DEU at ImageCLEF 2009 WikipediaMM Task: Experiments with Expansion and Reranking Approaches

Deniz KILINÇ, Adil ALPKOCAK
Dokuz Eylül University, Department of Computer Engineering
{dkilinc, alpkocak}@cs.deu.edu.tr

Abstract

This paper describes participation of Dokuz Eylül University to WikipediaMM task at ImageCLEF2009. This year we concentrated on two main topics: First is about expansion of native document, term phrase selection and query expansion processes which is based on WordNet, WSD and WordNet similarity functions. The second is a new reranking approach with Boolean retrieval and C^{3}M based clustering. Experimentation shows that reranking generated the best MAP and precision results among all participants in WikipediaMM 2009 task.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Performance evaluation; H.2.3 [**Database Management**]: Languages

General Terms

Reranking, Clustering, Information Retrieval, Vector Space Model

Keywords

Reranking, Clustering, WordNet, WSD, Query Expansion, WordNet Similarity, Term Phrase Selection, Boolean Retrieval, Information Retrieval, Vector Space Model

1 Introduction

This paper presents details of our participation to the WikipediaMM task of ImageCLEF 2009. This is our first year in WikipediaMM task, and we participated both retrieval experiment and relevance assessment steps. We propose expansion and reranking approaches. Reranking is used to re-order the initial retrieved documents for better results by increasing the precision. Reranking is recently most popular research subject for Information Retrieval. We also used expansion techniques for both dataset and queries. Although there are numerous works on query expansion, document expansion is one of the major proposed novel approaches. Expansion phase is implemented using WordNet [1] (WSD, WN Similarity Functions). During the baseline retrieval, expanded and original datasets are combined with the form of Pivoted Unique Normalization [9].

The main focus of this work is to improve search results by two phased reranking. The set of initial retrieved documents are re-ordered for better results by increasing the precision. The first phase comprises reranking and reordering with the Boolean retrieval approach. The main objective of the second step is reranking with the clustering algorithm. C^{3}M [23] clustering algorithm is executed on the new result sets and the similarity score of each document with its related query is calculated with C matrix (c_{ij}). Final ranking score $R_{clustering}$ is calculated by using Boolean ranking score ($R_{boolean}$) and query-document similarity score($R_{CCSimScore}$). Experimental results show that, phased reranking approach improves results over the baseline and over expansion results.

Rest of the paper is organized as follows: Sections 2 gives the details of our retrieval system. In Section 3, we present, Document and Query expansion methods, we have tested, by using WordNet system. Section 4 gives our two phased reranking approaches based on Boolean retrieval and C^{3}M clustering. Section 5 concludes the paper, discusses the results we obtained and gives a look at the future studies on this subject.

2 Retrieval Framework

Figure 1 shows the retrieval system framework and our experimentations. First of all, preprocessing step is done; dataset is expanded using WordNet (WSD, WN Similarity Functions) and term phrases are selected. Both the original and the expanded form of dataset are used and converted to document vectors before baseline retrieval.

Queries can also be expanded using TPS and/or WordNet for experimental purposes. During the baseline retrieval, expanded and original datasets are combined with the form of Pivoted Unique Normalization (Pivoted VSM). Each baseline query resultset is kept (new Pivoted VSMs) and two phased reranking steps start. The first phase aims reranking and reordering with the boolean retrieval approach. The resultsets of each query and the base ranking scores are again saved for the next reranking step. The main objective of the second step is reranking with the clustering algorithm. Final ranking score $R_{clustering}$ is calculated using boolean ranking score ($R_{boolean}$) from the first step and query-document similarity score($R_{CCSimScore}$) from the second step. The two phased reranking process is completed and final ranked resultsets are generated.

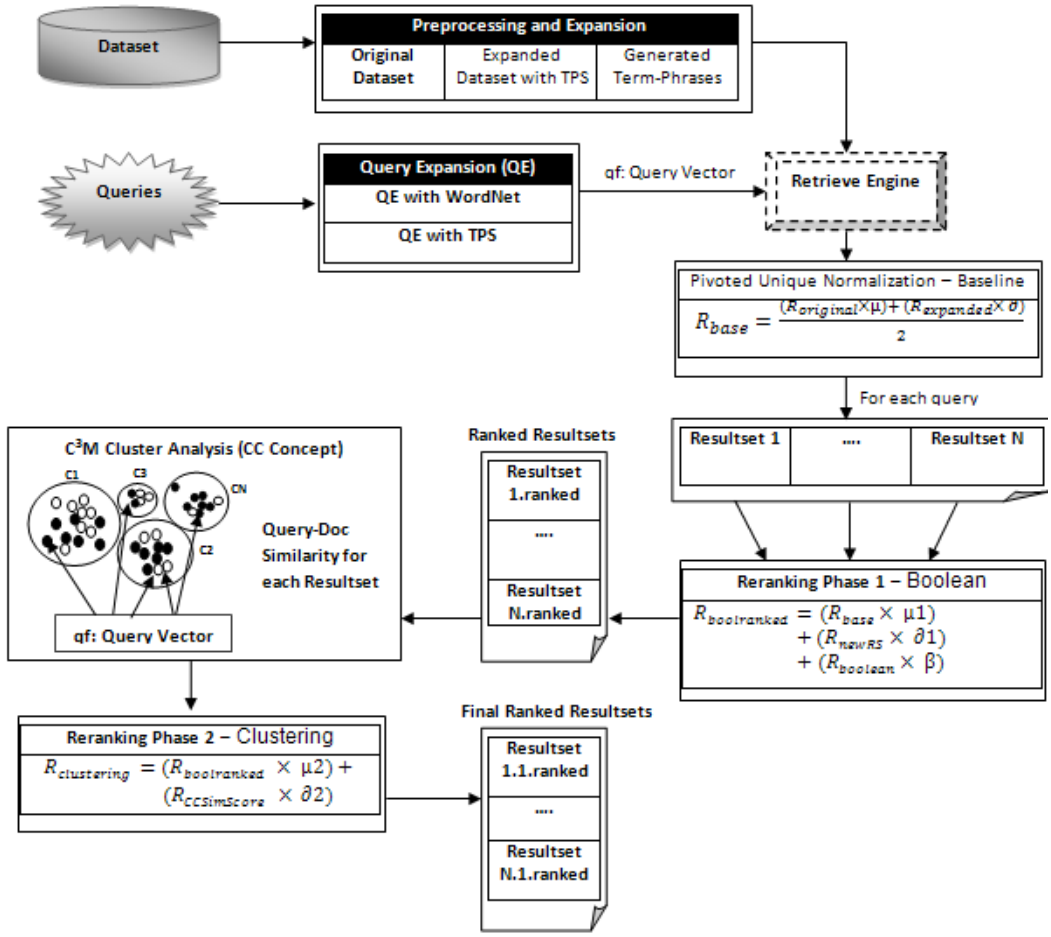

Figure 1 - Retrieval Framework Overview

2.1 Term Weighting and Normalization

Term weighting is an important aspect of modern text retrieval systems [10]. There are three major parts that affects the importance of a term in a text, which are the term frequency factor (tf), the inverse document frequency factor (idf), and document length normalization. Cosine normalization is the mostly used normalization technique in the vector space model [8]. Normalization factor is computed as in the formula (1),

$$\sqrt{w_1^2 + w_2^2 + \ldots + w_t^2} \tag{1}$$

where each w equals $(tf \times idf)$. Since the lengths of the document vectors are converted into unit vectors, the information content is deformed for longer documents, which contain more terms with higher *tf* values and also more distinct terms.

In this study we used Pivoted Unique Normalization [9] which is a modified version of classical cosine normalization. A normalization factor is added to the formula which is independent from term and document frequencies. Since working dataset has many longer documents and these documents are also expanded in the

DE phase, Pivoted Unique Normalization affects the retrieval performance positively by increasing recall. This work purposes to improve search results by reranking. If the recall level of the retrieval process is not high, reranking becomes non-effective.

$$w_{ij} = \frac{\log(dtf) + 1}{sumdtf} \times \frac{U}{1 + 0.0118U} \times \log\left(\frac{N - nf}{nf}\right) \quad (2)$$

where *dtf* is the number of times the term appears in the document, *sumdtf* is the sum of (log(dtf)+1)'s for all terms in the same document, *N* is the total number of documents, *nf* is the number of documents that contain the term, *U* is the number of unique terms in the document. The uniqueness means that the measure of document length is based on the unique terms in the document. In this work, 0.0118 is used as the pivot value.

When retrieving, the rank is the product of the weight and the frequency of the term in the query.

$$R = \sum_{i=1}^{n} (w_{ij} \times qf_i) \quad (3)$$

where *n* is the number of term in the query, w_{ij} is the weight and qf_i is the count of term in the query.

3 Preprocessing and Expansion Phase

In our work, document expansion (DE), term phrase selection (TPS) and query expansion (QE) phases are realized by using WordNet [1] system which is an on-line lexical reference system developed by a group of people leaded by George Miller at the Cognitive Science Laboratory at Princeton University. WordNet attempts to model the lexical knowledge of English and can also be seen as ontology for natural Language terms. It contains nearly 100,000 terms, divided into four taxonomic hierarchies; nouns, verbs, adjectives and adverbs.

The first stage of preprocessing is the expansion stage. Although only query expansion is mostly common in the text retrieval, in this work, both the documents and queries are expanded using the same approaches. Namely, the word "expansion", is used for both the document expansion (DE) and query expansion (QE).

Expansion is realized with Word sense disambiguation (WSD), so that the terms are expanded, with the most appropriate sense, based on the context in which they occur. Word sense disambiguation (WSD) is the process of finding out the most appropriate sense of a word based on the context in which it occurs. The Lesk algorithm [3] disambiguates a target word by selecting the sense whose dictionary gloss shares the largest number of words with the glosses of neighboring words. Since numerous senses exists in different domains for a single term, expanding the term with all of these senses results in noisy and exhaustive documents and queries. By selecting the most appropriate sense with WSD, unnecessary expansions are prevented.

Before the expansion stage, both the documents and queries are processed through some phases. The first phase is stop-words elimination. The stop words in the documents and queries are eliminated by controlling each term's existence in the stop-words dictionary. The second phase is lemmatizing. Lemmatization is the process of reducing an inflected spelling to its lexical root or lemma form. The lemma form is the base form or head word form that can be found in the WordNet dictionary. The combination of the lemma form with its POS is called the lexeme.

Although it is commonly argued that language semantics are mostly captured by nouns and noun term-phrases, in this work, both noun and adjective representations (POS) are used in the documents and queries. After all of these phases the documents and queries are available for the expansion process. To emphasize that in the expansion stage, the new terms are also processed through these phases. If they pass successfully through them, then they are added to the documents and queries.

In general, the documents in the datasets are domain-specific. On the other hand, Wiki has a heterogeneous structure [11][12]. It contains nearly 150.000 documents, created by the different users for different aims. However queries are more target focused and they are created by the users according to their needs without knowing the documents. The aim of expanding both documents in the dataset and the queries is, to assimilate the queries to the documents, and documents to queries. Expanding the poorly defined documents and adding new terms or term-phrases, results in higher ranking performance, or similarly expanding the queries and widening the search terms, increase the quality of ranking by bringing relevant documents not matching literally with the original user query.

As an example; the document, related with the query *"blue flower"*, includes the term *"sea lavender"*. Without expansion, they are not matching literally and they seem irrelevant. However, as we expand "*sea lavender"* with WordNet, the new terms *"blue flower"* are added to the document. So both the query and the document include the same terms and their ranking score increases.

WSD and WordNet similarity functions decrease the risk of creating exhaustive documents and queries. Especially, the documents contains numerous terms and expanding all of these terms can cause creating exhaustive documents and can also affect ranked results negatively. For this reason, only in DE stage, the original forms of documents are also kept to compensate the ranking weight score during the retrieval process. An example of document expansion and matching the document literally with the query, is given above. And the query expansion stage is illustrated in Figure 2.

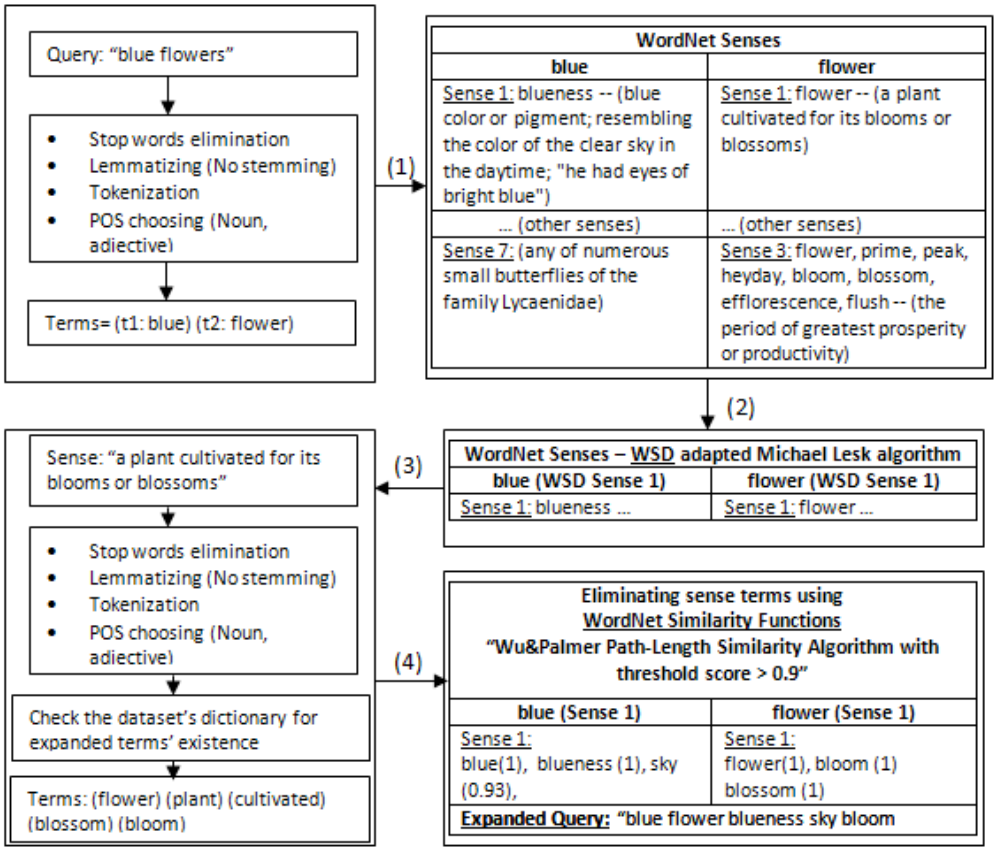

Figure 2: Query Expansion Example

For each expanded term in the query or document, similarity score is calculated. Different methods have been proposed in the literature for determining the semantic similarity between terms [2, 4, 5, 6, 7]. In our work we used Wu and Palmer [2]'s edge counting method is used for term similarity measure. The terms, above a specific threshold value, are added to the final document or query. Besides, the threshold values for adjective terms and noun terms are determined differently. For this reason, in the (4) formula, the threshold value for noun terms are 0.9, and the threshold for adjective terms are 0.7.

$$t_{QDE(i,j)} \mathrel{+}= t_{EXP(i,j)} \text{ where, } \forall\, t_{ORG(i)}, (\text{WNSimScore}(t_{EXP(i,j)}, t_{ORG(i)}) > \text{threshold}) \quad (4)$$

where $t_{ORG(i)}$ is a term in the original document or query, $t_{EXP(i,j)}$ is a generated term for $t_{ORG(i)}$ using WordNet and Lesk's WSD[3]. $t_{QDE(i,j)}$ is the final expanded document or query with selected and added new terms.

3.1 Term Phrase Selection (TPS)

TPS phase goes parallel with the expansion phase. While the documents and the queries are being expanded, each twosome terms are controlled in WordNet for existence. If the twosome terms exist in WordNet as a noun-phrase, they are accepted as term-phrases. These terms are added to the both dictionary and the expanded document or query as a new term. In this work, 6,808 term-phrases are generated and added into dictionary for Wiki dataset. For example, a document or a query contains "hunting", "dog" terms sequentially, these two successive tokens are searched as "hunting dog" in WordNet. If this phrase exists in WordNet, the document or

query is expanded with the term "hunting-dog". And finally the term phrase is added to the term phrase dictionary. TPS idea is showed in the equation (5);

$$TP = \begin{cases} 1, & \text{if } t_i\, t_{i+1} \text{ EXISTS in WordNet} \\ 0, & \text{if } t_i\, t_{i+1} \text{ does not EXISTS in WordNet} \end{cases} \quad (5)$$

where, t_i and t_{i+1} represents two successive terms.

3.2 Baseline Retrieval

The equations of final query selection and ranking score calculation with the original and expanded datasets are given below;

$$qf\_final_i = \begin{cases} qf_{i_{original}}, & if\ the\ original\ query\ is\ used \\ qf_{i_{TPS\_expanded}}, & if\ the\ original\ query\ is\ expanded\ with\ TPS \\ qf_{i_{WordNet\_expanded}}, & if\ the\ original\ query\ is\ expanded\ with\ WN \\ qf_{i_{TPS\_and\_WordNet\_expanded}}, & if\ the\ original\ query\ is\ expanded\ \ with\ WN\ and\ TPS \end{cases} \quad (6)$$

Where, qf_final_i is the number of times the term appears in the final query which can be the original query or expanded using WSD and/or TPS .

$$R_{original} = \sum_{i=1}^{n} (w_{ij\ (original\_dataset)} \times qf\_final_i) \quad (7)$$

where, $R_{original}$ is the ranking weight of original document.

$$R_{expanded} = \sum_{i=1}^{n} (w_{ij\ (expanded\_dataset)} \times qf\_final_i\ ) \quad (8)$$

where, $R_{expanded}$ is the ranking weight of expanded document.

$$R_{base} = \frac{(R_{original} \times \mu) + (R_{expanded} \times \partial)}{2} \quad (9)$$

where, μ and ∂ are constant parameters and can be possibly changed for different datasets and queries. For wiki2009 subtask [11] optimal parameters, μ=1 and ∂=0.9 are used.

4 Reranking

Reranking is a methodical technique to reorder the initial retrieved documents for better results by increasing the precision. Basically, relevant documents that have low ranking weights are reweighted and reordered in a retrieved resultset. According to the literature several methods can be used for reranking, such as unsupervised document clustering, semi-supervised document categorization, relevance feedback, probabilistic weighting, collaborative filtering or a combination of them.

Some researchers proposes methods based on clustering, inter document similarity or user supported relevance data for document reranking [16][17][20]. Some has proposed a modification in weighting scheme proposed [13][14][15][18]. Most of the researcher studied on text retrieval. Similarly, an application of Lee's method is performed on image dataset by Park et al. [19]. Image features used in proposed method are color histogram in HSV color space, Gray-scale co-occurrence matrixes and edge histograms.

In our work, we propose a new reranking approach in two phases. After the base retrieval results are generated, the result sets of each query and the base ranking scores (R_{base}) are saved for the reranking phases. The first phase comprises reranking and reordering with the Boolean retrieval approach. The result sets of each query and the base ranking scores (R_{base}) are again saved for the next step after the Boolean retrieval is performed. The main objective of the second step is reranking with the clustering algorithm. Firstly, expanded forms of relevant queries are appended to the end of the Boolean ranked result sets. After that, C^3M clustering algorithm is executed on the new result sets and the similarity score of each document with its related query is calculated with C matrix (c_{ij}). Finally, new reranking score $R_{clustering}$ is combined with Boolean ranking score ($R_{boolean}$) from the first step and with the query-document similarity score($R_{CCSimScore}$) from the second step. The two phased reranking process is completed after the documents are ranked according to this score.

4.1 Reranking with Boolean Retrieval

Boolean retrieval is the first phase for reranking. The Boolean retrieval is a model for information retrieval in which any query can be formulized in the form of a Boolean expression of terms. Query terms are combined with the classical Boolean operators AND, OR, and NOT [24]. In this work, each query's terms are searched for the exact match by keeping the orders. It can be considered as the relational database query operators', LIKE or CONTAINS functionality. Basic Boolean approach is showed in (11);

$$R_{boolean} = \begin{cases} 1, & \text{if document CONTAINS query terms with the exact order} \\ 0, & \text{if document does not CONTAINS query terms with the exact order} \end{cases} \quad (10)$$

Definite calculation for the reranking score is showed in (11)

$$R_{boolranked} = \begin{cases} (R_{base} \times \mu 1) + (R_{newRS} \times \partial 1) + (R_{boolean} \times \beta), & if\ R_{boolean} > 0 \\ (R_{base} \times \mu 1) + (R_{newRS} \times \partial 1), & if\ R_{boolean} < 0 \end{cases} \quad (11)$$

where $R_{base}$is the baseline retrieval score using WordNet (DE, WSD, WNSimScore) and QE, $R_{newRS}$is the new pivoted normalization ranking score after the retrieval process on the new resultset, $R_{boolean}$is boolean ranking score using boolean retrieval (1 or 0), and $R_{boolranked}$ is the calculated ranking score for boolean retrieval phase. $\mu 1$, $\partial 1$and β are constant parameters. In this work, their values are set as 0.8, 1, and 4 respectively for experimental results.

4.2 Reranking with C^{3}M Clustering

Clustering is a method for grouping a set of documents into clusters. The algorithms' goal is to create clusters that are relevant internally, but clearly different from each other. In other words, documents within a cluster should be as similar as possible and documents in one cluster should be as dissimilar as possible from documents in other clusters [24].

Cover Coefficient-based Clustering Methodology (C^{3}M) is originally proposed by Can and Ozkarahan [23] to cluster text documents. The base concept of the algorithm, the cover coefficient (CC), provides a means of estimating the number of clusters within a document database and relates indexing and clustering analytically. The CC concept is used also to identify the cluster seeds and to form clusters with these seeds. The retrieval experiments show that the information retrieval effectiveness of the algorithm is compatible with a very demanding complete linkage clustering method that is known to have good retrieval performance.

In their paper Can and Ozkarahan they showed that the complexity of C^{3}M is better than most other clustering algorithms, whose complexities range from $O(m^2)$ to $O(m^3)$. Also their experiments show that C^{3}M is time efficient and suitable for very large databases. Its low complexity is experimentally validated. C^{3}M has all the desirable properties of a good clustering algorithm. C^{3}M is a seed-based partitioning type clustering scheme. Basically, it consists of two different steps that are cluster seed selection and the cluster construction. *D* matrix is the input for C^{3}M, which represents documents and their terms. It is assumed that each document contains *n* terms and database consists of *m* documents. The need is to construct *C* matrix, in order to employ cluster seeds for C^{3}M. *C*, is a document-by-document matrix whose entries c_{ij} ($1 < i, j < m$) indicate the probability of selecting any term of d_i from d_j. In other words, the *C* matrix indicates the relationship between documents based on a two-stage probability experiment. The experiment randomly selects terms from documents in two stages. The first stage randomly chooses a term t_k of document d_i; then the second stage chooses the selected term t_k from document d_j. For the calculation of *C* matrix, c_{ij}, one must first select an arbitrary term of d_i, say, t_k, and use this term to try to select document d_j from this term, that is, to check if d_j contains t_k. Each row of the *C* matrix summarizes the results of this two-stage experiment.

Let s_{ik} indicate the event of selecting t_k from d_i at the first stage, and let s'_{jk} indicate the event of selecting d_j, from t_k at the second stage. In this experiment, the probability of the simple event "s_{ik} and s'_{jk}" that is, $P(s_{ik}, s'_{jk})$ can be represented as $P(s_{ik}) \times P(s'_{jk})$. To simplify the notation, s_{ik} and s'_{jk} can be used respectively, for $P(s_{ik})$ and $P(s'_{jk})$, where;

$$s_{ik} = \frac{d_{ik}}{\sum_{h=1}^{n}(d_{ih})}, \text{ and } s'_{jk} = \frac{d_{jk}}{\sum_{h=1}^{m}(d_{hk})}, \text{ where } 1 \le i,\ j \le m, 1 \le k \le n \quad (12)$$

By considering document d_i, *D* matrix can be represented with respect to the two-stage probability model. Each element of *C* matrix, c_{ij}, (the probability of selecting a term of d_i from d_j) can be founded by summing the probabilities of individual path from d_i to d_j.

$$c_{ij} = \sum_{i=1}^{n} (s_{ik} \times s'_{jk}) \quad (13)$$

this can be written as;

$$(c_{ij} = \alpha_i \sum_{k=1}^{n} (d_{ik} \times \beta_k \times d_{jk}), \text{ where } 1 \leq i, j \leq m) \quad (14)$$

In our work, we used C^{3}M clustering algorithm during final reranking phase. For each query, the boolean reranking resultsets are utilized as inputs for clustering process. Expanded forms of queries are also appended into these resultsets. For each resultset, C^{3}M algorithm is run and C matrix is constructed. C matrix includes similarity scores by keeping a document-by-document matrix. Since, the expanded form of query is appended as a document, query-by-document similarity scores (c_{ij}) are also generated for each query and document.

Both $R_{boolranked}$ and c_{ij} are used for final ranking score calculation. Since the calculation and results of these two values are different, these values should be approximated and compensated to each other mathematically. Formally,

$$R_{CCSimScore} = c_{ij} \times \frac{(maxBoolRankedScore \times simScoreEffect)}{(100 \times maxCMatrixScore)} \quad (15)$$

where, $maxBoolRankedScore$ is the maximum boolean ranking score for the query resultset, $simScoreEffect$ specifies the percentage of ranking score effect for experiment, $maxCMatrixScore$ is the maximum query-by-document similarity score for the query. Final C^{3}M ranking score equation is showed in (16)

$$R_{clustering} = (R_{boolranked} \times \mu 2) + (R_{CCSimScore} \times \partial 2) \quad (16)$$

Where, $R_{boolranked}$ is the Boolean retrieval score, $R_{CCSimScore}$ is the compensated query-document ranking score and $R_{clustering}$ is the final ranking score. $\mu 2$ and $\partial 2$ are constant parameters. In this work, their values are set as 0.9 and 1 respectively for experimental results.

5 Experimental Results

WikipediaMM task provides a test bed for the system-oriented evaluation of visual information retrieval from a collection of Wikipedia images. The aim is to investigate retrieval approaches in the context of a larger scale and heterogeneous collection of images (similar to those encountered on the Web) that are searched for by users with diverse information needs. It contains 151,519 images that cover diverse topics of interest. These images are associated with unstructured and noisy textual annotations in English. WikipediaMM dataset includes 45 queries in 2009 sub-track [11].

The main focus of this work is to improve search results by two phased reranking. The set of initial retrieved documents are re-ordered for better results by increasing the precision. It is obvious that two phased reranking approach improves results over the baseline and over expansion results. The baseline retrieval, expansion and reranking methods are realized on Wiki 2009 sub-track [11]. We have participated in Wiki 2009 with 6 runs and 4 of our runs ranked the best MAP values.

In all of the runs, pivoted unique normalization is used. The documents are expanded with WSD and only noun and adjective representations (POS) are used. The original forms of documents are also kept to calculate the ranking weight score as a combination of original and expanded dataset weights.

The differences between the runs are based on the different techniques of query expansion and reranking. In the first run (200); the original forms of the queries are used in the retrieval process. In the second run (201); term phrases are selected and added to the queries. And the documents are retrieved with the expanded queries. In the third run (202); the expanded queries from the second run are used. In addition to this; the one-length queries are expanded with WSD. Starting from the fourth run, in the next three runs, different re-ranking methods are applied to the retrieved result set from the third (201) run. In the fourth run (203); Boolean retrieval is applied to the retrieved result set and the resultset is re-ranked according to this. The difference between the fifth run (204) and the fourth run is that only the documents in the result set above a threshold value are taken for the Boolean retrieval process. Finally in the sixth run (205), the resultset of the fourth run is saved and C^{3}M clustering is applied to this result set.

In conclusion, expanding the query with WSD and term-phrase selection increases the quality of the retrieved result set and, reranking the retrieved result set improves precision values. The increase of precision values by reranking is represented in Figure 3 for Wiki 2009. As it can be seen from the MAP values in Table 1 for Wiki

2009, the best result obtained from the sixth run, in which the result set is re-ranked with C^3M clustering. And the second best result obtained by reranking with Boolean retrieval. Our experimentation shows that reranking generated the best results among all participants (205 and 204).

ID	MAP	P@5	P@10	R-Precision	Retrieved	Rel.Ret.	Relevant
200	0.1861	0.3244	0.2956	0.2133	41242	1283	1622
201	0.1865	0.3422	0.2978	0.2146	41242	1283	1622
202	0.2358	0.4844	0.3933	0.2708	43052	1352	1622
203	0.2375	0.4933	0.4000	0.2692	43053	1351	1622
204	**0.2375**	**0.4933**	**0.4000**	0.2692	39257	1351	1622
205	**0.2397**	**0.5156**	**0.4000**	0.2683	43052	1351	1622

Table 1: Performance of our runs in Wikipedia MM Task.

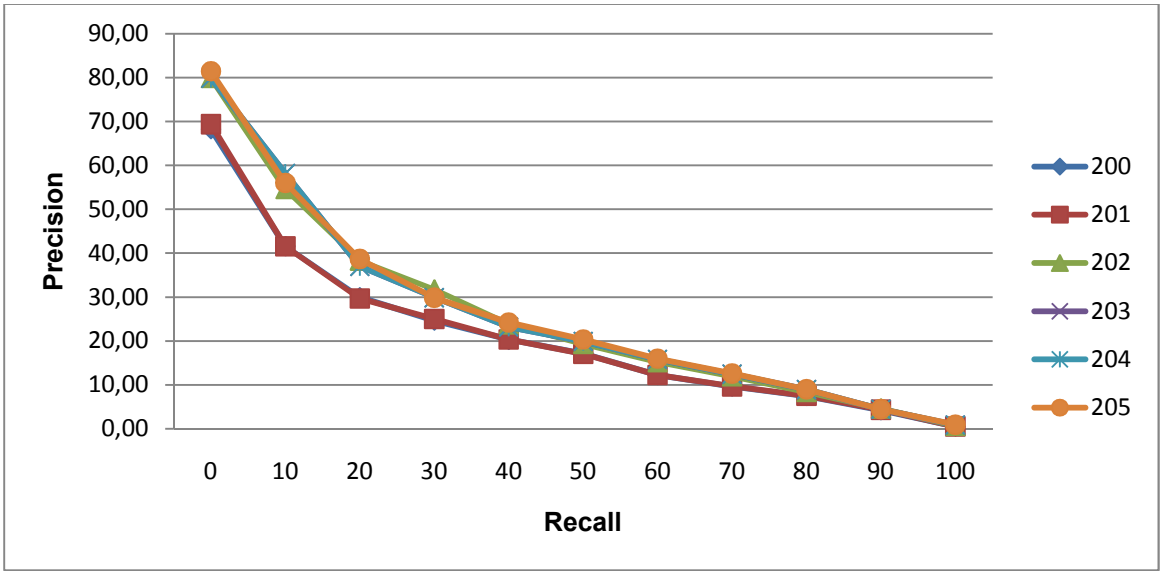

Figure 3 – Wiki 2009 P/R Graph

Acknowledgement

This work is supported by Turkish National Science Foundation (TÜBİTAK) under project number 107E217.

References

[1] Miller, G.A. et al., 1990. Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography*, Vol. 3, pp. 235-312.

[2] Z. Wu and M. Palmer. Verb Semantics and Lexical Selection. In *Annual Meeting of the Associations for Computational Linguistics (ACL'94)*, pages 133–138, Las Cruces, New Mexico, 1994.

[3] M. Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine code from an ice cream cone, in: *Proceedings of the 5th annual international conference on Systems documentation, ACM Press*, 1986, pp. 24–26.

[4] R. Richardson, A. Smeaton, and J. Murphy. Using WordNet as a Knowledge Base for Measuring Semantic Similarity Between Words. Techn. Report Working paper CA-1294, School of Computer Applications, Dublin City University, Dublin, Ireland, 1994.

[5] Y. Li, Z. A. Bandar, and D. McLean. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. *IEEE Trans. On Knowledge and Data Engineering*, 15(4):871–882, July/Aug. 2003.

[6] O. Resnik, Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity and Natural Language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[7] A. Tversky, Features of Similarity. *Psychological Review*, 84(4):327–352, 1977.
[8] Gerard Salton, A.Wong, and C.S. Yang. A vector space model for information retrieval. *Journal of the American Society for Information Science*, 18(11):613-620,November 1975.
[9] E. Garcia. Implementation and application of term weights in mysql environment, 10 2006.
[10] Chris Buckley. The importance of proper weighting methods. In M. Bates, editor. *Human Language Technology*. Morgan Kaufman, 1993.
[11] Theodora Tsikrika and Jana Kludas. Overview of the wikipediaMM task at ImageCLEF 2009, CLEF working notes 2009, Corfu, Greece, 2009
[12] Theodora Tsikrika and Jana Kludas. Overview of the wikipediaMM task at ImageCLEF 2008. In Evaluating Systems for Multilingual and Multimodal Information Access, Proceedings of the 9th Workshop of the Cross-Language Evaluation Forum, Lecture Notes in Computer Science, vol. 5709, pp. 539-550, Springer 2009.
[13] Lingpeng Yang, Donghong Ji, Guodong Zhou, Yu Nie, Guozheng Xiao, "Document re-ranking using cluster validation and label propagation" *Proceedings of the 15th ACM international conference on Information and knowledge management CIKM '06, pp. 690 – 697*.
[14] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. *In SIGIR '98 Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval,* pages 335 336, New York, NY, USA, 1998. ACM.
[15] Lingpeng Yang, Donghong Ji, Guodong Zhou and Yu Nie. Improving retrieval effectiveness by using key terms in top retrieved documents. *Advances in Information Retrieval*, pages 169 184, 2005.
[16] Jaroslaw Balanski and Czeslaw Danilowicz. Re-ranking method based on inter-document distances. *Information Processing & Management*, 41(4):759 775, 2005.
[17] James Allan, Anton Leuski, Russel Swan, and Donald Byrd. Evaluating combinations of ranked lists and visualizations of inter-document similarity. *Information Processing & Management*, 37(3):435 458, 2001.
[18] James Callan, W. Bruce Croft, and Stephen M. Harding. The inquery retrieval system. *In Proceedings of the Third International Conference on Database and Expert Systems Applications*, pages 78 83. Springer-Verlag, 1992.
[19] Guhnan Park, Yunju Beak, and Heung-Kyu Lee. Re-ranking algorithm using post retrieval clustering for content-based image retrieval. *Information Processing & Management*, 41(2):177 194, 2005.
[20] Kyung-Soon Lee, Young-Chan Park, and Key-Sun Choi. Re-ranking model based on document clusters. *Information Processing & Management*, 37(1):1 14, 2001.
[21] B. Chidlovskii, N. Glance, and A. Grasso. Collaborative reranking of search results. In *Proc. AAAI-2000 Workshop on AI for Web Search.*, 2000.
[22] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of WWW 2004*, pages 675 – 684, 2004.
[23] Can, F., Ozkarahan. E.A. [1990]. "Concepts and Effectiveness of the Cover Coefficient Based Clustering Methodology for Text Databases", *ACM Transactions on Database Systems*, Vol. 15, No. 4.
[24] Manning D. Chirstopher, Raghavan Prabhakar and Schütze Hinrich. An Introduction to Information Retrieval, Cambridge University Press, 2009.