Exploiting Knowledge Structure for Proximity-aware Movie Retrieval Model

Sansung Kim
KAIST
335, Gwahakro
Yuseong-gu, Daejeon
Republic of Korea
sansung@kaist.ac.kr

Keejun Han
KAIST
335, Gwahakro
Yuseong-gu, Daejeon
Republic of Korea
keejun.han@kaist.ac.kr

Mun Y. Yi *
KAIST
335, Gwahakro
Yuseong-gu, Daejeon
Republic of Korea
munyi@kaist.ac.kr

Sinhee Cho
KAIST
335, Gwahakro
Yuseong-gu, Daejeon
Republic of Korea
chosinhee@kaist.ac.kr

Seongchan Kim
KAIST
335, Gwahakro
Yuseong-gu, Daejeon
Republic of Korea
sckim@kaist.ac.kr

ABSTRACT
Current movie title retrieval models, such as IMDB, mainly focus on utilizing structured or semi-structured data. However, user queries for searching a movie title are often based on the movie plot, rather than its metadata. As a solution to this problem, our movie title retrieval model proposes a new way of elaborately utilizing associative relations between multiple key terms that exist in the movie plot, in order to improve search performance when users enter more than one keyword. More specifically, the proposed model exploits associative networks of key terms, called knowledge structures, derived from movie plots. Using the search query terms entered by Amazon Mechanical Turk users as the golden standard, experiments were conducted to compare the proposed retrieval model with the extant state-of-the-art retrieval models. The experiment results show that the proposed retrieval model consistently outperforms the baseline models. The findings have practical implications for semantic search of movie titles in particular, and of online entertainment contents in general.

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

General Terms
Algorithms

Keywords
Movie search; knowledge structure; proximity

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2661949

1. INTRODUCTION
Considering and incorporating query term proximity has been shown to be an effective probabilistic retrieval model in multiple studies [2, 11, 13, 14]. A key underlying assumption for proximity is that the more compact the query terms, the more likely that they are closely related; thereby, the more potentially relevant the documents will be to the topic represented in that particular set of user queries. For movie contents in particular, users often use scenic queries. For example, consider the following actual question that was observed on a commercial Q&A website (Naver Knowledge-iN[1]) in Korea: "Please tell me the title of the movie, in which a car is transformed into a robot. I want to watch it, but don't remember it's title" (translated). This example supports the idea that users tend to recall movies by describing the scenes or impressive moments from the movies, indicating that the query terms are related to each other, rather than being independent. In general, the term 'car' is not associated with 'robot'; however, those terms become closely linked in the context of the movie 'Transformers', in which a 'car' is transformed into a 'robot.'

To verify our assumption that the query terms entered to find a movie title are closely related, we analyzed the query sets of approximately 1,000 movies collected via Amazon Mechanical Turk.[2] We asked users to type queries for movies that they had seen once, but did not remember the titles clearly. The analysis results showed that a significant number of user queries were formulated from the movie plot, meaning that those terms have considerable associative relations.

Although our analysis demands the full utilization of the query term proximity information for movie retrieval, probabilistic proximity measures suggested in previous studies [2, 11, 13, 14] do not fully reflect the genuine relationships between the terms. For instance, the current best proximity measure is MinDist, reported in [13], which is the smallest positional distance of all pairs of unique matched query terms. Consider the following two

* Mun Y. Yi is the corresponding author
[1] http://kin.naver.com
[2] https://www.mturk.com

terms as an example: one that occurs at the end of a paragraph, and the other that occurs at the beginning of the next paragraph. The MinDist of the two terms is 1 and they are considered to have a significant relationship, but because a paragraph is a semantically separated segment, there is a high probability that the two terms are not semantically associated. More specifically, continuing from a previous example, given the actual query set Q = {giant, robot, car}, MinDist model located the target movie *Transformers* at the third position while located a non-target movie (i.e., *Monsters vs. Aliens*, designated *MvA*) at the first. This unsatisfying retrieval result is due to the minimum distance scores, which were 1 for both *Transformers* and *MvA*, thereby failing to provoke the re-ranking process.

To counteract the aforementioned limitation, in this paper we suggest a new proximity measure for exploiting knowledge structure, which was originally conceptualized in the field of educational psychology [5]. Unlike the probabilistic proximity measures, knowledge structures depict the various concepts and their associative relationships that exist in people's minds with regard to a specific domain. The knowledge structures of domain experts regarding a specific domain are known to be similar [5] and can be reliably extracted from a document [7]. By representing each movie as a knowledge structure that preserves the proximity semantics among the terms, the movies can be more reachable using descriptive sets of queries. Thus, we present a new movie title retrieval model that effectively searches for movie titles by leveraging the knowledge structures extracted from movie plots. Furthermore, the experiment results reveal that the proposed model outperforms other state-of-the-art retrieval models.

2. RELATED WORK

In this section, we review some of previous studies that are related to our movie retrieval model.

Proximity-aware retrieval model. Numerous studies have applied proximity measures to retrieval models. The early works discussed in [11] calculates a proximity score by considering the co-occurrence of a pair of queries in a document. In [2], the proximity computation process was then tuned to be faster for large text collection. In [13], a systematic approach was provided to heuristically combine proximity measures with the existing models of BM25 [12] and Language Model [9]. A probabilistic model [8] and enumerating sub-tree model [4] were proposed for multi-field documents such as XML. Furthermore, in [14], proximity factor was integrated into the unigram language model to weight the parameters of the multinomial document language models. In movie search, collaborative filtering method was used to generate personalized item authorities which were combined with item proximities for better search ranking [10].

Knowledge structure. A person is said to be knowledgeable if he or she knows the concepts present in a domain, and the relations between those concepts, all of which are captured in a knowledge structure [5]. Knowledge structures have mainly been used to understand cognitive behaviors during the learning process in the field of education [3]. Based on the co-occurrence of terms and the Pathfinder algorithm [6], a knowledge structure can be automatically created from a document. It was proven that the knowledge structure produced from a series of automated processes was similar to that produced by domain experts [7], meaning that the relations between terms were adequately represented in the generated knowledge structure.

To the best of our knowledge, knowledge structure has never been applied to information retrieval, though it can potentially be effective in developing probabilistic retrieval models. In this paper, we propose an automatic method of generating knowledge structures for movies, and exploit the use of knowledge structures on proximity-aware movie retrieval models.

3. PROPOSED MODEL

In this section, we first introduce an automated method for generating knowledge structures from movie plots, then moving on to explain how to utilize it in a movie retrieval model.

3.1 Knowledge Structure Creation

The proposed method that automatically generates a knowledge structure from a movie plot requires two specific information of the source: A set of keywords and distance scores of the keywords. These pieces of information are then processed with a number of refining steps to remove weak relations for noise deduction. As a first step, the keywords of the movie m need to be extracted to form the basis of a knowledge structure. To capture concepts, we only extracted nouns from a synopsis document D_m that contains a movie plot about movie m and added those nouns to concept list l_m. The distance between each pair of the keywords in the list l_m then can be measured by sentence co-occurrences similarity (SS). For SS, co-occurrence between two terms is defined only if those two terms appear in the same sentence. The distance score for SS between two terms w_i and w_j ($w_i, w_j \in l_m$) are defined as follows:

$$C_s(w_i, w_j) = \sum_{1}^{N_s} n(w_i \cap w_j) \quad (1)$$

$$distance_{SS}(w_i, w_j) = \frac{C_s}{max(C_s)} \quad (2)$$

where N_s is the number of sentences, $n(w_i \cap w_j)$ is the co-occurrences of two terms w_i and w_j, $max(C_s)$ indicates the maximum C_s between any terms in l_m for normalization. Similar to SS, we define paragraph co-occurrences similarity (PS) as the co-occurrence of two terms in the same paragraph.

Table 1. Example of co-occurrence matrix.

	S_1	S_2	S_1	…	S_N
w_i	3	0	1	…	1
w_j	2	1	0	…	2

On the other hand, we also can measure the distance score between two terms w_i and w_j in a different way by adapting cosine similarity of the co-occurrence matrices as shown in Table 1 as follows:

$$distance_{SCS}(w_i, w_j) = \frac{SV_i \cdot SV_j}{|SV_i| \times |SV_j|} \quad (3)$$

where SV_i and SV_j are vectors based on the frequencies of w_i and w_j occurring in each sentence. Paragraph co-occurrence cosine similarity (PCS), can be defined similar to equation (3) but only to consider co-occurrence per paragraph.

As the manual knowledge structure creation in [3] measured the distance between two terms by involving human judges, for automatic knowledge structure creation, we also convert our initial distance score into a 7-point Likert Scale (1: strongly related, 7: not related at all) as follows:

$$distance_f(w_i, w_j) = 7 - distance(w_i, w_j) \times 6 \quad (4)$$

Finally, the knowledge structure goes through the pathfinder algorithm [6] to eliminate data noise by removing redundant nodes.

3.2 Proximity-aware Retrieval Model

To combine the word associative relations into a new retrieval model, we obtained the original ranking scores using the existing retrieval model, Okapi BM25 [12] at first, and then re-ranked the result based on the proximity distance in a knowledge structure.

Once the original ranking was retrieved, proximity scores between the terms in a query were calculated in each movie plot. Because the query can have more than two words, average distance scores are calculated to integrate all associative relations. Given a query set Q and synopsis document D_m, the formula to determine the proximity score (PS) between Q and D_m is as follows:

$$PS(Q, D_m) = average\left(\frac{distance_f(q_i, q_j)}{maxDistance(D_m)}\right) \times n = \frac{2}{n-1}\frac{\sum_{q_i,q_j \in Q \cap D_m, q_i \neq q_j} distance_f(q_i, q_j)}{maxDistance(D_m)} \quad (5)$$

where t_i and t_j are terms in a query, $distance_f(q_i, q_j)$ is the distance score between the two terms q_i and q_j in the knowledge structure of document D_m, n is the number of queries in the query set Q, and $maxDistance(D_m)$ is the longest distance between any two terms in the knowledge structure. For normalization, the formula is divided by $maxDistance(D_m)$ because the average distance between two terms and the plot length have a positive correlation ($\rho = 0.5638$). Furthermore n is multiplied to differentiate the score based on the length of queries. In the case that either of two terms does not occur in a movie plot, the distance between the terms is defined as $maxDistance(D_m)$.

To reflect proximity characteristic that a distance score drops fast when the distance between two terms is small while it does not change much as the distance becomes larger [13], we used a convex curve of which the first derivative is negative, and the second one is positive as follows:

$$PS_f(Q, D_m) = \exp(-PS(Q, D_m) \times \alpha) \quad (6)$$

We used an exponential function to put the range of the proximity score in the [0, 1] range, and to introduce α as a parameter for variation. As α becomes smaller, the proximity function becomes linear. Finally, we combined this function with the existing retrieval model, BM25, as follows:

$$R(Q, D_m) = BM25(Q, D_m) \cdot PS_f(Q, D_m) \quad (7)$$

$$\text{where BM25}(Q, D_m) = \sum IDF(q_i) \cdot \frac{f(q_i, D_m) \cdot (k_1+1)}{f(q_i, D_m) + k_1 \cdot \left(1 - b + b \cdot \frac{|D_m|}{avgdl}\right)}$$

where k_1 and b are two parameters often set to the standard values of 2 and 0.75, $f(q_i, D_m)$ is the term frequency of q_i in document D_m, $|D_m|$ is the length of the document vector, and $avgdl$ is the average length of all synopsis document vectors.

4. EXPERIMENTS

In this section, we describe our evaluation methodology and the evaluations we performed.

4.1 Experimental Setup

To evaluate our re-ranking model, we have crawled top 1,000 movies (based on box office sales) from IMDB,[3] one of the most popular movie portals. Among several sources for movie profiles, we chose to exploit synopsis offered by IMDB as it contains abundant amount of movie plot information. The average number of words, sentences, and paragraphs in a synopsis is 904.613, 94.316, and 18.158 respectively, indicating that synopsis is substantial enough to create a representative knowledge structure for individual movies.

We also collected 10 queries[4] for each movie via Amazon Mechanical Turk, which has been used in information retrieval research for relevance assessment [1]. To control the quality of the input, we restricted users whose HIT approval rate was greater than or equal to 90%. The participants were asked to formulate a search query consisting of multiple keywords for a given movie and received $0.02 per query. In the end, 355 users participated, with an average time to formulate each query of 40.051 seconds, and an average number of words in each query of 3.749.

As our algorithm considers semantics of the words, we compared our algorithm with the proximity retrieval model (PRM) [13], which calculates proximity between words by measuring the minimum pair distance between the query terms. It was also reported to perform best among other state-of-the art models. Given a query $Q = (q_1, \ldots, q_m)$, the PRM score is tuned to show a consistent performance in our dataset as follows:

$$S_{prm}(Q, D_m) = BM25(Q, D_m) \cdot S_\pi(Q, D_m) \quad (8)$$

$$S_\pi(Q, D_m) = \log(\alpha + \exp(-\delta(Q, D_m))) \quad (9)$$

where α is a constant, and $\delta(Q, D_m)$ is a proximity-distance measure defined as the smallest positional distance of all pairs of uniquely matched-query terms. In the experiment, we set $\alpha = 0.3$, which is known to work best in a prior study [13]. This parameter value is also shown to perform best in our dataset.

For evaluation metrics, we use the Mean Reciprocal Rank (MRR) metric that assigns a value of performance for a target resource of 1/r, where r is the position of the relevant document 'd' in the result list. We also provide the P@N (Precision at position N) metric, which has a value of 1 iff $r \leq N$.

4.2 Experiment Results

In this section, we analyze the performance of the proximity approaches while our approach adopts the different distance measures: SS, PS, SCS, and PCS. Table 2 shows MRR and P@N values of the different proximity approaches. An asterisk indicates that the value is statistically significantly higher than the BM 25 counterpart (Wilcoxon test, $p < 0.01$). A † indicate that the value is statistically significantly higher than the PRM counterpart (Wilcoxon test, $p < 0.01$), on top of the significant difference in relation to the BM25 approach.

Table 2. Proximity-aware model performances

	BM25	PRM	SCS	PCS	PS	SS
MRR	0.6222	0.6410*	0.6550*	0.6596*	0.6742†	**0.6749†**
P@1	0.5046	0.5254*	0.5445*	0.5525*	0.5679†	**0.5704†**
P@2	0.6222	0.6407*	0.6552*	0.6596*	0.6793†	**0.6798†**
P@5	0.7646	0.7816*	0.7885*	0.7898*	0.7999†	**0.8036†**
P@10	0.8515	0.8691*	0.8691*	0.8698*	0.8797†	**0.8813†**

Our approach shows higher performance (statistically significant) than the existing state-of-the-art algorithms in all metrics, regardless of the distance metric used, indicating that consideration of semantics of words through knowledge structure

[3] http://www.imdb.com

[4] The data are available at http://courseshare.kaist.ac.kr/movie/

positively affects search performance consistently. In particular, our algorithm with SS presents the best performance. This combination outperformed BM25 and PRM by 8.47% and 5.30% on MRR, 13.02% and 8.56% on P@1, 9.27% and 6.10% on P@2, respectively. Especially, P@1 result for SS implies that users are more likely to find their target movie in the top of the search result compared to PRM, indicating that our model can be more effective in the case that users would like to find a specific movie.

To understand why the performance of our approach increases compared to the other state-of-the-art methods, we re-visited our motivating example and analyzed the results. Given the query set Q = {giant, robot, car}, Table 3 shows that the distance for each pair of Q produced different results depending on the distance metric used. Note again that PRM adopts the minimum distance (MinDist), and thus does not provoke the re-ranking process. On the other hand, we can see that three queries are closer to each other in the knowledge structure[5] for *Transformers*, rather than in the knowledge structure for *MvA*. This implies that our model can discover that semantics among the three terms are stronger for *Transformers*, relative to *MvA*.

Furthermore, we investigated the effect of varying α for overall search performance. Enlarging the constant α in Equation 6 forms convex relations between two terms in a document. Figure 1 shows the MRR and P@N values when the constant α varies from 0 to 2. We can see that the best performance is achieved when α is between 0.6 and 0.8 in PS, PCS, and SCS, while the overall performance gradually reduces as α increases. However, SS shows the most stable performance regardless of the value of α, suggesting that our algorithm, in combination with the SS distance metric, has promise as an effective, parameter-free method.

Table 3. Comparison of distance measures

	PRM		SS	
	Transformers	MvA	Transformers	MvA
d(giant, robot)	1	1	0.2097	0.4700
d(giant, car)	12	256	0.2097	0.5112
d(robot, car)	13	23	0.4194	0.4706

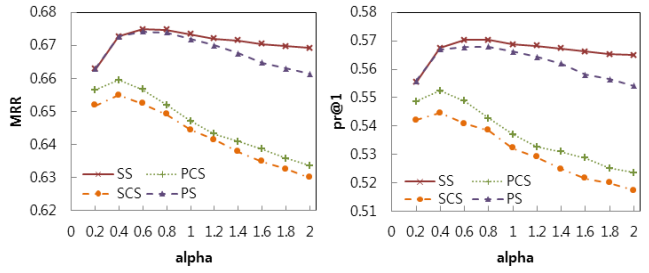

Figure 1. Performance depending on varying α

5. CONCLUSION

In this paper, we have observed that user queries are more descriptive and associative in searching movies because users tend to recall the scenes, or impressive moments of the movies, mainly in relation to the movie plot. We then presented a new movie-retrieval model that effectively searches movies by exploiting knowledge structures extracted from movie plots and measuring the proximity of terms in a query. Our algorithm outperformed other state-of-the-art proximity algorithms as it effectively utilizes the semantics of terms from the movie plots.

Our study needs further work. First, we should expand knowledge structure incorporating other information about movies, not only movie plots. Second, we expect that other multimedia content is also likely to have similar associative queries, thus we should test our algorithm on other types of multimedia content, such as music and books. Despite the need for further work, the proposed algorithm already shows promise for utilizing the potential of knowledge structure to enhance proximity-probabilistic retrieval of multimedia content.

[5] Sample knowledge structures for those two movies are shown at http://courseshare.kaist.ac.kr/movie/

6. ACKNWLEDGMENTS

This research was supported by the BK21 Plus Program of Research and Talent Management on Intelligent Knowledge Service for Innovating Human-Machine Communication and Cooperation hosted at the Department of Knowledge Service Engineering, KAIST.

7. REFERENCES

[1] R. Blanco, H. Halpin, D. M. Herzig, P. Mika, J. Pound, H. S. Thompson, and T. T. Duc. Repeatable and reliable search system evaluation using crowdsourcing. In *SIGIR '11*, pages 923-932, 2011.

[2] S. Büttcher, C. L. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *SIGIR'06*, pages 621-622, 2006.

[3] F. D. Davis, M. Y. Yi. Improving computer skill training: behavior modeling, symbolic mental rehearsal, and the role of knowledge structure. *Journal of applied psychology*, 89(3), 2004.

[4] K. Goldenberg, B. Kimelfeld, and Y. Sagiv, Keyword proximity search in complex data graphs, In *SIGMOD'08*, pages 927-940, 2008.

[5] T. E. Goldsmith, P. J. Johnson, and W. H. Acton. Assessing structural knowledge. *Journal of educational psychology*, 83(1), pages 88-96, 1991.

[6] S. Hauguel, C. Zhai, and J. Han. Parallel PathFinder algorithms for mining structures from graphs. In *ICDM'09*, pages 812-817, 2009.

[7] H. W. Kim, and M. Y. Yi. Empirical validation of an automated method of knowledge structure creation from single documents. In *IEEE ICT-KE'12*, pages 161-165, 2012.

[8] J. Y. Kim, X. Xue, W. Bruce Croft, A probabilistic model for semistructured data, In *ECIR'09*, pages 228-239, 2009.

[9] J. Lafferty, and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR'01*, pages 111-119, 2001.

[10] S. T. Park, D. M. Pennock, Applying collaborative filtering techniques to movie search for better ranking and browsing, In *KDD'07*, pages 550-559, 2007.

[11] Y. Rasolofo and J. Savoy. Term Proximity Scoring for Keyword-Based Retrieval Systems. In *ECIR'03*, pages 207–218, 2003.

[12] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *TREC-3*, 1994.

[13] T. Tao, and C. Zhai. An exploration of proximity measures in information retrieval. In *SIGIR'07*, pages 295-302, 2007.

[14] J. Zhao, and Y. Yun. A proximity language model for information retrieval. In *SIGIR'09*, pages 291-298, 2009.