The Lowlands team at TRECVID 2007

Robin Aly[1], Claudia Hauff[1], Willemijn Heeren[1],
Djoerd Hiemstra[1], Franciska de Jong[1], Roeland Ordelman[1],
Thijs Verschoor[1], and Arjen de Vries[2]
[1]University of Twente, The Netherlands [2] CWI, The Netherlands

October 23, 2007

Abstract

Type	Run	Description	MAP
	Official		
A	UTen	English ASR	0.0031
A	UTt_hs-t2-nm	Top-2 concepts from t_hs graph method with neighbor multiply	0.0137
A	UTwiki-t2-nm	Top-2 Wikipedia concepts with neighbor multiply	0.0131
A	UTwiki-t2-en-nm	Top-2 Wikipedia concepts and English ASR with neighbor multiply	0.0107
A	UTwiki-t2-nl-nm	Top-2 Wikipedia concepts and Dutch ASR with neighbor multiply	0.0096
A	UTwordnet-t2-mult	Top-2 Wordnet concepts with neighbor multiply	0.0083
	Additional		
A	UTnl	Dutch ASR	0.0031
A	UTwikiS-t2-nT	Top-2 Wikipedia concepts on stemmed queries with neighbor using the concept detector scores from the B_tsinghua-icrc_5 run	0.0410
A	UTt_hs-t2-n	Top-2 concepts from t_hs graph method of stemmed queries with neighbor the concept detector scores from the B_tsinghua-icrc_5 run	0.0346
I	UTinter-wiki-nm	Interactive Search Task using Wikipedia concepts with neighbor multiply	0.0405
I	UTinter-en	Interactive Search Task using ASR based search	0.0338

Summary: Concept to Query does not differ very much; Best combination method neighbor; Preprocessing of Queries helps; Choice of detector source helps. For all components further investigations needed. Interactive system: rather poor but good insights why.

1 Introduction

Bridging the semantic gap is a key problem for multimedia information retrieval tasks such as video search. [9] It requires coupling of the well understood extraction methods for low level features from media files (e.g. color histograms or audio energy) and the semantically rich descriptions or concepts[1] in which users express their information needs (e.g. *Find me pictures of a sunrise*). In this paper we investigate how the concept combination methods we developed [1] [3] perform against an ASR-only method[2], and whether combining the two helps.

Concept detectors are commonly trained through positive and negative examples on a certain training dataset. For a particular domain appropriate sets of concepts and training data have to be selected. A less straightforward issue is how to handle queries that do not correspond to exactly one concept from the selected set of concepts. Due to the lack of knowledge about the structure of the *semantic space*, it is not an option to simply increase the number of detectors up to the point where all requested concepts are covered. The hypothesis is that in order to support searching for *Condoleezza Rice* with a search system that only has the concepts *Face* and *Women* available, the uncovered concept has to be expressed as a combination of concepts for which detectors exist.

In this paper we describe three novel techniques to combine concept scores. The main innovations are in the score modification via the scores of preceding and following shots, and in combining the output for one detector with the output of other detectors. We also ran our IR system PF/Tijah [5] on the ASR output and investigated ways to integrate the results with the results from concept combination. At last we performed unofficial user studies on a baseline interactive version of our system to measure the effectiveness of user interaction.

This paper is structured as follows. In Section 2 we introduce the system we used for our experiments. In Section 3 we elaborate on our concept combination methods. Section 4 briefly outlines the PF/Tijah system. Section 5 shows the setup for the interactive search task. Section 6 describes the experiments we undertook to verify our methods. Section 7 concludes the paper.

2 Minos System Overview

We named the IR System which we used to carry out the runs *Minos*[3]. It is designed to allow several search strategies as well as to combine them. The system architecture is shown in Figure 1.

In the data access layer we use the XML database, MonetDB-XQuery, with PF/Tijah as a Text IR extension. The data is stored in MPEG7 documents which contain time interval, English and Dutch ASR output and the scores of the concept detectors from the University of Amsterdam. MonetDB-XQuery provides a method to execute queries using an XML remote procedure call (XRPC).

[1] In TRECVID terminology high level features

[2] ASR: automatic speech recognition

[3] Minos is a mythologic King of Crete who created a unescapable labyrinth

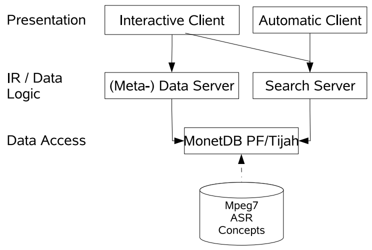


Figure 1: Architecture

In the IR logic layer, the search server is concerned with encapsulating the information retrieval logic and to hide the system's complexity from the presentation layer. It has the ability to use different search modules. The two search modules implemented at the moment provide concept based and text (ASR) based search. The data server provides a unified interface to deliver (meta-) data to the user. Bulk binary data, such as key frames are provided through a URL. The protocol from the IR logic layer to the presentation layer is using Web Services defined by the web service description language (WSDL) to ensure interoperability.

In the presentation layer we implemented two clients. One client is designed to carry out automatic search tasks. It gets passed the TRECVID topic file and automatically executes one topic after each other for all system configurations using the web service. The interactive client allows a human user to interact with the search system. At start up the client program gets configured which search module it should use. This setting, together with the text query, gets passed to the search server. The server returns a list of shot identifiers together with a rank and a score. For all the shot identifiers the needed metadata is retrieved from data server. The key frame pictures get loaded from a potentially independent web server.

3 Concept Combination

As was mentioned earlier concept combination is carried out because one concept is unlikely to be enough to answer a user's query. Our notion of combination[1] focuses mainly on the co-occurrence of concepts. Unlike techniques mentioned in [11] we do not take relationships between concepts into account. Therefore the two concepts *Animal* and *Dog* would be treated the same for a query "Find me dogs" allowing the *Animal* concept to introduce noise (e.g. *Cats*) into the result. A big advantage is that there is no need for an ontology to represent those relationships.

3.1 Query To Concepts

Users cannot be expected to know the concepts that are available to the system. User queries usually either consist of a few keywords (e.g. *Beach*) or more elaborate natural language requests (e.g. *Find me pictures of a beach with people.*). In the best case, the query contains one or more concept names and syntactic matching is sufficient. However, often this will not be the case. For instance, the set of concepts included in TRECVID include *Outdoor*, *Waterscape* and *People* but not *Beach*. Hence, the first task is the extraction of TRECVID concepts underlying the queries. The natural language query and the concepts available for the collection are matched and a ranking of relevant concepts is derived that shall resemble the information need expressed in the query as close as possible. We implemented two query to concept approaches: one is based on WordNet [2] glosses and Wikipedia pages, the second is based on WordNet's graph structure.

In the gloss (Wikipedia) approach, we consider WordNet glosses (Wikipedia pages) describing a concept as substitutes of the concepts. The relevant concepts to a query can then be found by using Text IR methods on the collection of the documents describing the concepts.

In the second approach, WordNet's graph structure is exploited. TRECVID concepts are mapped to synsets in WordNet. The distances between query terms and concepts on the graph are used to rank the concepts.

3.2 Concept Preprocessing

Given the ranked list of concepts that are returned for a text query the system still has to select some concepts from this list for their combination. Using the whole list is not advisable as the query to concept step might return all concepts available to the system, although the irrelevant ones only with very small score. In [3] we performed studies on various strategies. Taking the top-2 concepts from the list showed the best performance. We used this setting in all experiments throughout this paper.

We used the concept detector scores from the A_uva.Coeus_4 run of the high level feature detection task. We chose this run because we used the detector results from the University of Amsterdam[10]. Because we used these detectors in earlier experiments[1, 3], we expect better comparability. As our methods need scores within the interval $[0..1[$ we linearly scaled the scores to the desired interval. We had to take this decision as probabilistic scores were not available.

3.3 Combination of Concept Scores

In the following we describe the combination methods we used to calculate a joint score from the output of multiple detectors.

Figure 2 shows the definition of all used combination functions. The function $r)$ (1) returns the previous described derived score of the shot s_j as calculated from the rank. The function $smooth$ (2) assumes that it is more likely that a concept c appears in the shot s_j if it also appears in previous or following shots. Similar approaches have been investigated using the text from automatic speech recognition associated with shots [4]. We define a surrounding neighborhood as a fixed number nh of shots before and after the actual shot s_j that contribute to the score of s_j.

The function $mult$ (3) multiplies adds the logarithm of the scores of all concept detectors. At the end it applies the $exp()$ function to bring the resulting score back into the interval $[0..1[$.

Functions on single concept:

$$r(c, s_j) = \frac{rank(s_j) - minRank(c)}{maxRank(c) - minRank(c)} \tag{1}$$

$$smooth(c, s_j) = \frac{\sum_{i=j-nh}^{j+nh} r(c, s_i)}{2nh + 1} \tag{2}$$

Functions on multiple concepts:

$$mult(C, s_j) = exp(\sum_{c \in C} log(r_c(s_j))) \tag{3}$$

$$n(C, s_j) = \frac{\sum\limits_{c \in C} r_c(s_j) \frac{\sum\limits_{c' \in C \setminus c} smooth(c, s_j)}{|C|-1}}{|C|} \tag{4}$$

$$nm(C, s_j) = \frac{\sum\limits_{c \in C} r_c(s_j) exp(\sum\limits_{c' \in C \setminus c} log(smooth(c, s_j)))}{|C|} \tag{5}$$

Figure 2: Combination Functions

The Neighbor function n (4) considers all base scores multiplied with the average of the smoothed scores of the other concepts to apply. nm (5) is an extension of the $mult$ function which weighs the individual scores by the $log()$ of averaged smoothed scores of other concepts.

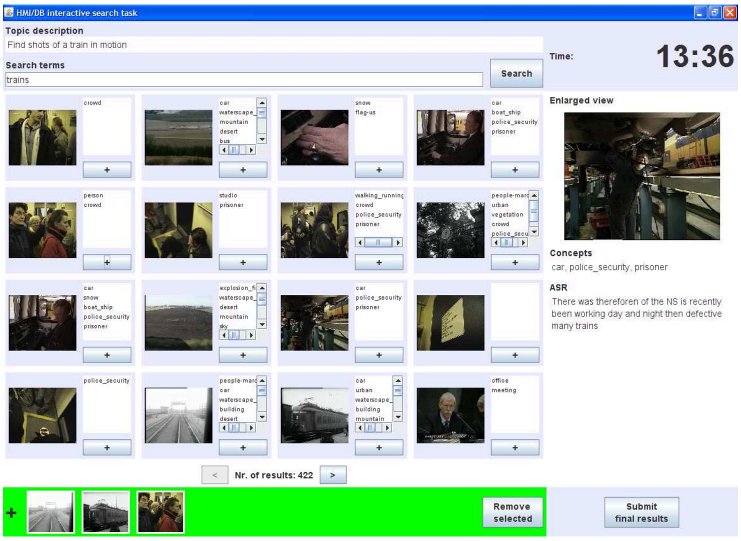
4 PF/Tijah TextIR

We kept all information in an MPEG7 conform documents. To store the scores of the feature donations we extended the mpeg7:VideoSegmentType to include Concepts subelement which in turn contains all concept scores of each subject.

Because the unit of retrieval was a shot, we used all ASR and automatic speech translation [6] from speaker segments overlapping with the shot segment to retrieve a shot. In this way the text associated with the shot could be a little more than what was actually spoken during the shot. Neighboring shots are considered to have a similar relevance; therefore this is not problematic.

In order to keep the data format to MPEG7 we extended the available vocabulary to also contain concept scores. This was done through creating a new schema on top of the existing MPEG7 schemas extending the existing type $VideoSegmentType$ to allow definition of concepts. standard.

We used the protocol XML Remote Procedure We implemented three such XRPC functions: (i) one which gets passed the query text and the language returning a ranking of shots, (ii) one which gets passed the query text and returns a list of concepts and (iii)



HMI/DB interactive search task

Topic description: Find shots of a train in motion

Search terms: trains

Time: 13:36

Concepts: car, police_security, prisoner

ASR: There was thereforen of the NS is recently been working day and night then defective many trains

Nr. of results: 422

Figure 3: Screen shot of the search interface.

a function which retrieved all metadata for a list of shot identifiers.

To see if a joint result of ASR output and concept combination could be beneficial we use the score from the shots found from ASR as "artificial" concept that could get combined like the others.

5 Interactive Search

We developed a baseline video search interface and addressed its effectiveness and acceptance in unofficial interactive runs. The system will be developed further to study search in collections where the spoken content can be exploited as time-stamped metadata generated through e.g., ASR. This holds for audio and video collections whose visual content mainly consists of talking heads; e.g., lecture recordings, meeting recordings, and interview collections. For speech-driven metadata the TRECVID tasks may be considered difficult as they target visual features in the video documents. However, this platform allows us to compare our baseline system's performance to that of other systems.

5.1 User Interface

Since most users, i.e. non-expert users, normally formulate text queries when using search engines, we only included query-by-keyword search (as opposed to query-by-example or query-by-concept search) in our baseline search system. However, we tested two manners of query processing for retrieval: (i) ASR-based search (UTinter_en) and (ii) concept-based search (UTinter_wiki_nm). These differences currently do not affect the type or manner of information presentation in the user interface.

A screen shot of the user interface is given in Figure 3. After processing a query, the total number of results found is reported. Results are shown per 16 keyframes in a 4 x 4 matrix. For each keyframe the concepts most strongly associated with it are given as well as the option to move that particular shot to the list of results that users definitely want to keep. This is done by clicking the plus-button next to a shot. The definitive selection is shown in the green bar at the bottom of the screen. Clicking on a keyframe gives more precise information on that frame on the right hand side of the screen: an enlarged view of the shot, the list of concepts associated with it, and the machine-translated English version of the Dutch ASR text associated with the shot.

As opposed to more advanced video search systems, we have not (yet) included ways to present relations between results, such as time relations or stories, or concept relations. Six Dutch participants (age range=21-27; 1 female, 5 males) each completed eight topics, four on each system variant. They all used search engines on a daily basis and three out of six indicated to also search for videos. They furthermore regularly searched online library catalogs. They were novice users of the system.

Topics, queries and results were in English, the second language of our users. Tests were run on PCs with 19" monitors in a quiet room. Before the actual test, users filled out a demographic questionnaire, which was followed by an explanation of and practice with the search system. This lasted about 20 minutes. During testing, system and topic order was counterbalanced across participants. They received monetary compensation for their efforts.

Between performing the search tasks on the two different systems, participants got a short break, and after each topic they filled in a post-topic questionnaire (translated to Dutch from the CMU2006 example[4]). Participants used the full 15 minutes per topic. During testing we measured the interaction with the system by logging user actions. After the interactive task a post-test questionnaire was administered on the system's general usability.

For score computation, result sets were filled to 1000 results. If the user's result set was not large enough it was completed with the results from his/her last query, and if necessary the set was further completed with the results from the automatic run for that topic. Double entries were of course removed.

6 Experiments

In this section we describe the experiments we did to verify our methods. First in Section 6.1 Runs according to the automatic search task description of TRECVID are described. The following Section 6.2 describes the outcome of our interactive user studies with the search system.

6.1 Automatic Runs

All our official runs are automatic runs. For the six runs we used the text IR based method with the Wikipedia and WordNet corpus and the graph based query to concept

[4] Last visited on Oct. 22 2007: http://www-nlpir.nist.gov/projects/tvpubs/tv6.papers/cmu_talk_search.slides.pdf

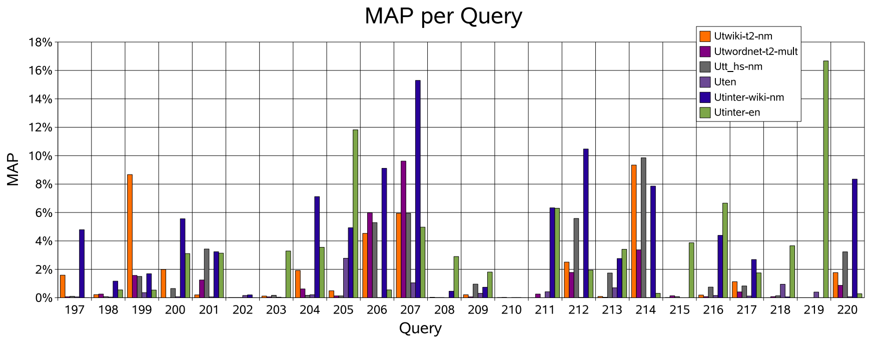


Figure 4: Per Query Average Precision

method hierarchical shortest path. We left out other graph based methods as they did not help increasing the performance in [3]. The given topics where then fully automated executed by the system.

Overall one of our runs reached the median of all submitted runs. Later we found out that there were some simple changes of our methods which improved the results significantly.

To compare the different Query to Concept mechanisms we compare the two official runs UTwiki-t2-nm and UTt_hs-nm together with the unofficial run UTwordnet-t2-nm (MAP 0.0139) it is not possible to conclude whether graph or text based methods are to be preferred.

A comparison between the combination methods based on the official run is problematic. There is an indication that the neighbor multiply method is better to the multiply method. To what extend this is true would have to be verified by runs using the same Query to Text method but varying combination methods.

We also compared the performance of our system when using Dutch and English language. For Dutch we used the direct ASR output and human translated topics. The result of this unofficial run UTnl was 0.0031 and therefore exactly the same as the one from English, which was machine translated.

Furthermore, we investigated whether using text scores, as another concept, helps. From the listed runs we have to conclude that using ASR - at least in this manner - is decreasing performance.

Additional checks on the returned concepts from the Query to Concept phase revealed that very often the same concepts were chosen. Investigations showed that this was due the nearly constant beginning of the textual topic “Find shots of”. Introducing a stop word mechanism which removed this bit yielded significant improvements. Hereafter all reported results were achieved using this stop wording.

To see whether the chosen source of concept detector scores matters we ran the combination UTwiki-t2-nm on all available sources, see Figure 5. It can be seen that the achieved MAP is significantly different depending on the source. The source we chose for the official runs (A_uva.Coeus_4) performed within the upper third of the sources. The run B_tsinghua-icrc_5 yielded the best results. We used this detector

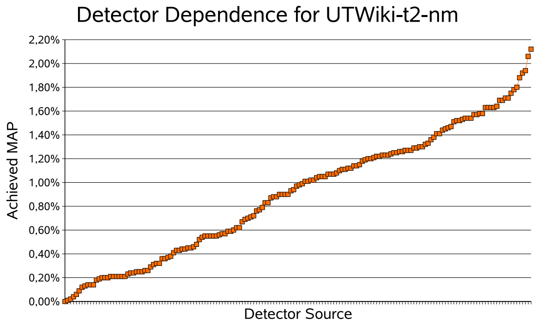


Figure 5: Dependence on Base Detectors

source for another intensive investigation of the performance of all query to concept and combination methods. As reference we report the run which resulted in 0.0410 MAP, which was using the wiki explanation of the concept, and the graph based method t_hs.

6.2 Interactive Runs

The UTinter_en interactive run got 3.5% and the UTinter_wiki_nm run got 4.63%. This difference, however, was not significant, since UTinter_en scored higher on some topics whereas UTinter_wiki_nm got better results on others. In comparison with the official interactive runs of other groups, our baseline system ranks among the lowest scoring interactive systems. This may be considered unsurprising given the basic nature of our user interface and the fact that we had novice users. In comparison with the corresponding automatic runs (0.31% and 1.37%, respectively) an improvement was found with users in the loop.

The interactive results per topic can also be found in Figure 4. For topics 0197, 0207, 0212, 0214 and 0220 concept-based search scored much higher than ASR-based search. For topics 0205, 0215, 0218 and 0219 it was the other way around. Most noticable are the results for topic 0219 (*Find shots that contain the cook character in the Klokhuis series*), where ASR-based interactive search outperforms all other conditions. Given that the content as well as our searcher are Dutch, he could use his knowledge of the TV show during search in the ASR text.

6.2.1 User Performance and Usability

In the UTinter_wiki_nm run participants on average formulated almost 17 queries, looked at 25 previews and saved almost 12 shots per topic. Average query length was 2.8 words. In the UTinter_en run participants on average formulated almost 27 queries, looked at 25 previews and saved almost 12 shots per topic. Average query length was 1.7 words. Even though the interface did not differ between the two system variants, users might have adapted to the situation at hand (with longer, but less queries for the concept-based run). This is an interesting observation, since searchers were only told that result generation differed between the variants, whereas the actual difference was not explained. We need to explore the user logs further to study this trend.

As for the post-topic and post-test questionnaires, we found that users rated the ASR-based search higher than the concept-based search with medians of 4 and 3, respectively (on a scale of 1=poor to 5=good). The individual questions concerning (i) ease to find results, (ii) sufficient time to complete search, and (iii) overall satisfaction with results showed the same trend between system variants.

With respect to the individual topics, users found topics 0197, 0202, 0203, 0208, 0210, and 0211 especially difficult, rating the ease to find relevant shots at 1 or 2. On the other hand, topics 0199, 0204, 0212, and 0213 were answered relatively successful.

The post-test questionnaire addressed the user interfaces usability by asking about learnability, satisfaction, ease of use, and interface design on a scale of 1 (=poor) to 5 (=good). The median for ease of use was high, i.e. 5, but overall satisfaction was just below average at 2.5. The system was judged relatively easy to learn (3.5) and also its design was rated positively (4). According to the participants improvement was needed in the match between the shot and its associated concepts, but none of them mentioned the relatively poor quality of the ASR text. Possibly, they did not use the ASR text shown with the previews as it has been found that low-quality ASR does not help users, e.g., [8][7].

7 Conclusion

We conclude that we achieved in the official runs around the median of the other systems. Later we found that stemming and query stop words improved the results significantly. The usage of English or Dutch ASR (or machine translated ASR) did not yield a significant difference. In comparison to combination methods the performance was worse. To incorporate them as an artifical detectors score into the combination lowered MAP. Finally we found that our method strongly depends on the kind of detector source. The interactive part of our system still needs to be improved but we gained a lot of insight on how to proceed there. As future work we will look into using direct scores from concept detectors and will improve our user interface further.

References

[1] R. B. N. Aly, D. Hiemstra, and R. J. F. Ordelman. Building detectors to support searches on combined semantic concepts. In *Proceedings of the Multimedia Information Retrieval Workshop, Amsterdam, The Netherlands*, pages 40–45, Amsterdam, August 2007. Yahoo! Research.

[2] Christiane Fellbaum. *Wordnet: An Electronic Lexical Database*. The MIT Press, 1998.

[3] C. Hauff, R. B. N. Aly, and D. Hiemstra. The effectiveness of concept based search for video retrieval. In *Workshop Information Retrieval (FGIR 2007), Halle,*

Germany, volume 2007 of *LWA 2007 Lernen - Wissen Adaption*, pages 205–212, Halle-Wittenberg, 2007. Gesellschaft fuer Informatik.

[4] A.G. Hauptmann, R. Baron, M. Christel, R. Conescu, J. Gao, Q. Jin, W.-H. Lin, J.-Y. Pan, S. M. Stevens, R. Yan, J. Yang, and Y. Zhang. Cmu informedias TRECVID 2005 skirmishes. In *Proceedings of the 3rd TRECVID Workshop*, 2006.

[5] Djoerd Hiemstra, Henning Rode, R. van Os, and J. Flokstra. Pftijah: text search in an xml database system. In *Proceedings of the 2nd International Workshop on Open Source Information Retrieval (OSIR), Seattle, WA, USA*, pages 12–17. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2006.

[6] Marijn Huijbregts, Roeland Ordelman, and Franciska de Jong. Annotation of heterogeneous multimedia content using automatic speech recognition. In *Proceedings of the second international conference on Semantics And digital Media Technologies (SAMT)*, Lecture Notes in Computer Science, Berlin, December 2007. Springer Verlag.

[7] C. Munteanu, R. Baecker, G. Penn, E. Toms, and D. James. The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives. In *Proceedings of CHI 2006*, pages 493–502, Montreal, Quebec, Canada, April 22-27 2006.

[8] A. Ranjan, R. Balakishnan, and M. Chignell. Searching in audio: the utility of transcripts, dichotic presentation and time-compression. In *Proceedings of CHI 2006*, 2006.

[9] Nicu Sebe. The state of the art in image and video retrieval. In *Image and Video Retrieval*, volume Volume 2728/2003, pages 1–8. Springer Berlin / Heidelberg, 2003.

[10] Cees G. M. Snoek, Marcel Worring, Jan C. van Gemert, Jan-Mark Geusebroek, and Arnold W. M. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *MULTIMEDIA ’06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430, New York, NY, USA, 2006. ACM Press.

[11] Rong Yan. *Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval*. PhD thesis, Canegie Mellon University, 2006.