Diversifying Search Results Using Semantic Resources

Cristian Neamțu
"Alexandru Ioan Cuza" University,
Faculty of Computer Science
General Berthelot, No. 16
cristian.neamtu@info.uaic.ro

Adrian Iftene
"Alexandru Ioan Cuza" University,
Faculty of Computer Science
General Berthelot, No. 16
adiftene@info.uaic.ro

ABSTRACT
In the last years, multimedia content has grown increasingly over the Internet, especially in social networks (like Facebook or Flickr), where users often post images using their mobile devices. In these networks, the content is later used in search operations, when some users want to find something using a specific query. Nowadays, searching into these networks is primarily made using the title, description and the keywords associated to resources added by users that have posted the content. In this paper we address the problem of query ambiguity. The usage of semantic resources, like ConceptNet and DBpedia ontologies, has been proven to retrieve a better set of results.

Author Keywords
Image Retrieval; Search Diversification; Semantic Resources; ConceptNet; DBpedia.

ACM Classification Keywords
H5.2. Information interfaces and presentation; H3.3. Information Search and Retrieval.

INTRODUCTION

Over time, various theories involving search results diversification [11] have been developed, theories that have been taken into consideration [5]: (i) content [7], i.e. how different are the results to each other, (ii) novelty [3, 4], i.e. what does the new result offer in addition to the previous ones, and (iii) semantic coverage [16], i.e. how well covered are the different interpretations of the user query. The problem of query ambiguity was approached in [2, 13 and 14].

Capannini et al. [2] approached the problem by mining the query log for specializations of a newly submitted query. The query log is divided into sets of possible user sessions; later these sets are further refined in sessions by finding chains of linked queries. The system has been evaluated using the metrics and the datasets provided for the TREC 2009 Web Track's Diversity Task. In terms of efficiency the system performs faster than its competitors and, in terms of effectiveness, outperforms IA-Select [1] and shows comparable performance with XQuAD [15].

Navigli et al. [14] approached the problem by constructing a set of configurations as follows: for each word from the original query, the system selects a sense and then builds a semantic network for that sense and adds it to the configuration. The semantic network is built by extracting some elements (synonyms, hyponyms, homonyms, etc.) from the WordNet lexical database [6] and from finding words that co-occur by running a natural language processor on the SemCor annotated corpus [12]. At the end, the system returns the configuration with the highest score, obtained by counting the number of common nodes across the semantic networks. They devised an experiment in order to test five different sense-based expansion methods and all of them show an improvement over the plain query-methods.

Minkoo et al. [13] proposed a system that captures the user query concepts. They approach the problem by building a set of rules of related concepts, using top-down refinement strategy, where the rules are represented as an and-or tree. Next, the tree is transformed into to a neural network with the same topology and then applies the back propagation algorithm in order to adjust the weights based on the user's relevance feedback. A set of rules is determined by a fuzzy evaluation of the tree (with the updated weights) and finally these rules are used in an extended Boolean retrieval model.

In a recent paper from 2014, the authors build a novel image retrieval framework that performs a semantic interpretation of the user queries and returns a diversified and accurate result set [8].

OUR MODEL

In this section the system's workflow is described along with its main components. The system has been designed to expand ambiguous queries, in order to obtain different types of results. Thus, the initial query is expanded with ConceptNet or DBpedia ontologies, then a set of images are retrieved from Flickr, finally these are cached and are presented to the end-user.

Query processing

Given a query q from a user, the system builds a set of categories (clusters of related entities that are annotated with a common, broader concept). At the beginning, the initial query is passed to the ConceptNet sub-module; if it cannot obtain any relevant data, the initial query will be passed to the DBpedia sub-module. In order to decide whether the DBpedia sub-module should be used, the system analyzes the number of categories and the total number of elements within each cluster received from the ConceptNet sub-module.

ConceptNet

For a given query *q*, the sub-module removes the stop-words only from the beginning and the end of the query due to the way the data is processed and stored in the ConceptNet ontology. Then it retrieves from ConceptNet [10] a list of relations between different concepts, where at least one of them is related to *q*.

Next, the system extracts from each relation, the more general concept based on its type and stores it temporarily. At the next step, for each extracted relation, the system finds all it's the specializations and then removes the elements that are considered too long. The system considers a query too long if it has a large number of words (5-6) or total length of the string is greater than a threshold). These queries are removed because they may not produce any results when submitted to Flickr.

At the next step, the system calculates the similarity coefficient between two clusters, using the Jaccard Index [9], and merges the similar clusters. Lastly, the system removes the all clusters that do not provide new data (i.e. a large part of the elements can be found in larger clusters). The remaining nodes will be used in the image retrieval process.

Given the query *gnu* has at least two senses: the first one is *a genus of antelopes*, the other one being *operating system*. The system retrieves from ConceptNet the relations and the associated concepts. Next it starts building the clusters based on the relations (ignoring longer elements e.g. "gnu large African antelope have…"). For this query the system cannot find any clusters that can be merged however there are a few small clusters that will be removed, although they might be relevant (e.g. the cluster *aircraft* contains the element S*opwith Gnu*). At the end the module returns three clusters: *mammal* which contains two subspecies, *software* (contains a list of programs related to the gnu operating system) and *organization* (folk group and another element related to the Free Software Movement).

DBpedia

Using the DBpedia Lookup[1] web service the system queries for a list of entities that might be related to the initial query *q*. Each one of the returned elements contains: a label, an URI, a description and a list of categories.

Next, these elements must be filtered because it may contain concepts that are not relevant to the query. A slightly modified version of the Tf-Idf [11] score is used to filter the non-relevant elements, as follows: first, the system builds a corpus from the definitions of every element, then for each element, the system calculates the Tf-Idf weight vector for each of its categories. For words that are very common, the Tf-Idf score will return a special value instead of 0. Finally, if the number of vectors that are zero exceeds a threshold it removes the element.

[1] DBpedia Lookup: https://github.com/dbpedia/lookup

For example, for the query *android*, some of the concepts obtained from DBpedia Lookup are*: Android (operating system)*, *Android (robot)* and *Testosterone*. After filtering the irrelevant elements, the underlined concept is removed.

For the remaining elements the system will first search among its categories for one that has a label similar to the one of the element. (i) If found, system will run a query to obtain specializations for that resource. (ii) Otherwise, the system will attempt to identify a set of common concepts by running a query for each category of an element, intersecting the results and keeping those elements that appear frequently.

Given the query *microprocessor* the system attempts to build a set of clusters using the data from ConceptNet. The module will not produce any useful results because majority of the clusters are small. Next the system will obtain from the DBpedia Lookup web service the following elements: Central Processing Unit, Embedded system, Microprocessor, 32-bit, CPU Cache, Microcontroller, Instruction set, ARM architecture and Motorola 68000. In the next step, it will keep only the elements that are considered relevant to the query and will expand them. The end result produced by the module is **Central Processing Unit** (Dual-voltage CPU, Tag RAM, etc.), **Embedded system** (Logic analyzer, Debit card, etc.), **Microprocessor** (KOMDIV-64, Apple A4, etc.), **CPU Cache** (Smart Cache, Tag RAM, etc.), **Microcontroller** (Intel MCS-48, Motorola MC14500B, etc.), **Instruction set** (Berkeley RISC, MIPS-X, etc.), **ARM architecture** (Tegra, ARM Cortex-A15, etc.) and **Motorola 68000** (Motorola 68000).

Image processing

This module receives a category previously built by the query expansion modules and returns a set of relevant images for it. Depending on the size of the built category (cluster), this module will apply one of the two different methods.

The first method is applied only to small clusters (up to 6 elements). For each element of such a cluster the module will perform a search on Flickr to obtain a set of photos whose title, description or tags contain the specified text. The other method is applied to large clusters. These will be divided into smaller sets. Next, for each set, it will perform a new search in order to obtain a set of pictures whose tag list contains at least one of the elements from that set.

Next, the images will be grouped by the user's id and the system will cluster the elements inside a non-singleton group based on the title using the edit-distance and it will select a single image from each cluster. Finally the system will select the top K images from the previous step.

CASE STUDIES

Case Study 1 – Query *Formula One Racer*

Given the query *formula one racer*, the system returns a single cluster containing approximately 70 elements (see

Figure 1) where four of them are not related to the query. For instance: *Mike Park* does not belong to the cluster (he is an American musician).

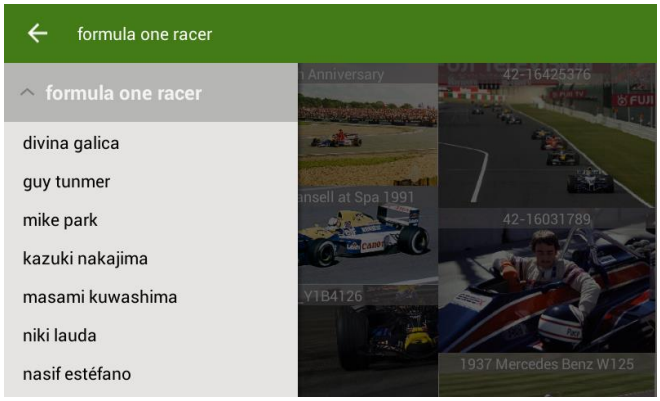

Figure 1. A few elements from the returned cluster.

In Figure 2 is presented a subset of the results returned by the system. A fraction of the images returned are not relevant to the initial query, but are related to the career of that person. For example, David Brabham was a formula one racer, however he worked in other fields of motorsports such as: Touring Car Championships and Le Mans competitions). A relevant example for this case would be the image in the lower right corner of the Figure 2.

Figure 2. A small subset of the images returned after processing the query *formula one racer*.

Case Study 2 – Query *Eiffel*

After submitting the query *Eiffel*, the system returns 7 clusters, all of them are presented in Figure 3: The finer grained clusters are: *Architectural Structure*, *Band*, *Programming Language* and *Film*. The rest of the clusters have a greater degree of generality and contain elements from the other clusters. For example: the cluster *Organization* contains some of the elements from the cluster *Band*.

The cluster *Architectural Structure* contains the following elements: *Eiffel Tower*, *Eiffel Bridge Zrenjanin*, *Eiffel Bridge Ungheni* and *Ponte Eiffel*. Below is presented a subset of the images returned for this cluster.

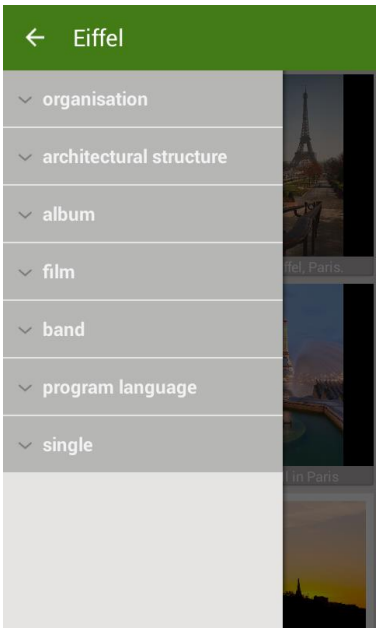

Figure 3. The clusters returned for the query Eiffel.

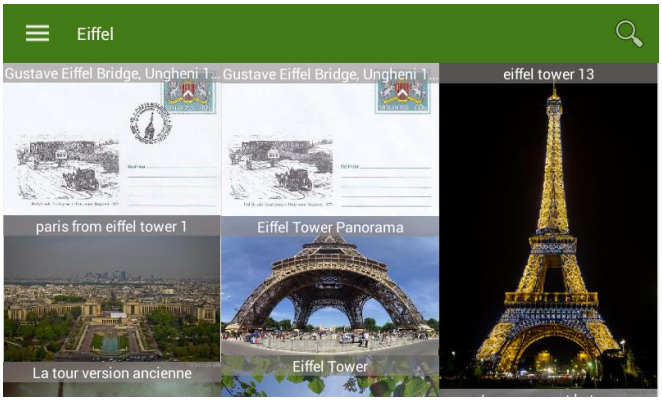

Figure 4. The results for the *Architectural Structure* category.

The system has been built to expand ambiguous queries that are generally short, for which the system returns relevant results. However, submitting more complex and elaborate queries may result in a failure to return a set of images.

We attempted to tokenize the user query, clear all the stop-words and try to find a set of linked concepts based on the processed query, but it resulted in set concepts that could not be used in the current context.

EVALUATION

In this section the system will be evaluated from three different points: (1) *the time required to process a query* (with and without caching), (2) *the relevance of the items inside a cluster* and (3) *the relevance of the images returned by the system*. In order to evaluate the system, there were considered 10 queries.

The time required to process a previously submitted query is on average 4 seconds, while the time required to process a newly submitted query may take up to 60 seconds. The execution time depends on the number of elements in a

cluster (retrieving images for smaller clusters may take up to 10 seconds, while bigger clusters take a significantly longer time).

Regarding the relevance of the elements inside a cluster, results show that approximately 70% of the clusters elements are related to the initial query. In a relevant cluster, up to 80% of the items are relevant to initial query and are relevant for the current cluster.

After analyzing the set of images returned by processing a cluster it results that on average 65% of the images are relevant to the query. In general, the relevant images are among the first results; however we found some exceptions due to the fact that the cluster contains some irrelevant items positioned at the beginning. We remark that, there are some cases in which the system cannot obtain, due to some limitations, a relevant set of images.

For instance, when "*a view from Eiffel Tower*" is used as initial query, the system identifies two meanings: (i) *the panorama of Paris from the top of the Eiffel Tower* and (ii) *the movie*. The system tries to retrieve from Flickr a set of images for (ii) the second meaning of the query, but it receives results for the first sense.

CONCLUSIONS

This article presents a method in which semantic resources like ConceptNet and DBpedia are used to improve the quality of searches performed in an image retrieval system. Until now, the results are promising and we see how 70% of the clusters elements and up to 80% of the items are relevant to initial query.

In the future, we want to improve the current system by combining semantic resources ConceptNet and DbPedia (now the system uses them separately). Also, we want to replace the Tf-Idf score with other function.

ACKNOWLEDGMENTS

The research presented in this paper was funded by the project MUCKE (Multimedia and User Credibility Knowledge Extraction), number 2, CHIST-ERA/ 01.10.2012.

REFERENCES

1. Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. Diversifying search results. In *WSDM '09*, (2009), 5–14.
2. Capannini, G., Nardini, F. M., Perego, R. and Silvestri, F. Efficient Diversification of Web Search Results. In *Proc. VLDB Endow*., 4(7), (2011), 451-459.
3. Carbonell, J. G. and Goldstein, J. The use of mmr, diversity-based re-ranking for reordering documents and producing summaries. In *SIGIR*, (1998), 335-336.
4. Clarke, C. L. A., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Bttcher, S. and MacKinnon, I. Novelty and diversity in information retrieval evaluation. In *SIGIR (2008)*, (2008), 659-666.
5. Drosou, M. and Pitoura, E. Search Result Diversification. In *SIGMOD Record*, 39(1), (2010), 41-47.
6. Fellbaum, C. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 1998.
7. Gollapudi, S. and Sharma, A. An axiomatic approach for result diversification. In *WWW'2009*, (2009), 381-390.
8. Iftene, A. and Alboaie, L. Diversification in an image retrieval system. In *IMCS-50. The Third Conference of Mathematical Society of the Republic of Moldova dedicated to the 50th anniversary of the foundation of the Institute of Mathematics and Computer Science*. (2014).
9. Jaccard, P. *The distribution of the flora in the alpine zone*. New Phytologist, 1912.
10. Liu, H. and Singh, P. Focusing on ConceptNet's natural language knowledge representation. In *Commonsense Reasoning in and over Natural Language Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'2004)*, (2004).
11. Manning, C.D., Raghavan, P. and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press New York, NY, USA, 2008.
12. Miller, G. A., Chodorow, M., Landes, S., Leacock, C. and Thomas, R. G. Using a Semantic Concordance for Sense Identification. In *Proceedings of ARPA Human Language Technology Workshop*, (1994).
13. Minkoo, K. and Vijay, R. Adaptive Concept-based Retrieval Using a Neural Network. In *Proceedings of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*, Athens, Greece, (2000).
14. Navigli, R. and Belardi, V. An Analysis of Ontology-based Query Expansion Strategies. In *Workshop on Adaptive Text Extraction and Mining (ATEM), ECML 2003*, (2003).
15. Santos, R. L. T., Macdonald, C. and Ounis, I. Exploiting query reformulations for web search result diversification. In *WWW'10*, (2010), 881–890.
16. Zheng, W., Wang, X., Fang, H. and Cheng, H. Coverage-based search result diversification. In *Journal IR* (2012), 433-457.