Conceptualizer: a Method for Suggesting Concepts

Marcel Goksun, Gijs Pannebakker, Nicholas Piël, and Xandra van de Putte
Univerity of Amsterdam

Abstract

In this paper we discuss two methods for suggesting automatically detected semantic concepts used for searching through news video archives. It is based on the 436 concepts that are used in the MediaMill search engine. Given a query, a relevant word can be given as input and our methods suggest a list of concepts that can be useful for the search. The results are evaluated and compared with the current concept suggestion engine of MediaMill. The results show that using some simple methods, a concept suggestion engine can do quite well.

Keywords: Video retrieval, concept similarity.

1 INTRODUCTION

Every year, TREC organizes a benchmark for video retrieval (TRECVID). The goal is to encourage research in information retrieval by providing a large test collection, uniform scoring procedures and comparing the results. All teams that participate in the benchmark get a large news video collection and a number of queries. The task is to find as much as possible shots in fifteen minutes that are relevant for the query and sort them from most relevant to less relevant. The MediaMill team (www.mediamill.nl) participated in this benchmark since 2004. In their used engine, 436 automatically detected semantic concepts are used. Because it is difficult to find the relevant concepts to search for when some query is given, a suggestion engine has been implemented. This engine automatically detects some relevant words from the textual query and search for meaningful concepts. The problem with this engine is that it is too slow. Because the task for the TRECVID benchmark is to find as much as possible relevant shots in 10 minutes, this engine cannot be used. To date, most of the concepts can be remembered, but every year the number of concepts will grow very fast. It is therefore important that a fast suggestion engine will be implemented. In this paper we suggest an engine which uses simple methods to quickly suggest relevant concepts. We first discuss some related work which include the current concept suggestion engine of MediaMill. In section 3 we describe our methods and the results are shown in section 4. The paper ends with a conclusion and future work recommendations.

2 RELATED WORK

The current suggestion engine of MediaMill works as follows. First it assigns syntactic categories to groups of words in the input text using a chunking algorithm. Then a grammatical classification to each word is assigned by using a part-of-speech tagger and each noun chunk is being searched in WordNet [1]. When a noun has been found in WordNet, it is eliminated from further lookups. After that, any remaining nouns are looked up. In this stage, both the concepts in the text and the multimedia concepts are related to WordNet, so the semantic distance between these concepts can be computed. To compute this distance it uses Resnik's algorithm [2] which calculates the similarity of a concept to each of the found WordNet nouns from the query text. Finally the combined scores are used to rank each multimedia concepts in order of expected utility. [3]. Another functionality of this engine is that it also suggests concepts based on query image analysis. Because this is beyond the scope of our research project, we will only focus on the textual method to derive the relevant concepts.

3 METHODOLOGY

Like the current concept suggestion engine of MediaMill, we also mapped the 436 multimedia concepts to nouns in WordNet. We did this by transform-

ing the concepts names into possible word groups and then match them against the WordNet lexicon. When no results where found we used the porter stemmer to simplify the search query, for example "boats" will be stemmed to "boat". After this automatic tagging there where still a lot concepts with wrong connections or no connection at all. In order to add and or change these connections we implemented a graphical user interface (http://nichol.as:4321/admin) which allows easy CRUD operations on the data. The interface has been implemented in Pylons, a Python webframework [4], which follows the MVC (Model View Controller) methodology [5] and allows you to rapidly build web applications with an AJAX interface [6]. The use of AJAX allows for fast interaction on the relatively large database since only relevant information is being fetched from the server. On top of this concept to WordNet database we have implemented four search methods. The search engines using these methods can be found at http://nichol.as:4321/search. They are described in the following sections.

3.1 Exact Search

The simplest implementation is what we call 'exact search' while it name implies total preciseness it's results are actually a bit more fuzzy. The exact search looks up a query in WordNet and tries to match all found synsets in the concept database. The result is that when looking for something this algorithm will also search for synonyms and different interpretations of the query. For example, when the search query is 'man' it will respond with 'military_personnel', 'single_person_male', 'male' and 'male_person', which are all relevant concepts for the query.

3.2 TFIDF Search

Our Term Frequency Inverse Document Frequency algorithm is based on the glosses found in WordNet. These are small descriptions of the synsets, containing about 15 words on average. Using TFIDF weighting, we measured the cosine distance between the featurevectors of the combined glosses of every word in the query and the combined glosses of the set of synsets linked to a concept detector. This similarity measure was done for every concept detector. The detectors with a similarity score above the threshold of 0.25 are shown in the output list. This threshold was chosen as it seemed to make the best tradeoff between recall and precision. Before the featurevectors were extracted from the glosses, minimal stemming was done by deleting most non-alphanumericals (as in quotes, semicolons, etc).
When for example searching for 'party', the result list includes 'host', 'speaker_at_podium', and 'celebration_or_party'. Because for example 'host' in WordNet is described as 'a person who invites guest to a social event (such as a party in his or her own home)...', which is very similar to the description of party ('an occasion on which people can assemble for social interaction and entertainment'), this will be added to the result list.

3.3 Andes Search

The Andes Search is an extension of the exact search which not only searches for an exact synsets but also for its Ancestors (inherited hypernyms) and Descendants (full hyponyms). Since all synsets of the concepts are known beforehand the results of this method can be returned almost instantly. The philosophy behind this approach is that the initial query already determines the preciseness of the concept you are looking for. I.e., when searching for 'tiger' you are looking for that specific 'big cat' you are not interested in 'jaguars' or 'lions' since then your initial query would have been 'fierce animals' or 'big cat'. The result is that when your searching for 'bus' your result will be 'bus', 'car', 'ground vehicles', 'vehicles'. However when you search for 'vehicle' you will find ALL vehicles so it will also return 'motorcycle', 'airplane', 'boat' etc.. A general usage remark would thus be to always make your query as specific as you desire and let the algorithm do its work to come up with the correct results.

3.4 Combination of Flickr and Andes Search

While the Andes algorithm works great in most cases and its philosophy seems to catch most concepts, it sometimes does not. For example when searching for 'vegetation', a good concept detector is actually 'desert' even though its literal meaning is actually the exact opposite, namely the lack of vegetation. In shots depicting deserts, there is most of the time some vegetation in it. In order to tackle this problem we make use of 'Flickr Tags'. Flickr is

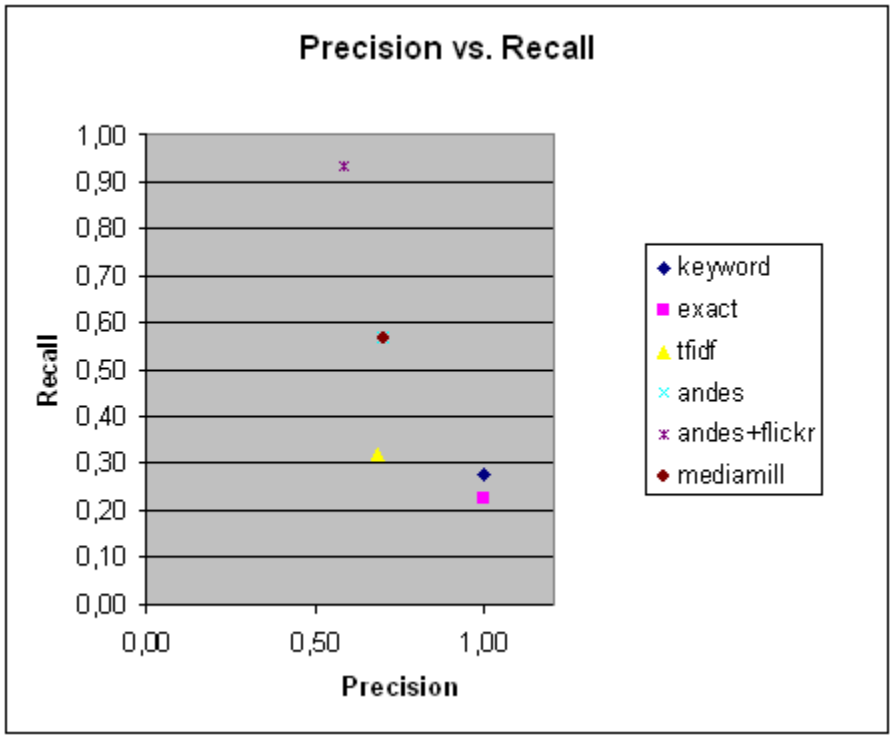

Figure 1: Recall and precision for the evaluated concept suggestion engines

a community website with over 10.000.000 pictures [7]. Most of these pictures have been tagged. When given a tag one can search the Flickr website for all pictures matching that tag and look up the tags each photo has been tagged with. The tags that occur multiple times in combination with your tag can be considered relevant. Luckily Flickr comes with a great API which already implemented this feature so we did not need to dig for the relevant tags ourselves. The implementation of the search method is as follows, it does an Andes search on the query and for each relevant tag according to flickr it does an exact search. The result of these queries is being combined. In order to speed up this method we tried to cache most of the possible queries to flickr beforehand. We only access the flickr service when a certain query cannot be found in the cache. In order to make the results from flickr more relevant we use a blacklist on which the most popular tags can be found for example 'dog', 'fun', 'girl', 'house', 'summer'. Using flickr does not only broaden your search but also makes the system more robust, ie when the user makes a typo in it's initial query it is very likely that some photos have this same typo in their tag which allows the algorithm to still return relevant results.

4 RESULTS

To evaluate our methods, we have chosen 5 queries from the queries that were used for the TRECVID benchmark 2006. From them, we picked single words that we thought should give the best concepts in which to perform the search. For each query we manually selected all concepts that would be relevant to search for given this query. One example of the queries we tested was "Find shots with one or more people leaving or entering a vehicle". We expected a list containing all types of vehicles and every concept with a part of its name being a vehicle (for example 'airplane_take_off'). We compared the results of each of the search engines with these lists. The results are also compared with the current concept suggestion engine of MediaMill. The evaluation measured we used are precision and recall. The results are shown in figure 1. The lists

of manually selected concepts were made by using our common sense. It is very difficult to implement 'common sense' into an engine, but because Flickr images are tagged by normal people, 'common sense' is somehow provided to the Flickr-Andes combination search. This is showed by the result list of this search engine; there were some relevant concepts that we did not think of when selecting the multimedia concepts for a given query.

5 FUTURE WORK

When using the Exact Search or the Andes Search, queries consisting of words in plural form will not give any results. This is because we did not yet stem the initial query. So the first task for these two searching methods would be to add a stemming algorithm. Also, stemming the glosses of the words in WordNet should increase the recall and precision of the TFIDF Searching method.

Another task is to cache the results. This way, the performance should improve even more, which is vital when multiple users are accessing the conceptualizer.

The tags obtained by Flickr could be filtered by using inversed document frequency instead of the blacklist of Flickr. This might improve the results, because the blacklist of Flickr is made for a specific domain. Tests must be done for determining this assumption. Because the output list can sometimes be quite long (15+ items), a simple but practical improvement would be to sort them. This could be done by using some simple similarity algorithm, the precision of the concept detector, or a combination of both.

6 CONCLUSION

This experiment suggests that using some simple methods, a concept suggestion engine can do quite well. For example, the results showed that for the highest recall, the combination of Andes and Flickr Search is the best searching method. When using sorted lists, the somewhat low precision would not be a problem. The most irrelevant results would be at the bottom of the list and the user just has to select the desirable items. The Andes algorithm showed that it is not necessary to calculate some complex similarity algorithm, as has been done in the concept suggestion engine of MediaMill. This can make the time to calculate the result list quite shorter.

References

[1] G.A. Miller, "WordNet: a lexical database for English", *Communications of the ACM*, 1995, Vol. 38, no. 11, pp. 39-41.

[2] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy", *In Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI'95*, Montreal, Canada, 1995, pp. 448-453.

[3] M. Worring, C.G.M. Snoek, B. Huurnink, J. van Gemert, D. Koelma & O. de Rooij, "The MediaMill Large-lexicon Concept Suggestion Engine", *Proceedings of the ACM International Conference on Multimedia*, October 2006, pp. 785-786.

[4] Pylons Rapid Web Development, http://pylonshq.com/, 2006.

[5] Wikipedia, "Model-view-controller — Wikipedia, The Free Encyclopedia", http://en.wikipedia.org/w/index.php?title=Model-view-controller&oldid=93760276, 2006.

[6] Asynchronous JavaScript and XML (AJAX), http://developer.mozilla.org/en/docs/AJAX.

[7] Flickr: Online photo management and sharing application, http://www.flickr.com, 2006.