The Cluster Hypothesis for Entity Oriented Search

Hadas Raviv, Oren Kurland
Faculty of Industrial Engineering and Management, Technion, Haifa 32000, Israel
hadasrv@tx.technion.ac.il,kurland@ie.technion.ac.il

David Carmel
Yahoo! Research, Haifa 31905, Israel
david.carmel@ymail.com

ABSTRACT

In this work we study the *cluster hypothesis* for entity oriented search (EOS). Specifically, we show that the hypothesis can hold to a substantial extent for several entity similarity measures. We also demonstrate the retrieval effectiveness merits of using clusters of similar entities for EOS.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

Keywords: cluster hypothesis, entity oriented search

1. INTRODUCTION

The entity oriented search (EOS) task has attracted much research attention lately. The main goal is to rank entities in response to a query by their presumed relevance to the information need that the query expresses. For example, the goal in the TREC's expert search task was to rank employees in the enterprise by their expertise in a topic [6]. The goal in INEX entity ranking track was to retrieve entities that pertain to a topic in the English Wikipedia [7, 8, 9]. The goal in TREC's entity track was to rank Web entities with respect to a given entity by their relationships [2, 3, 4].

The EOS task is different than the standard ad-hoc document retrieval task as entities are somewhat more complex than (flat) documents. That is, entities are characterized by different properties such as name, type (e.g., place or person), and potentially, an associated document (e.g., a homepage or a Wikipedia page). Despite the fundamental difference between the two tasks, we set as a goal to study whether an important principle in ad-hoc document retrieval also holds for the EOS task; namely, the *cluster hypothesis* [22]. We present the first study of the cluster hypothesis for EOS, where the hypothesis is that "closely associated *entities* tend to be relevant to the same requests".

We use several inter-entity similarity measures to quantify the association between entities, which is a key point in the hypothesis. These measures are based on the entity type which is a highly important source of information [18, 14]. We then show that the cluster hypothesis, tested using Voorhees' nearest neighbor test [23], can hold to a substantial extent for EOS for several of the similarity measures.

Motivated by the findings about the cluster hypothesis, we explore the merits of using clusters of similar entities for entity ranking. We show that ranking entity clusters by the percentage of relevant entities that they contain can be used to produce *extremely* effective entity ranking. We also demonstrate the effectiveness of using cluster ranking techniques that are based on estimating the percentage of relevant entities in the clusters for entity ranking.

Our main contributions are three fold: (i) showing that for several inter-entity similarity measures the cluster hypothesis holds for EOS to a substantial extent as determined by the nearest neighbor test; (ii) demonstrating the considerable potential of using clusters of similar entities for EOS; and, (iii) showing that using simple cluster ranking methods can help to improve retrieval performance with respect to that of an effective initial search.

2. RELATED WORK

Any entity in the INEX entity ranking track, which we use for our experiments, has a Wikipedia page. Hence, some previously proposed estimates for the entity-query similarity are based on Wikipedia's category information [20, 1, 18, 14]. Similarly, we measure the similarity between two entities based on several measures of the similarity between the sets of categories of the two entity pages.

In Cao et al.'s work on expert search [5] the retrieval score assigned to an entity was smoothed with the retrieval score assigned to a cluster constructed from the entity. In contrast, we explore a retrieval paradigm that ranks entity clusters and transforms the ranking to entity ranking. Work on document retrieval showed that cluster-based (score) smoothing and cluster ranking are complementary [16]. We leave the exploration of this finding for the EOS task for future work.

There are several tests of the cluster hypothesis for document retrieval [13, 10, 23, 19]. We use Voorhees' nearest-neighbor test for the EOS task as it is directly connected with the nearest neighbor clusters we use for ranking.

The findings we present for the EOS task echo those reported for document retrieval. Namely, the extent to which the cluster hypothesis holds [23], and the (potential) merits of using cluster ranking [12, 21, 15, 17].

3. THE CLUSTER HYPOTHESIS

Our first goal is to explore the extent to which the cluster hypothesis holds for EOS. To this end, we use the nearest

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

neighbor test [23]. Let $L_q^{[n]}$ be the list of n entities that are the highest ranked by an initial search performed in response to query q. For each relevant entity in $L_q^{[n]}$, we record the percentage of relevant entities among its K nearest neighbors in $L_q^{[n]}$. The nearest neighbors are determined using one of the inter-entity similarity measures specified in Section 5.1. The test result is the average of the recorded percentages over all relevant entities in $L_q^{[n]}$, averaged over all test queries.

Some of the inter-entity similarity measures assign discrete values including 0. Hence, for some relevant entities there could be less than K neighbors as we do not consider neighbors with a 0 similarity value. In addition, a relevant entity might be assigned with more than K nearest neighbors due to ties in the similarity measure. That is, we keep collecting all entities having the same similarity value as that of the last one in the K neighbors list.

4. CLUSTER-BASED ENTITY RANKING

Our second goal is studying the potential merits of using entity clusters to induce entity ranking. We re-rank the initial entity list $L_q^{[n]}$ using a cluster-based paradigm which is very common in work on document retrieval [17]. Let $Cl(L_q^{[n]})$ be the set of clusters created from $L_q^{[n]}$ using *some* clustering method. The inter-entity similarity measures used for creating clusters are those used for testing the cluster hypothesis. (See Section 5.1 for further technical details.) The clusters in $Cl(L_q^{[n]})$ are ranked by the presumed percentage of relevant entities that they contain. Below we describe two cluster ranking methods. Then, each cluster is replaced with its constituent entities while omitting repeats. Within cluster entity ranking is based on the initial entity retrieval scores which were used to create the list $L_q^{[n]}$.

The **MeanScore** cluster ranking method scores cluster c by the mean retrieval score of its constituent entities: $\frac{1}{|c|}\sum_{e\in c} S_{init}(e;q)$; $S_{init}(e;q)$ is the initial retrieval score of entity e; $|c|$ is the number of entities in c.

When $S_{init}(e;q)$ is a rank equivalent estimate to that of $\log(Pr(q,e))$ [18], the cluster score assigned by MeanScore is rank equivalent to the geometric mean of the joint query-entity probabilities' estimates in the cluster. Using a geometric-mean-based representation for document clusters was shown to be highly effective for ranking document clusters [17].

The regularized mean score method, **RegMeanScore** in short, which is novel to this study, smoothes c's score: $\frac{\sum_{e\in c} S_{init}(e;q)+\frac{1}{n}\sum_{e\in L_q^{[n]}} S_{init}(e;q)}{|c|+1}$. The cluster score is the mean retrieval score of a cluster composed of c's entities and an additional "pseudo" entity whose score is the mean score in the initial list. This method helps to address, among others, cluster-size bias issues.

5. EVALUATION

5.1 Experimental setup

We conducted experiments with the datasets of the INEX *entity ranking track* of 2007 [7], 2008 [8], and 2009 [9]. Table 1 provides a summary of the datasets. The tracks for 2007 and 2008 used the English Wikipedia dataset from 2006, while the 2009 track used the English Wikipedia from 2008. The set of test topics for 2007 is composed of 21 topics that were derived from the ad hoc 2007 assessments, and additional 25 topics that were created by the participants specifically for the track. In 2008, 35 topics were created and used for testing. The topics used for testing in 2009 were 55 topics out of the 60 test topics used in 2007 and 2008.

Data set	Collection size	# of documents	# of test topics
2007	4.4 GB	659,388	46
2008	4.4 GB	659,388	35
2009	50.7 GB	2,666,190	55

Table 1: INEX entity ranking datasets.

We used Lucene (http://lucene.apache.org/core/) for experiments. The data was pre-processed using Lucene, including tokenization, stopword removal, and Porter stemming.

Inter-entity similarity measures. The inter-entity similarity measures that we use utilize Wikipedia categories. Specifically, the categories associated with the Wikipedia page of the entity, henceforth referred to as its category set, serve as the entity type.

The **Tree** similarity between two entities e_1 and e_2 is $\exp(-\alpha d(e_1,e_2))$ where $d(e_1,e_2)$ is the minimum distance over Wikipedia's categories graph between a category in e_1's category set and a category in e_2's category set; α is a decay constant determined as in [18]. The **SharedCat** measure is the cosine similarity between the binary vectors representing two entities. An entity vector is defined over the categories space. An entry in the vector is 1 if the corresponding category is associated with the entity and 0 otherwise. Thus, SharedCat measures the (normalized) number of categories shared by the two entities [20]. The **CE** measure is based on measuring the language-model-based similarity between the documents associated with the category sets of two entities [14]. More specifically, each category is represented in this case by the text that results from concatenating all Wikipedia pages associated with the category. The similarity between the texts x and y that represent two categories is $\exp(-CE(p_x^{[0]}(\cdot)||p_y^{[\mu]}(\cdot)))$; CE is the cross entropy measure; $p_z^{[\mu]}(\cdot)$ is the Dirichlet-smoothed unigram language model induced from z with the smoothing parameter μ (=1000). The CE similarity between two entities is defined as the maximal similarity, over all pairs of categories, one in the first entity's category set and the other in the second entity's category set, of the texts representing the categories. Finally, the **ESA** (Explicit Semantic Analysis) [11] similarity measure is the cosine between two vectors, each represents the category set of an entity. The vectors representing the category sets are defined over the entities space. The value of an entry in the vector is the number of the categories in the given category set that are associated with the corresponding entity. Using ESA to measure inter-entity similarity is novel to this study.

Three different initially retrieved entity lists, $L_q^{[n]}$, are used for both the cluster hypothesis test and cluster-based ranking. The lists are created in response to the query using highly effective entity retrieval methods [18]. The first list, L_{Doc}, is created by representing an entity with its Wikipedia document (page). The documents are ranked in response to the query using the standard language-model-based approach with Dirichlet-smoothed unigram language models and the cross entropy similarity measure. The second list,

Similarity measure	Initial list	2007	2008	2009
Tree	L_{Doc}	30.0	32.0	42.7
	$L_{Doc;Type}$	29.8	35.5	44.9
	$L_{Doc;Type;Name}$	32.7	37.7	44.9
SharedCat	L_{Doc}	35.7	41.0	45.4
	$L_{Doc;Type}$	33.5	45.5	52.2
	$L_{Doc;Type;Name}$	37.9	44.3	52.7
CE	L_{Doc}	33.4	36.2	46.0
	$L_{Doc;Type}$	34.5	38.6	50.3
	$L_{Doc;Type;Name}$	37.5	41.7	49.7
ESA	L_{Doc}	34.3	36.2	46.2
	$L_{Doc;Type}$	33.7	41.0	49.5
	$L_{Doc;Type;Name}$	37.3	39.1	49.0

Table 2: The cluster hypothesis test: the average percentage of relevant entities among the 5 nearest neighbors of a relevant entity.

$L_{Doc;Type}$, is created by scoring entities with an interpolation of two scores. The first is that used to create the list L_{Doc}. The second is the similarity between the category set of the entity and the query target type (the set of categories that are relevant to the query, as defined by INEX topics). The Tree estimate described above is used for measuring similarity between the two category sets. The third list, $L_{Doc;Type;Name}$, is created by scoring an entity with an interpolation of the score used to create $L_{Doc;Type}$, and an estimate for the proximity-based association [18] between the query terms and the entity name (i.e., the title of its Wikipedia page) in the corpus. We employ the same train-test approach as in [18] to set the free-parameter values of the ranking methods used to create the initial lists. The number of entities in each initial list $L_q^{[n]}$ is $n = 50$.

We use a simple nearest neighbor clustering method to cluster entities in the initial list $L_q^{[n]}$. Specifically, each entity in $L_q^{[n]}$ and the K (= 5) entities in $L_q^{[n]}$ that are the most similar to it, according to the inter-entity similarity measures described above, form a cluster. Using such small overlapping clusters was shown to be highly effective for cluster-based document retrieval [16, 15, 17]. We note that not all clusters necessarily contain $K + 1$ documents due to the reasons specified in Section 3. For consistency, we also use $K = 5$ in the cluster hypothesis test.

Following the INEX guidelines, the evaluation metric for INEX 2007 is mean average precision (MAP) while that for INEX 2008 and 2009 is infAP. We also report the precision of the top 5 entities (p@5). Statistically significant differences of retrieval performance are determined using the two tailed paired t-test with a 95% confidence level.

5.2 Experimental results

5.2.1 The cluster hypothesis

Table 2 presents the results of the nearest neighbor cluster hypothesis test that was described in Section 3. The test is performed on the different initially retrieved entity lists using the various inter-entity similarity measures. We see that the average percentage of relevant entities among the nearest neighbors of a relevant entity ranges between 30% and 53% across the various experimental settings. We also found out that, on average, the percentage of relevant entities in a list is often lower than 25% and can be as low as 10%. Thus, due to the relatively high percentage of relevant entities among the nearest neighbors of relevant entities, we can conclude that the cluster hypothesis holds to a substantial extent, according to the nearest neighbor test, with various inter-entity similarity measures.

Table 2 also shows that for most of the data sets and similarity measures the test results for the $L_{Doc;Type}$ and $L_{Doc;Type;Name}$ lists are higher than for L_{Doc}. This finding is not surprising as $L_{Doc;Type}$ and $L_{Doc;Type;Name}$ were created using entity-query similarity measures that account for category information, while the similarity measure used to create L_{Doc} does not use this information. The highest test results are obtained for the SharedCat similarity measure which, as noted above, measures the (normalized) number of shared categories between two entities.

5.2.2 Cluster-based entity ranking

Table 3 presents the results of employing cluster-based entity re-ranking, as described in Section 4, upon the three initial entity lists. The various inter-entity similarity measures are used for creating the clusters. 'Initial' refers to the initial ranking of a list. 'Oracle' is the ranking of entities that results from employing the cluster-based re-ranking paradigm described in Section 4; the clusters are ranked by the *true* percentage of relevant entities that they contain.

The high performance numbers for Oracle, which are substantially and statistically significantly better than those for Initial, attest to the existence of clusters that contain a very high percentage of relevant entities. More generally, these numbers attest to the incredible potential of employing effective cluster ranking methods to rank entities.

RegMeanScore, which outperforms MeanScore due to the regularization discussed in Section 4, is in quite a few cases more effective than Initial; specifically, using the Tree and SharedCat inter-entity similarity measures. While the improvements for L_{Doc} are often statistically significant, this is not the case for $L_{Doc;Type}$ and $L_{Doc;Type;Name}$. Naturally, the more effective the initial ranking (Initial), the more challenging the re-ranking task. Yet, the very high Oracle numbers for $L_{Doc;Type}$ and $L_{Doc;Type;Name}$ imply that effective cluster ranking methods can yield performance that is much better than that of the initial ranking. Finally, for both $L_{Doc;Type}$ and $L_{Doc;Type;Name}$ the best performance is in most cases attained by using RegMeanScore.

6. CONCLUSIONS AND FUTURE WORK

We showed that the cluster hypothesis can hold to a substantial extent for the entity oriented search (EOS) task with several inter-entity similarity measures. We also demonstrated the potential merits of using a cluster-based retrieval paradigm for EOS that relies on ranking entity clusters. Devising improved cluster ranking techniques is a future venue we intend to explore.

7. ACKNOWLEDGMENTS

We thank the reviewers for their comments. Part of the work reported here was done while David Carmel was at IBM. This paper is based on work that has been supported in part by the Israel Science Foundation under grant no. 433/12 and by Google's faculty research award. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

8. REFERENCES

[1] K. Balog, M. Bron, and M. De Rijke. Query modeling for entity search based on terms, categories, and examples. *ACM Trans. Inf. Syst.*, 29(4), 2011.

[2] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the trec 2009 entity track. In *Proceedings of TREC*, 2009.

[3] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the trec 2010 entity track. In *Proceedings of TREC*, 2010.

[4] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the trec 2011 entity track. In *Proceedings of TREC*, 2011.

[5] Y. Cao, J. Liu, S. Bao, and H. Li. Research on expert search at enterprise track of TREC 2005. In *Proceedings of TREC*, volume 14, 2005.

[6] N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the trec 2005 enterprise track. In *Proceedings of TREC*, 2005.

[7] A. P. de Vries, A.-M. Vercoustre, J. A. Thom, N. Craswell, and M. Lalmas. Overview of the INEX 2007 entity ranking track. In *Proceedings of INEX*, pages 245–251, 2007.

[8] G. Demartini, A. P. de Vries, T. Iofciu, and J. Zhu. Overview of the INEX 2008 entity ranking track. In *Proceedings of INEX*, pages 243–252, 2008.

[9] G. Demartini, T. Iofciu, and A. P. de Vries. Overview of the INEX 2009 entity ranking track. In *Proceedings of INEX*, pages 254–264, 2009.

[10] A. El-Hamdouchi and P. Willett. Techniques for the measurement of clustering tendency in document retrieval systems. *Journal of Information Science*, 13:361–365, 1987.

[11] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of IJCAI*, pages 1606–1611, 2007.

[12] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of SIGIR*, pages 76–84, 1996.

[13] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217–240, 1971.

[14] R. Kaptein and J. Kamps. Exploiting the category structure of wikipedia for entity ranking. *Artificial Intelligence*, 0:111 – 129, 2013.

[15] O. Kurland and C. Domshlak. A rank-aggregation approach to searching for optimal query-specific clusters. In *Proceedings of SIGIR*, pages 547–554, 2008.

[16] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of SIGIR*, pages 194–201, 2004.

[17] X. Liu and W. B. Croft. Evaluating text representations for retrieval of the best group of documents. In *Proceedings of ECIR*, pages 454–462, 2008.

[18] H. Raviv, D. Carmel, and O. Kurland. A ranking framework for entity oriented search using markov random fields. In *Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search*, 2012.

[19] M. D. Smucker and J. Allan. A new measure of the cluster hypothesis. In *Proceedings of ICTIR*, pages 281–288, 2009.

[20] J. A. Thom, J. Pehcevski, and A.-M. Vercoustre. Use of wikipedia categories in entity ranking. *CoRR*, abs/0711.2917, 2007.

[21] A. Tombros, R. Villa, and C. Van Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing & management*, 38(4):559–582, 2002.

[22] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, second edition, 1979.

[23] E. M. Voorhees. The cluster hypothesis revisited. In *Proceedings of SIGIR*, pages 188–196, 1985.

$\mathbf{L_{Doc}}$						
	2007		2008		2009	
	MAP	p@5	infAP	p@5	infAP	p@5
Initial	20.2	26.1	12.6	19.4	19.1	35.3
Tree						
Oracle	31.8^{i}	51.3^{i}	22.3^{i}	49.7^{i}	25.5^{i}	68.0^{i}
MeanScore	21.6	26.1	13.3	17.1	20.2^{i}	36.7
RegMeanScore	21.6	26.0	13.4	18.9	**20.2**$^{\mathbf{i}}$	36.7
SharedCat						
Oracle	36.2^{i}	60.0^{i}	26.8^{i}	66.3^{i}	30.5^{i}	83.3^{i}
MeanScore	22.5	23.9	12.6	18.9	19.7	37.1
RegMeanScore	**23.1**^{i}	**27.8**^{i}	13.4^{i}	**20.6**^{i}	19.9	**38.5**
CE						
Oracle	32.3^{i}	53.5^{i}	23.3^{i}	53.7^{i}	28.0^{i}	74.2^{i}
MeanScore	22.6	27.4	13.5	20.6	19.1	36.7
RegMeanScore	22.0	26.5	**13.5**	20.6	19.1	36.7
ESA						
Oracle	33.7^{i}	57.0^{i}	26.0^{i}	60.6^{i}	28.8^{i}	80.0^{i}
MeanScore	20.9	22.6	12.8	14.9	19.2	35.6
RegMeanScore	21.9	23.9	12.9	14.9	19.2	35.3

$\mathbf{L_{Doc;Type}}$						
	2007		2008		2009	
	MAP	p@5	infAP	p@5	infAP	p@5
Initial	30.8	37.4	28.2	**44.0**	**23.8**	43.6
Tree						
Oracle	37.7^{i}	58.7^{i}	32.5^{i}	50.9^{i}	29.5^{i}	70.2^{i}
MeanScore	31.6	40.0	27.7	37.7	23.6	39.6
RegMeanScore	**31.6**	**40.0**	28.1	40.6	23.4	39.3
SharedCat						
Oracle	43.8^{i}	65.7^{i}	38.3^{i}	65.1^{i}	34.1^{i}	87.6^{i}
MeanScore	30.8	36.1	28.7	42.3	23.2	44.0
RegMeanScore	31.1	37.0	**28.9**	42.3	23.6	**45.1**
CE						
Oracle	39.0^{i}	60.0^{i}	34.3^{i}	58.3^{i}	31.3^{i}	75.6^{i}
MeanScore	31.3	38.3	28.5	40.6	23.7	42.2
RegMeanScore	31.0	37.4	28.7	40.0	23.7	42.2
ESA						
Oracle	42.1^{i}	64.3^{i}	37.7^{i}	66.9^{i}	32.9^{i}	83.6^{i}
MeanScore	28.6	34.3	28.4	42.3	22.8	42.5
RegMeanScore	29.1	34.8	28.9	44.0	22.7	41.5

$\mathbf{L_{Doc;Type;Name}}$						
	2007		2008		2009	
	MAP	p@5	infAP	p@5	infAP	p@5
Initial	33.3	40.4	35.4	**46.9**	24.4	**44.0**
Tree						
Oracle	39.6^{i}	57.4^{i}	42.3^{i}	62.3^{i}	30.3^{i}	72.0^{i}
MeanScore	34.1	43.5	35.7	42.3	24.7	42.9
RegMeanScore	**34.0**	**42.6**	**35.7**	43.4	24.6	42.2
SharedCat						
Oracle	47.4^{i}	70.4^{i}	47.8^{i}	74.9^{i}	34.3^{i}	87.^{i}3
MeanScore	32.9	38.7	33.5	42.9	24.6	41.5
RegMeanScore	33.1	39.1	34.8	44.0	**24.9**	42.9
CE						
Oracle	40.7^{i}	59.1^{i}	43.1^{i}	63.4^{i}	31.6^{i}	76.4^{i}
MeanScore	33.6	38.7	34.5	45.1	24.6	43.3
RegMeanScore	33.6	38.7	34.5	45.1	24.8	43.6
ESA						
Oracle	44.7^{i}	66.5^{i}	45.7^{i}	70.3^{i}	32.9^{i}	81.5^{i}
MeanScore	33.9	41.3	33.7	43.4	23.0	35.3
RegMeanScore	34.0	42.2	34.2	44.6	23.3	35.6

Table 3: Retrieval performance. The best result in a column (excluding that of Oracle) per an initial list is boldfaced. 'i' marks statistically significant differences with Initial.