation metric	Collection size
sentence-to-sentence	SemEval 2014 STS - Task 10 [1]	Pearson correlation	3750 sentence pairs
sentence-to-paragraph	SemEval 2014 STS - Task 3 [13]	Pearson correlation	500 sentence-paragraph pairs
document retrieval	MediaEval 2013/2014 retrieving diverse social images [10,11]	P@20	309 topics - 60739 documents

We used the English Wikipedia text corpus to train our word representation models. For Word2Vec, we created models in 50, 100, 200, 300, 400, and 600 dimensions. We trained our Word2Vec word representation using Word2Vec toolkit[3] by applying CBOW approach of Mikolov et al. [18] with context

[3] https://code.google.com/p/word2vec/.

windows of 5 words and subsampling at $t = 1e^{-5}$. The Random Indexing word representations were trained using the Semantic Vectors package[4] with the default parameter settings of the package which considers the whole document as the context window. In all the models, we considered the words with frequency less than five as noise and filtered them out.

In the following, we define two text-to-text similarity methods and report and discuss the results of their evaluations on the mentioned tasks.

4.2 Combinatorial Method

The first algorithm, denoted as *SimCombi*, is based on the mean of words' similarity values. The algorithm first calculates the similarity of a given text (A) to another one (B) by simply aggregating the word-level similarity values that are greater than a given threshold (Algorithm 1). Then, to make the similarity symmetric, we repeat the same algorithm from B to A and return the mean of these two values as the similarity of the two texts. Although the algorithm is very simple, the choice of the best threshold is not obvious. By increasing the threshold, we remove more word pairs and therefore lose a part of the information. Decreasing the threshold adds more word pairs and therefore more noise to the calculation.

Algorithm 1: **SimCombi**

Input: text A and B, and threshold value t
Output: similarity of the text A to B
$meanList \leftarrow []$;
for $w \in A$ **do**
  $simList \leftarrow []$;
  for $v \in B$ **do**
    if $\cos(\boldsymbol{w}, \boldsymbol{v}) >= t$ **then**
      $simList \leftarrow simList + \cos(\boldsymbol{w}, \boldsymbol{v})$;
  $meanList \leftarrow meanList + \text{mean}(simList)$
return $\text{mean}(meanList)$;

The performance for the three test collections for varying thresholds between 0 and 1 in 0.05 increments are shown in Figs. 2, 3, and 4. The best achieved results of the tasks are shown in Table 2. For the sentence paraphrasing task, the most impressive result is that the best result achieved an average correlation of 0.71 as the best overall performance. This represents rank 11th out of the 38 submitted runs. However, all 10 runs above use a knowledge base and/or NLPwhich would not generalize to other domains or languages. For the

[4] https://code.google.com/p/semanticvectors/.

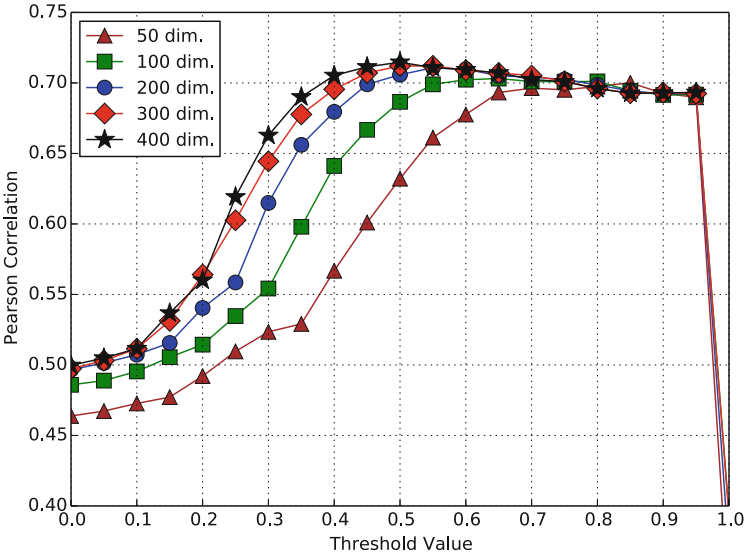

Fig. 2. Evaluation results of SemEval 2014 multilingual semantic textual similarity - Task 10 [1] English subtask

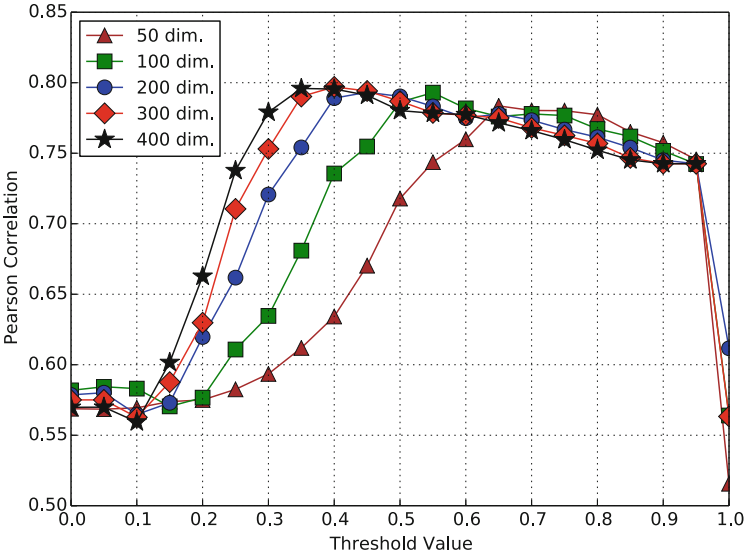

Fig. 3. Evaluation results of SemEval 2014 cross-level semantic similarity - Task 3 [13], paragraph to sentence subtask

MediaEval task, we observed that the best result of evaluating the SimCombi algorithm as a semantic-based similarity method outperforms the simple content-based approach.

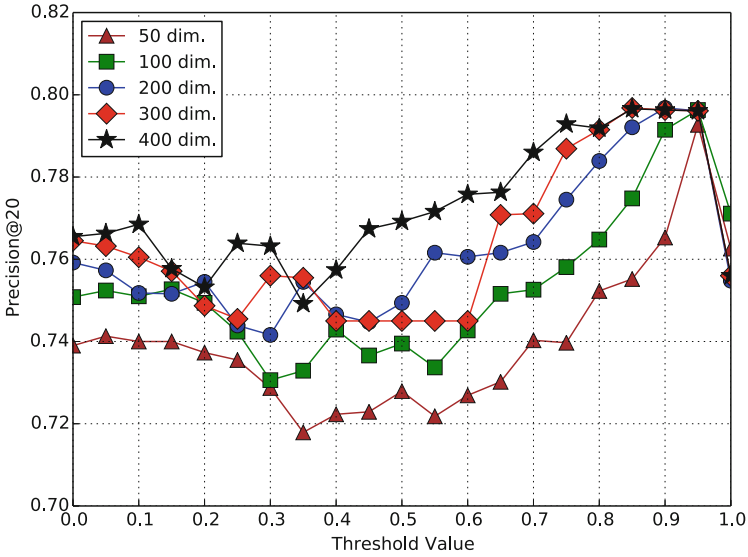

Fig. 4. Evaluation results of MediaEval 2013/2014 retrieving diverse social images Task [10,11]

As it is clear from the results, the choice of the threshold has an important effect on the effectiveness of the method. In order to effectively guess the parameters, for the SemEval tasks we can consider the following observations on the behavior patterns: (1) the performance is very low in smaller threshold values and increases steadily as the threshold increases until reaching a peak and then it slightly decreases. (2) Higher dimensions have overall better performance, while all the models finally converge. (3) The peak of performance is in lower similarity values for higher dimensions.

While the mentioned observations can be useful for parameter tuning of the SemEval tasks, it cannot be clearly extended to MediaEval (document retrieval). The reason could be due that the tasks are more complicated such that more factors confound the final performance. For example, the length of the documents are much more varied here than in the SemEval tasks.

In order to address the problem of parameter tuning in the SimCombi method, in the following we define a parameter-free semantic similarity method which shows very similar performance to the best results of the SimCombi method.

4.3 Greedy Method

The Greedy method, denoted *SimGreedy* [21], is inherited from the SimCombi method while applying a greedy approach in selecting the words. The algorithm measures the semantic-based text-to-text similarity by considering only the word with the highest similarity value in the B document to each word in the A

Table 2. Best results of mean Pearson correlation of SemEval 2014 Task 10 [1], SemEval 2014 Task 3 [13] paragraph to sentence subtask, and P@20 of MediaEval retrieving diverse social images Task 2013/2014 [10,11]. All the MediaEval results are significantly better than the Solr baseline

Dimension	SemEval task 3	SemEval task 10	MediaEval
50	0.783	0.700	0.792
100	0.792	0.703	0.796
200	0.793	0.710	0.796
300	0.797	0.712	0.796
400	0.795	0.714	0.796

document. The approach calculates the relatedness of document A to document B based on $SimGreedy(A, B)$ defined as follows:

$$SimGreedy(A, B) = \frac{\sum_{t \in A} idf(t) * maxSim(t, B)}{\sum_{t \in A} idf(t)} \quad (1)$$

where t represents a term of document A and $idf(t)$ is the Inverse Document Frequency of the term t. The function $maxSim$ calculates separately the cosine of the term t to each word in document B and returns the highest value. In this method, each word in the source document is aligned to the word in the target document to which it has the highest semantic similarity. Then, the results are aggregated based on the weight of each word to achieve the document-to-document similarity. $SimGreedy$ is defined as the average of $SimGreedy(A, B)$ and $SimGreedy(B, A)$. Considering n and m as the number of words in documents A and B respectively, the complexity of SimGreedy is of order $O(n * m)$.

We checked the effectiveness of SimGreedy by first evaluating the sentence paraphrasing (SemEval 2014 Task 10 [1]) and the paragraph to sentence similarity task (SemEval 2014 Task 3 [13]). Tables 3 and 4 show the mean Pearson correlations between the similarity methods and the gold standard. In both the tasks, SimGreedy exposes very similar results to the best performing results of SimCombi. It also appears that the effect of similarity method is more important than the number of dimensions of the vector representation such that after the dimension of 100 in both the tasks the results are very similar.

In the next step, we evaluated the SimGreedy method on MediaEval Retrieving Diverse Social Images Task 2013/2014 [10,11] as a more complicated task, shown in Table 5. We observed that using SimGreedy as a semantic-based similarity method outperforms the simple content-based approach while after the dimension of 100, its performance is very similar to the best results of the SimCombi method. Similar to the previous tasks, the number of dimensions does not have a significant effect on the result of the method.

In the following, we want to examine the effect of the word representation method on the performance of the semantic similarity method. To answer the question, we selected the models with 200 dimensions (as a generally good

Table 3. Mean Pearson correlation of SemEval 2014 Task 10 [1] using Word2Vec (W2V) [18] word representation

Dimension	SimGreedy	SimCombi (best)
50	0.697	0.700
100	0.707	0.703
200	0.712	0.710
300	0.713	0.712
400	0.714	0.714

Table 4. Mean Pearson correlation of SemEval 2014 Task 3 [13], paragraph to sentence subtask using Word2Vec (W2V) [18] word representation

Dimension	SimGreedy	SimCombi (best)
50	0.778	0.783
100	0.787	0.792
200	0.789	0.793
300	0.790	0.797
400	0.790	0.795

Table 5. MediaEval retrieving diverse social images Task 2013/2014 [10,11]. Models trained on Wikipedia using Word2Vec (W2V). The sign † denotes statistical significant difference

Dimension	SimGreedy	SimCombi (best)
50	0.766	†0.792
100	†0.787	†0.796
200	†0.795	†0.796
300	†**0.801**	†0.796
400	†0.799	†0.796
Solr (Baseline)		0.760

performance model) together with 600 as a much higher dimension and evaluated the MediaEval tasks on the models created with Word2Vec and Random Indexing methods. As shown in Table 6, Word2Vec shows slightly better results than Random Indexing while Random Indexing is still significantly better than the baseline. We can then conclude that the similarity method has more effect on the results than the number of dimensions or word representation method.

In order to compare the results with the participating systems in the task, we repeated the experiment on 2014 test dataset. As it is shown in Table 7 using SimGreedy and Word2Vec, we achieved the state-of-the-art result of 0.842 for

Table 6. MediaEval retrieving diverse social images Task 2013/2014 [10,11]. Models trained on Wikipedia using Random Indexing (RI) and Word2Vec (W2V). The sign † denotes statistical significant difference

Representation	Dimension	SimGreedy
Word2Vec	200	†**0.795**
Word2Vec	600	†0.793
Random Indexing	200	†0.788
Random Indexing	600	†0.787
Solr (Baseline)		0.760

P@20 between 41 runs including even the ones which used image features but not external resources.

Table 7. MediaEval retrieving diverse social images Task 2014 Results using query expansion. Models are trained on Wikipedia corpus with 200 and 600 dimensions. Our semantic-based approach only uses the textual features. *Best* indicates the state-of-the-art performing system in the 2014 task for different runs

Representation	Dimension	P@20
Word2Vec	200	0.833
Word2Vec	600	**0.842**
Random Indexing	200	0.813
Random Indexing	600	0.817
Best text (Run1)		0.832
Best text-visual (Run3)		0.817
Best all resources (Run5)		0.876

Considering the achieved results, in the next section we focus on optimizing the performance of the SimGreedy algorithm, to face the practical requirements of real-world application problems.

5 Optimizing Semantic Text Similarity

Although SimGreedy performs better in comparison to the content-based approach, based on the time complexity discussed before, it has a much longer execution time. We observed that SimGreedy is approximately 40 times slower than Solr so that SimGreedy generally has the query processing time of about 110 to 130 minutes while it takes about three minutes for Solr. The method can be especially inefficient when the documents become longer. Therefore, we apply two optimization techniques for SimGreedy to achieve a better execution time without degrading its effectiveness.

5.1 Two-Phase Process

In the first approach, we turn the procedure into a two-phase process [6]. In order to do this, we choose an alternative method with considerably less execution time in comparison to SimGreedy such as using Solr. Then, we apply the faster algorithm to obtain a first ranking of the results and afterwards, the top n percent of the results is re-ranked by applying SimGreedy. Therefore, the SimGreedy algorithm computes only on a portion of the data which is already filtered by the first (faster) one.

Considering that the alternative algorithm has the execution time of t and is k time faster than SimGreedy, applying this approach takes $t + t \cdot k \cdot n/100$ where n is the percentage of the selected data. In fact, this approach is $k/(1 + k \cdot n/100)$ times faster than running the SimGreedy algorithm standalone. While achieving better execution time, the choice of the parameter n can reduce the effectiveness of the SimGreedy method. Finding the optimal n such that performance remains in the range of significantly indifferent to the non-optimized SimGreedy is a special problem of this method.

Table 8. Execution time in minutes of the standard, Two-Phase, and Approximate Nearest Neighbor (ANN) approaches of SimGreedy. Models are trained on the Wikipedia corpus with 200 dimensions. There is no statistically significant difference between the achieved results of the evaluation metric (P@20).

Repres	Algorithm	Indexing time	I/O	Query time	Overall	P@20
W2V	SimGreedy	-	0:16	1:50	2:06	0.795
	SimGreedy + Two-Phase	-		0:50	1:06	0.772
	SimGreedy + ANN	0:28		0:17	1:01	0.782
RI	SimGreedy	-	0:14	2:07	2:24	0.788
	SimGreedy + Two-Phase	-		1:00	1:14	0.770
	SimGreedy + ANN	0:21		0:19	0:54	0.782

To apply this technique on the MediaEval collection, we selected Solr as the first phase. SimGreedy as the second phase uses vector representations trained on Wikipedia by Word2Vec and Random Indexing methods, both with 200 dimensions. For all the integer values of n from 1 to 100, we found an extremely similar behaviour between the two methods summarized in Fig. 5. To find the best value for n as the cutting point, we identified the highest precision value that is not significantly different (using Fisher's two-sided paired randomization test with $p = 0.05$) from the best one (i.e. when n is 100%). This corresponds to $n = 49$. Giving the second phase (SimGreedy) is about 40 times slower than the first

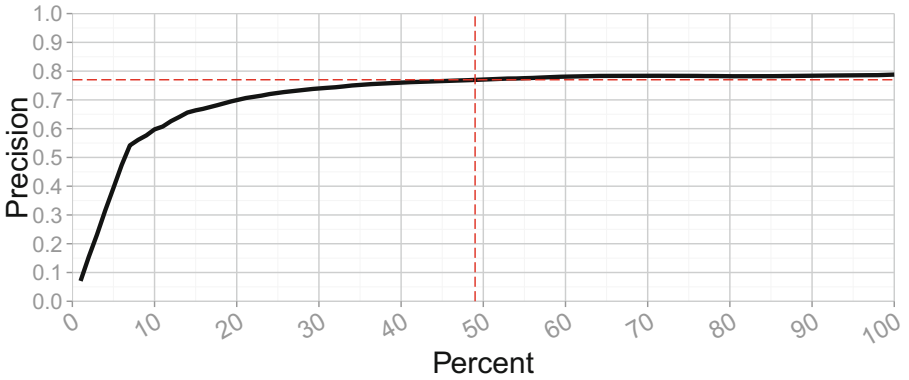

Fig. 5. Average performance of the two-phase approach with best value at around 49%

(Solr), using this approach improves the execution time to almost two times (48%) while the performance remains the same.

5.2 Approximate Nearest Neighborhood

In this technique, we exploit the advantages of Approximate Nearest Neighbor (ANN) methods [2]. Similar to Nearest Neighbor search, ANN methods attempt to find the closest neighbors in a vector space. In contrast to the Nearest Neighbor method, ANN approaches approximate the closest neighbors using pre-trained data structures, while in a significantly better searching time. Considering these methods, we can adapt the *maxSim* function of *SimGreedy* to an approximate nearest neighbor search where it attempts to return the closest node to a term. Therefore in this approach, first we create an optimized nearest neighbor data structure (indexing process) for each document and then use it to find the most similar terms.

The overhead time of creating the semantic indices depends on different factors such as the vector dimension, the number of terms in a document, and the selected data structure. While this excessive time can influence the overall execution time, it can be especially effective when the indices are used frequently by many queries.

We apply this technique on MediaEval by first creating an ANN data structure—denoted as semantic index—for each document using the scikit-learn library[5]. Due to the high dimension of the vectors (>30), we choose the Ball-Tree data structure with the leaf size of 30. The Ball-Tree data structure recursively divides the data into hyper-spheres. Such hyper-spheres are defined by a centroid C and a radius r so that points with a maximum leaf size are enclosed. With this data structure, a single distance calculation between a test point and the centroid is sufficient to determine a lower and upper bound on the distance

[5] http://scikit-learn.org/stable/.

to all points within the hyper-sphere. Afterwards, we use the semantic indices to calculate the SimGreedy algorithm. We run the experiment using vector representations with 200 dimensions using both Word2Vec and Random Indexing methods trained on Wikipedia.

Table 8 shows the results compared with the original SimGreedy as well as Two-Phase algorithm. The I/O time consists of reading the documents, fetching the corresponding vector representations of the words and writing the final results which is common between all the approaches. Although the ANN approach has the overhead of indexing time, its query time is significantly less than the original SimGreedy and also Two-Phase approach. We therefore see an improvement of approximately two times in the overall execution time in comparison to the original SimGreedy method. In spite of the time optimization, there is no significant difference between the evaluation results of the methods.

It should also be noted that since in MediaEval task, each topic has its own set of documents, the semantic index of each document is used only one time by its topic. Considering this fact, we expect a larger difference between the overall execution times when the indexed documents are used by all the topics as is the normal case in many information retrieval tasks.

6 Conclusions and Future Work

We explored the effect of textual semantic and optimization methods in the social media domain as an example of a semantic index. In addition, we checked the sanity and effectiveness of the methods on two information seeking tasks, namely sentence paraphrasing and sentence-to-paragraph similarity. We ran experiments on the MediaEval Retrieving Diverse Social Images Task 2013/2014 using Word2Vec and Random Indexing vector representations. Beside achieving state-of-the-art results, we show that SimGreedy—a semantic-based similarity method—outperforms a term-frequency-based baseline using Solr. We then focused on two optimization techniques: Two-Phase and Approximate Nearest Neighbor (ANN) approaches. Both the methods reduced the processing time of the SimGreedy method by half while keeping precision within the boundary of statistically insignificant difference.

Although these techniques similarly optimize the processing time, they show different characteristics in practice. While the Two-Phase approach needs preknowledge on the performance of the other search methods for setting the parameters, the ANN method can easily be applied on new domains with no need for parameter tuning. In addition, in the ANN approach, despite the overhead time of creating semantic-based data structures, the query time is significantly faster which is a great benefit in real-time use cases.

In future work, we will exploit the semantics of different facets (e.g. text, image, etc.) by first indexing and then combining them in the scoring process of our multimodal information retrieval platform. The concept index is achieved differently for text and image: For image facets, it represents the probability of a visual concept that has been learned from an image (e.g. from a visual classifier).

For text facets, it represents the probability of a term being conceptually similar to its context (e.g., document, window of the terms, and etc.). Despite the effectiveness of SimGreedy (as an approach for semantic similarity), for each term in the source document, it only finds the highest similar term in the destination and ignores the others with less similarity value. We therefore want to study new, alternative similarity measures that match terms with groups of related terms.

References

1. Agirrea, E., Baneab, C., Cardiec, C., Cerd, D., Diabe, M., Gonzalez-Agirrea, A., Guof, W., Mihalceab, R., Rigaua, G., Wiebeg, J.: Semeval-2014 task 10: multilingual semantic textual similarity. In: SemEval (2014)
2. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J. ACM (JACM) **45**(6), 891–923 (1998)
3. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 238–247 (2014)
4. Clinchant, S., Ah-Pine, J., Csurka, G.: Semantic combination of textual and visual information in multimedia retrieval. In: Proceedings of the 1st ACM International Conference on Multimedia Retrieval (2011)
5. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision (at ECCV) (2004)
6. Dang, V., Bendersky, M., Croft, W.: Two-stage learning to rank for information retrieval. In: Proceedings of European Conference on Information Retrieval (2013)
7. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. (JASIS) **41**, 391 (1990)
8. Depeursinge, A., Müller, H.: Fusion techniques for combining textual and visual information retrieval. In: Müller, H., Clough, P., Deselaers, T., Caputo, B. (eds.) ImageCLEF, vol. 32, pp. 95–114. Springer, Heidelberg (2010)
9. Eskevich, M., Jones, G.J., Aly, R., et al.: Multimedia information seeking through search and hyperlinking. In: Proceedings of the Annual ACM International Conference on Multimedia Retrieval (2013)
10. Ionescu, B., Popescu, A., Lupu, M., Gînsca, A.L., Boteanu, B., Müller, H.: Div150cred: a social image retrieval result diversification with user tagging credibility dataset. In: ACM Multimedia Systems Conference Series (2015)
11. Ionescu, B., Radu, A.-L., Menéndez, M., Müller, H., Popescu, A., Loni, B.: Div400: a social image retrieval result diversification dataset. In: Proceedings of ACM Multimedia Systems Conference Series (2014)
12. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
13. Jurgens, D., Pilehvar, M.T., Navigli, R.: Semeval-2014 task 3: cross-level semantic similarity. In: SemEval 2014, p. 17 (2014)

14. Liu, C., Wang, Y.-M.: On the connections between explicit semantic analysis and latent semantic analysis. In: Proceedings of Conference on Information and Knowledge Management, New York, USA (2012)
15. Liu, N., Dellandréa, E., Chen, L., Zhu, C., Zhang, Y., Bichot, C.-E., Bres, S., Tellez, B.: Multimodal recognition of visual concepts using histograms of textual concepts and selective weighted late fusion scheme. Computer Vision and Image Underst. **117**, 493–512 (2013)
16. Magalhaes, J., Rüger, S.: Information-theoretic semantic multimedia indexing. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, pp. 619–626. ACM (2007)
17. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: the next frontier for innovation, competition, and productivity. McKinsey Global Institute (2011)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint 2013 arXiv:1301.3781
19. Paramita, M.L., Grubinger, M.: Photographic image retrieval. In: Müller, H., Clough, P., Deselaers, T., Caputo, B. (eds.) ImageCLEF, vol. 32, pp. 141–162. Springer, Heidelberg (2010)
20. Pham, T.-T., Maillot, N., Lim, J.-H., Chevallet, J.-P.: Latent semantic fusion model for image retrieval and annotation. In: Proceedings of Conference on Information and Knowledge Management (2007)
21. Rekabsaz, N., Bierig, R., Ionescu, B., Hanbury, A., Lupu, M.: On the use of statistical semantics for metadata-based social image retrieval. In: Proceedings of the 13th International Workshop on Content-Based Multimedia Indexing (CBMI) (2015)
22. Sabetghadam, S., Lupu, M., Bierig, R., Rauber, A.: A combined approach of structured and non-structured IR in multimodal domain. In: Proceedings of ACM International Conference on Multimedia Retrieval (2014)
23. Sahlgren, M.: An introduction to random indexing. In: Methods and Applications of Semantic Indexing Workshop in the Proceedings of Terminology and Knowledge Engineering (2005)
24. Thomee, B., Popescu, A.: Overview of the ImageCLEF 2012 Flickr photo annotation and retrieval task. In: Proceedings of Cross-Language Evaluation Forum (CLEF) (2012)