VERGE: A Video Interactive Retrieval Engine

Stefanos Vrochidis, Anastasia Moumtzidou, Paul King, Anastasios Dimou, Vasileios Mezaris
and Ioannis Kompatsiaris
Informatics and Telematics Institute
6th Km Charilaou-Thermi Road, Thessaloniki, Greece
{stefanos, moumtzid, king, dimou, bmezaris, ikom}@iti.gr

Abstract

This paper presents the video retrieval engine VERGE, which combines indexing, analysis and retrieval techniques in various modalities (i.e. textual, visual and concept search). The functionalities of the search engine are demonstrated through the supported user interaction modes.

1 Introduction

Advances in multimedia technologies combined with the decreasing cost of storage devices and the high penetration of the world wide web have led to huge and rapidly growing archives of video content in recent years. This fact places the need for the development of advanced video search engines that extend beyond traditional text retrieval and exploit modern image and video analysis techniques, such as content-based indexing and high level concept extraction.

This paper demonstrates the interactive retrieval engine VERGE[1] built by ITI-CERTH[2], which supports multimodal retrieval functionalities including text, visual, and concept search. Earlier versions of VERGE have been used in a number of video evaluation workshops (TRECVID 2006, 2007 2008, 2009) [1], [2], as well as in related events, such as VideOlympics 2007, 2008 and 2009 [3]. In the rest of the paper, section 2 discusses the problem addressed by VERGE, while in section 3 a detailed description of the latter is presented. Section 4 demonstrates the search engine through use cases and finally section 5 concludes the paper.

2 Problem Addressed

Due to the already discussed growth of accessible video content, there is an increasing user demand for searching in video collections, in order to spot specific incidents or events. For instance, a user could be interested in finding a scene in a movie, in which two actors are arguing, or view a part of a documentary where a famous politician is speaking to a crowd. In the case that no search capabilities are available, the user would have to browse the entire video to find the desired scene, a very time-consuming and difficult task, especially when the video is long and the target scene is short in terms of time.

To support the user in such a use case, video should be pre-processed in order to be indexed in smaller segments and semantic information should be extracted. The proposed video search engine is built upon a framework that employs modern image and video analysis technologies to support the user in such search tasks. The performed analysis targets general case video data (i.e. documentaries, sports, educational videos, etc.), so it can be applied in almost any domain.

3 System Description

VERGE is an interactive video retrieval system, which realizes the framework of Figure 1. It combines basic retrieval functionalities in various modalities (i.e., visual, textual), accessible through a friendly Graphical User Interface (GUI) (Figure 2). The system supports the submission of hybrid queries that combine the available retrieval functionalities, as well as the accumulation of relevant retrieval results. The following basic indexing and retrieval modules are integrated in the developed search application:

- Visual Similarity Indexing Module;

- Text Processing and Recommendation Module;

- High Level Concept Extraction Module;

- Visual and Textual Concepts Fusion Module;

Besides the basic retrieval modules, the system integrates a set of complementary functionalities, including

---

[1] http://mklab.iti.gr/verge
[2] Informatics and Telematics Institute - Centre for Research & Technology Hellas
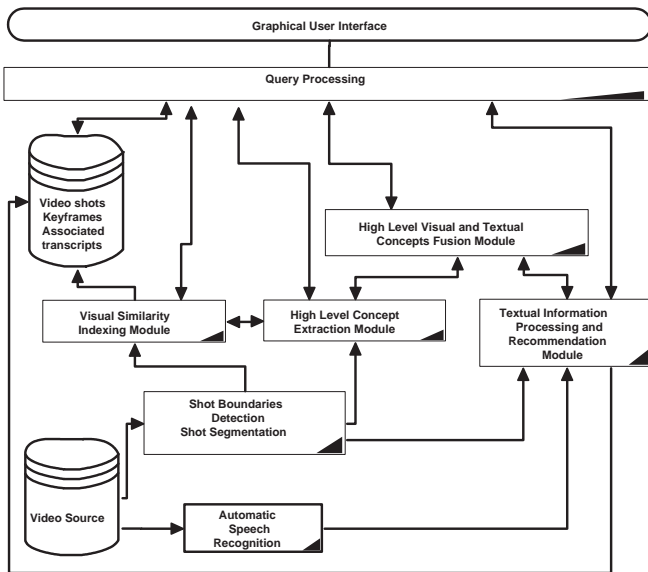
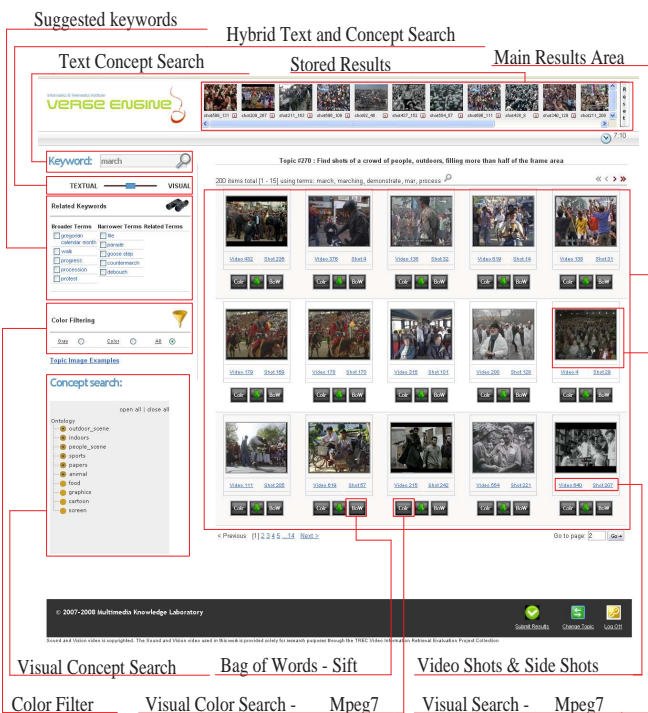**Figure 1. Architecture of the video search engine.**

Figure 2. User interface of the VERGE search engine.

temporal queries, color filtering options, fast video shot preview, as well as a basket storage structure. More specifically, the system supports basic temporal queries such as the presentation of temporally adjacent shots of a specific video shot and the shot-segmented view of each video. In addition, the system supports a fast shot preview by rolling three different keyframes. In that way, the user obtains adequate information of the video shot in a very short time. Furthermore, the system offers a color filter option that filters the presented results to either grayscale or color images. Finally, the selected shots, considered to be relevant to the query, can be stored by the user by employing a storage structure that mimics the functionality of the shopping cart found in electronic commerce sites. In the next sections we will describe the VERGE interface, the aforementioned basic retrieval modules and the system's specifications.

3.1   Interface Description

The Graphical User Interface (GUI) of VERGE is comprised of basically two parts, i.e. the surrounding layout and the results container. The surrounding layout of the interface contains the necessary options for formulating a query, presented in the left column, and the storage structure (basket) on the top of the page, where the user can store the relevant images. The text search resides in the left column of the GUI, where the user can insert a keyword or a phrase to search. A sliding bar residing just below this form is used for combining the results of the textual and visual concepts by assigning weights to each technique. Below this slider, the related terms of a text query are presented, which are produced dynamically according to the inserted textual information and aim at helping the user by suggesting broader, narrower and related terms. Underneath, there is a color filter option that limits the retrieved results to either colored or grayscale shots. Finally, on the bottom, the high level visual concepts are listed as a hierarchy for easier navigation.

The results container, which is located on the main part of the interface, comprises the part of the GUI, in which the results of a query are presented. Specific information and links are provided underneath each presented shot offering the following options to the user in the form of clickable buttons: a) to view all the shots of the specific video, b) to view the 12 temporally adjacent shots, c) to search for visually similar shots based on MPEG-7 Color descriptors, d) to search for visually similar shots exploiting SIFT descriptors, and e) to submit a shot to the basket. In addition, the user can watch a short preview of any shot by hovering the mouse over it, and can fire a visual search based on MPEG-7 Color and Texture descriptors by clicking on the specific shot.

3.2 Visual Similarity Indexing Module

The visual search module exploits the visual content of images in the process of retrieving visually similar results. Given that the input considered is video, these images are obtained by representing each shot with its temporally middle frame, called the representative keyframe. Visual similarity search is realized by extracting either global information, such as the MPEG-7 visual descriptors that capture different aspects of human perception (i.e. color and texture) from representative keyframes of video shots, or local information such as interest points described using [4]. Such information was extracted from the representative keyframe of each shot.

In the first case, we formulate a feature vector by concatenating different MPEG-7 descriptors to compactly represent each image in a multidimensional space. Based on an empirical evaluation of the system's performance, two different schemas where selected. The first one relied on color and texture, while the second scheme relied solely on color.

In the second case, the method employed was an implementation of the bag-of-visual-words approach as described in [5]. A large amount of local descriptors extracted from different frames was clustered to a fixed number of clusters using a k-means algorithm, and the resulting cluster centers were selected as visual words. Each image was subsequently described by a single feature vector that corresponded to a histogram generated by assigning each keypoint of the image to each closest visual word. The number of selected clusters (100 in our case) determined the number of bins and, consequently the dimensionality of the resulting feature vectors.

To enable faster retrieval, a multi-dimensional r-tree [6] indexing structure was constructed off-line using the feature vectors of all the shots. During the query phase the feature vector of the query image is fed to the index structure, which outputs a set of keyframes that are found to resemble the query one. Since the order of these identifiers is not ranked according to their level of similarity with the query example, an additional step for ranking these keyframes, using custom distance metrics between their feature vectors, was further applied to yield the final retrieval outcome.

3.3 Text Processing and Recommendation Module

The text search module exploits any textual information, which is associated with the video. This implementation utilizes text, which was extracted by Automatic Speech Recognition (ASR) from the embedded audio. The resulting annotations were used to create a full-text index utilizing Lemur [7], a toolkit designed to facilitate research in language modelling. The standard pipeline of text processing techniques was used in creating the index, including stopwords removal and application of the Porter [8] algorithm.

To assist the user in query iteration tasks, a hierarchical navigation menu of suggested keywords was generated at runtime from each query submission. Conventional term relationships were supplied by a local WordNet database [11], whereby hypernyms were mapped to broader terms and hyponyms to narrower terms. Related terms were provided by synset terms that were not used for automatic query expansion [2].

All standard post-coordination techniques were implemented for query formulation and editing tasks, including logical Boolean operators (e.g., OR, AND, and NOT), term grouping (using parentheses), and phrases (delimited by quotes), thus allowing the user to manipulate expanded query terms to improve precision.

Recall was boosted by automatic query expansion, also based on WordNet, whereby a list of expanded terms were generated from WordNet synsets. The system was constrained to using the first term in the query string due to the observation that short queries often produce more effective expansions [9]. Terms were chosen for expansion by measuring the semantic similarity between each synset term and the original (initial) query term by utilizing the extended gloss overlap method [10].

Although performance of the module is satisfactory in terms of time-efficiency and navigation, the quality of the results greatly depends on the reliability of the speech transcripts.

3.4 High Level Concept Extraction Module

This search modality offers selection of high level visual concepts (e.g. water, aircraft, landscape, crowd, etc.). The extraction of high level concept information was based on the approach described in [2]. A set of MPEG-7-based features were concatenated to form a single MPEG-7 feature vector for every shot, while a Bag-of-Words feature based on SIFT descriptors of local interest points was also calculated for every shot, as described in section 3.2.

A set of SVM classifiers (LIBSVM) [12] was trained for each high level visual concept and each possible shot representation (MPEG-7, SIFT) separately. Subsequently, a second set of SVM classifiers, one per high level concept, was trained for fusing the previous classification results and produce a final score in the range [0,1] associating each shot with each considered high level concept. In the present version of the VERGE engine, 42 of these concepts (e.g. building, car, mountain, snow, etc.) are considered. These concepts are depicted in the GUI in the form of a hierarchy (Figure 2).

3.5 Visual and Textual Concepts Fusion Module

This module combines high level visual concepts, with textual information by applying a manually assisted linear fusion. From a usability point of view, the user provides a keyword and specifies, with the aid of a slider, the relevant significance of the textual versus the visual results.

The procedure is reflected by Equation 1, where i is a specific shot, Sim_i is the final similarity score after the fusion, $VScore_i$ is the normalized degree of confidence for a given visual concept, $TScore_i$ is the normalized similarity score of the textual module and, finally, α and β are the weights assigned to their original values, respectively.

$$Sim_i = \alpha \cdot VScore_i + \beta \cdot TScore_i, \text{ where } \alpha + \beta = 1 \quad (1)$$

It should be noted that the similarity scores are normalized to a range between 0 and 1, where the higher the value, the more relevant it is to the query. Obviously, if one of the weights is set to zero or one, the results obtained are either exclusively textual or visual.

3.6 Implementation Issues

The search system, combining the aforementioned modules, was built on open source web technologies, more specifically Apache server, PHP, JavaScript, mySQL database, Strawberry Perl. It also requires the Indri Search Engine that is part of the Lemur Toolkit [7] and a local installation of WordNet.
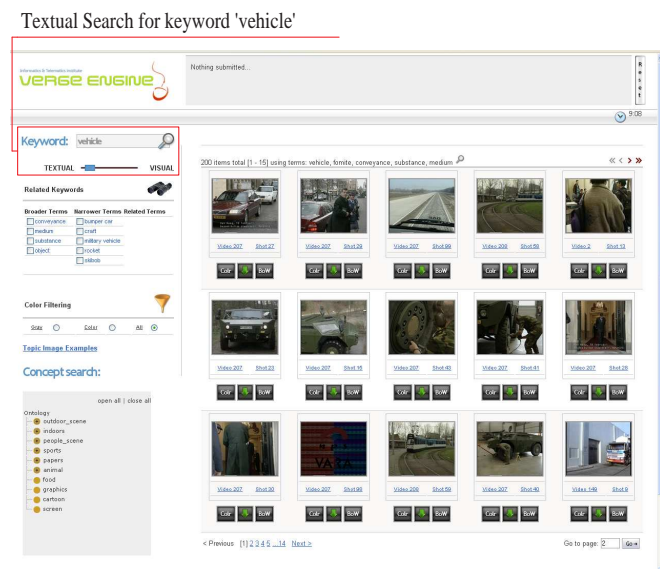
Textual Search for keyword 'vehicle'

Figure 3. Textual search for keyword 'vehicle'.

4 Interaction Modes

In this section, we present two use cases of the VERGE system to demonstrate its functionalities.

In the first usage scenario we suppose that a user is interested in finding 'street scenes with moving vehicles'. At first, the user inserts the keyword 'vehicle' into the text query box and moves the slider to the left, which leads to placing emphasis on the textual information generated from ASR (Figure 3). Although the results retrieved are quite satisfactory, it is expected that by increasing the weight assigned to the high level visual concept module, the results will be enhanced. Therefore, the user moves the textual-versus-visual information slider to the right, thus assigning more weight on the results obtained by processing the visual information. Figure 4 depicts the output obtained, demonstrating that the high level visual concept module improves the results. In addition, the user can view the sideshots (i.e. the temporally adjacent shots) and the associated textual information of a specific video shot. Figure 5 illustrates the interface when sideshots are visualized. From the results it can be observed that some of the sideshots depict vehicles, which is very reasonable as temporally adjacent shots are usually semantically related. Finally, the user selects a shot from the retrieved results and searches for visually similar ones, based on the shot's MPEG-7 color and texture features. By executing this query, more relevant results are retrieved, as it is illustrated in Figure 6.
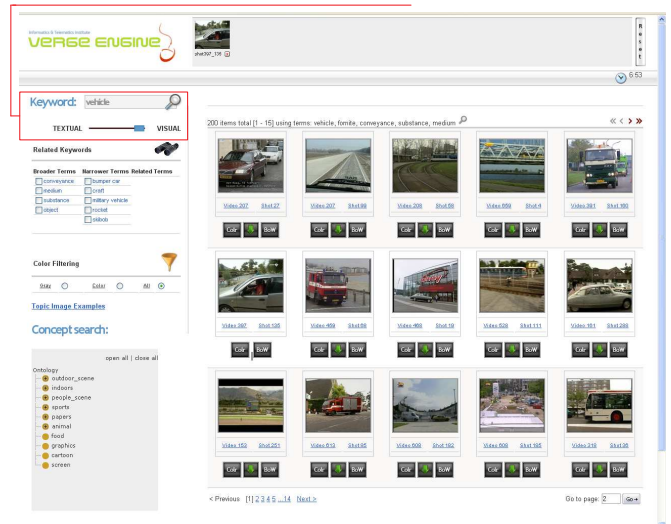
High Level Visual Concept 'vehicle' Results

Figure 4. High level visual search for concept 'vehicle'.

In the second use case, the user is searching for video shots depicting 'scenes with crowd'. At first, the user in-

serts the keyword 'crowd' into the text query box and moves the slider to the middle. The shots retrieved (Figure 7) result from the linear fusion of the results of the textual and the high level visual concept modules. To retrieve more related shots, the user selects one of the retrieved keyframes and searches for visually similar shots using MPEG-7 color features. The retrieved results are illustrated in Figure 8. Finally, the user can search using local features (SIFT) by clicking on the 'BoW' button (Figure 9) of a specific video shot.
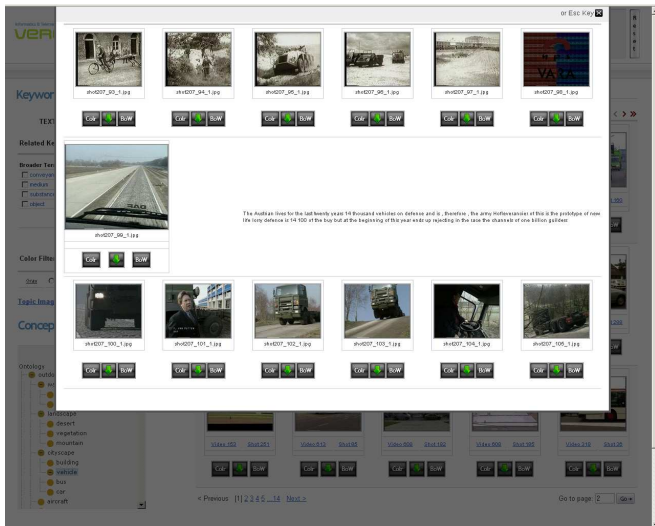
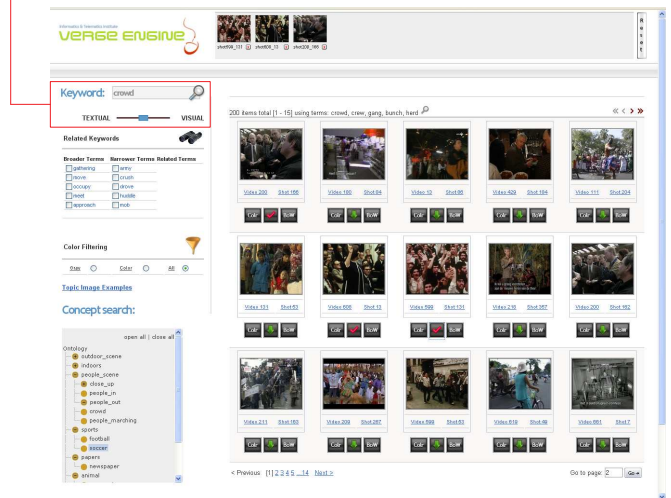**Figure 5. Temporally adjacent shots.**

Figure 6. Visual search using MPEG-7.

Figure 7. Linear fusion of textual and high level visual concept module results for the query 'crowd'.
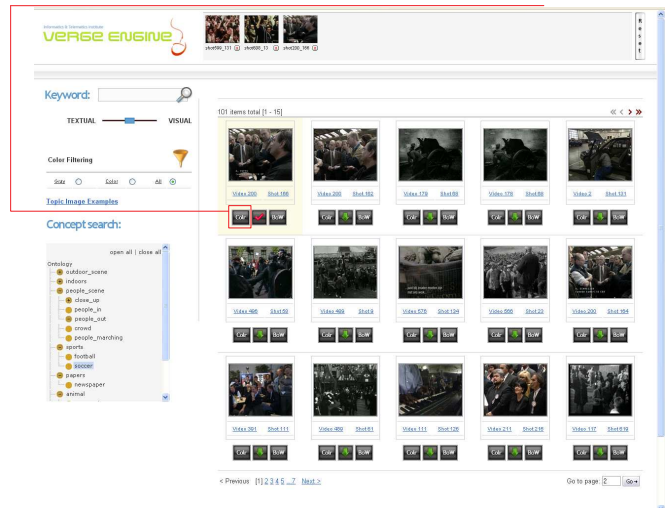
Figure 8. Visual search using color-based MPEG-7 features.
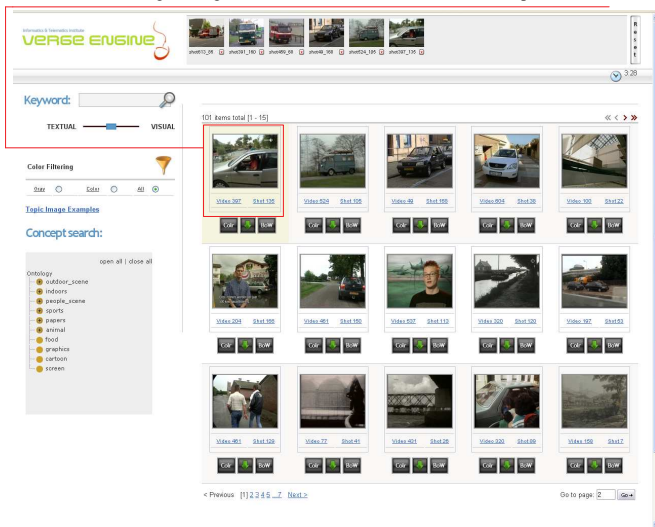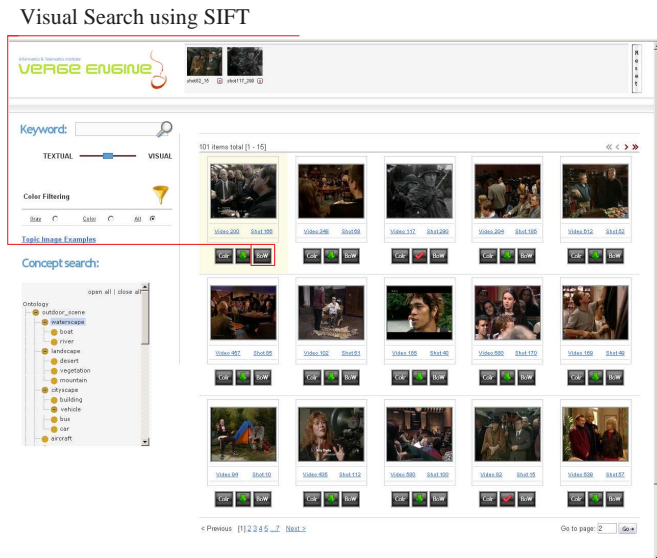
Visual Search using SIFT

Figure 9. Visual search using SIFT.

5 Summary and Future work

This paper presented a description of the interactive video search engine VERGE. Its performance was demonstrated through different user interaction modes in order to reveal its potential. Currently VERGE is employed in research experiments dealing with the exploitation of user implicit feedback, expressed by user navigation patterns (i.e. mouse clicks and keystrokes), as well as by sensor outputs (e.g. eye tracker) during interactive retrieval tasks, in order to optimize and complement existing content based retrieval functionalities and provide recommendations. Future activities include the adaptation of VERGE modules to support domain specific search with a view to retrieving web pages offering environmental services.

The website of VERGE search engine is available at: http://mklab.iti.gr/verge/, including up to date information about the latest implementations, video tutorials and demonstrations, as well as links to the different online versions of VERGE.

6 Acknowledgements

References

[1] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid, In Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, pp 321–330, New York, NY, USA, 2006.

[2] A. Moumtzidou, A. Dimou, P. King, S. Vrochidis, A. Angeletou, V. Mezaris, S. Nikolopoulos, I. Kompatsiaris, L. Makris. ITI-CERTH participation to TRECVID 2009 HLFE and Search, 7th TRECVID Workshop, Gaithersburg, USA, November 2009, 2009.

[3] C.G.M. Snoek, M. Worring, O. de Rooij, K.E.A. van de Sande, V. Rong, A.G. Hauptmann. VideOlympics: Real-Time Evaluation of Multimedia Retrieval Systems, Multimedia, IEEE, 15(1):86–91, 2008.

[4] D. G. Lowe. Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60(2):91–110, 2004.

[5] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos, in Proceedings of the International Conference on Computer Vision, pp 1470–1477, 2003.

[6] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *SIGMOD '84: Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57, New York, NY, USA, 1984. ACM.

[7] The lemur toolkit. http://www.cs.cmu.edu/ lemur.

[8] M. F. Porter. An algorithm for suffix stripping, *Program*, 14(3):130–137, 1980.

[9] H. Fang and C. X. Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122, New York, NY, USA, 2006. ACM.

[10] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, 2003.

[11] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, The MIT Press, May 1998.

[12] C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001.