รายงานวิจัยฉบับสมบูรณ์

โครงการ **การพัฒนาและการปรับปรุงวิธีการสืบค้นข้อมูลจากฐานข้อมูลการแพทย์ไทย**

โดย **ดร. ภคินี เอมมณี สถาบันเทคโนโลยีนานาชาติสิรินธร**

ธันวาคม 2553

สัญญาเลขที่ MRG5080273

รายงานวิจัยฉบับสมบูรณ์

โครงการ การพัฒนาและการปรับปรุงวิธีการสืบค้นข้อมูลจากฐานข้อมูลการแพทย์ไทย

ดร. ภคินี เอมมณี สถาบันเทคโนโลยีนานาชาติสิรินธร

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

รูปแบบ Abstract (บทคัดย่อ)

(ภาษาไทย และภาษาอังกฤษ)

Project Code : MRG5080273

(รหัสโครงการ)

Project Title : A Development and Improvement of Information Retrieval in Medical Database

(ชื่อโครงการ) การพัฒนาและการปรับปรุงวิธีการสืบค้นข้อมูลจากฐานข้อมูลการแพทย์ไทย

Investigator : Dr. Pakinee Aimmanee, Sirindhorn International Institute of Technology, Thammasat University

(ชื่อนักวิจัย) ดร. ภคินี เอมมณี สถาบันเทคโนโลยีนานาชาติสิรินธร, มหาวิทยาลัยธรรมศาสตร์

E-mail Address : pakinee@siit.tu.ac.th

Project Period : 2/7/2007 to 2/7/2010

(ระยะเวลาโครงการ)

Abstract:

The s-gram or $s_{n,k}$-gram is a generalization of n-gram term modeling obtained by allowing k-term skipping in the n-gram representation. This paper presents a framework of a multi-modal $s_{n,k}$-gram similarity combination, a combination of similarities between a document and a query encoded with several $s_{n,k}$-grams with various n and k. Adjusting weights in the similarity aggregation enables us to create a suitable approximate matching model between a relevant document and a query although such document does not include any exact terms as in the query or vice versa. In the experiments, three different types of weightings are used and compared in the combination of similarities between a document and a query each of which is encoded with a multi-modal $s_{n,k}$-gram. Two collections of medical documents that are alike in context but different in written languages (English and Thai) are the testing domain. The result shows that the proposed approach significantly outperforms the conventional approaches such as the unigram and bigram models.

(บทคัดย่อ)

s-gram หรือ $s_{n,k}$-gram เป็นการโมเดล n-gram ให้อยู่ในรูปแบบทั่วไปโดย n-gram นี้ยอมให้มีkคำไม่ปรากฏในn-gramตั้งต้น. วารสารนี้เสนอโครงงานในการรวมหลายๆ ค่าความเหมือนของเอกสารกับสิ่งที่ต้องการค้นหาจากหลายๆ$s_{n,k}$-gramโดยเปลี่ยนค่า n และ k ให้เกิดความหลากหลาย การเพิ่มน้ำหนักของค่าความเหมือนจาก s_{n,k}$-gramจากแต่ละโมเดลของ$s_{n,k}$-gramช่วยให้เกิดการหาค่าความเหมือนระหว่างเอกสารกับสิ่งที่ต้องการค้นหาได้ดียิ่งขึ้นถึงแม้ว่าเอกสารกับสิ่งที่ต้องการค้นหาจะไม่เหมือนกันตรงๆเนื่องจากมีคำอื่นๆแทรกเข้ามาเยอะในเอกสารหรือสิ่งที่ต้องการค้นหา ในการทดลอง เราใช้3วิธีที่ต่างกันในการให้น้ำหนักในการหาค่าความเหมือนของเอกสารกับสิ่งที่ต้องการค้นหาจากหลายๆ$s_{n,k}$-gram กลุ่มของเอกสารทางด้านการแพทย์จำนวน2กลุ่มซึ่งแตกต่างกันทางด้านภาษาที่ใช้ โดยกลุ่มหนึ่งเป็นเอกสารที่เป็นภาษาอังกฤษ และอีกกลุ่มหนึ่งเป็นเอกสารที่เป็นภาษาไทย ผลการทดลองแสดงให้เห็นว่าวิธีที่เราเสนอนี้ดีขึ้นอย่างชัดเจนจากวิธีดั้งเดิมที่ใช้โมเดล 1-gram และ 2-gram

Keywords : Multimodel similarity aggregation, $S_{n,k}$ gram, information retrieval

(คำหลัก)

เนื้อหางานวิจัยประกอบด้วย

1. บทคัดย่อภาษาไทย และภาษาอังกฤษ

s-gram หรือ $s_{n,k}$-gram เป็นการโมเดล n-gram ให้อยู่ในรูปแบบทั่วไปโดย n-gram นี้ยอมให้มีkคำไม่ปรากฏในn-gramตั้งต้น. วารสารนี้เสนอโครงงานในการรวมหลายๆค่าความเหมือนของเอกสารกับสิ่งที่ต้องการค้นหาจากหลายๆ$s_{n,k}$-gramโดยเปลี่ยนค่า n และ k ให้เกิดความหลากหลาย การเพิ่มน้ำหนักของค่าความเหมือนจาก s_{n,k}$-gramจากแต่ละโมเดลของ$s_{n,k}$-gramช่วยให้เกิดการหาค่าความเหมือนระหว่างเอกสารกับสิ่งที่ต้องการค้นหาได้ดียิ่งขึ้นถึงแม้ว่าเอกสารกับสิ่งที่ต้องการค้นหาจะไม่เหมือนกันตรงๆเนื่องจากมีคำอื่นๆแทรกเข้ามาเยอะในเอกสารหรือสิ่งที่ต้องการค้นหา ในการทดลอง เราใช้3วิธีที่ต่างกันในการให้น้ำหนักในการหาค่าความเหมือนของเอกสารกับสิ่งที่ต้องการค้นหาจากหลายๆ$s_{n,k}$-gram กลุ่มของเอกสารทางด้านการแพทย์จำนวน2กลุ่มซึ่งแตกต่างกันทางด้านภาษาที่ใช้ โดยกลุ่มหนึ่งเป็นเอกสารที่เป็นภาษาอังกฤษ และอีกกลุ่มหนึ่งเป็นเอกสารที่เป็นภาษาไทย ผลการทดลองแสดงให้เห็นว่าวิธีที่เราเสนอนี้ดีขึ้นอย่างชัดเจนจากวิธีดั้งเดิมที่ใช้โมเดล 1-gram และ 2-gram

The s-gram or $s_{n,k}$-gram is a generalization of n-gram term modeling obtained by allowing k-term skipping in the n-gram representation. This paper presents a framework of a multi-modal $s_{n,k}$-gram similarity combination, a combination of similarities between a document and a query encoded with several $s_{n,k}$-grams with various n and k. Adjusting weights in the similarity aggregation enables us to create a suitable approximate matching model between a relevant document and a query although such document does not include any exact terms as in the query or vice versa. In the experiments, three different types of weightings are used and compared in the combination of similarities between a document and a query each of which is encoded with a multi-modal $s_{n,k}$-gram. Two collections of medical documents that are alike in context but different in written languages (English and Thai) are the testing domain. The result shows that the proposed approach significantly outperforms the conventional approaches such as the unigram and bigram models.

2. Executive summary

ตามเอกสารแนบ

3. วัตถุประสงค์

ตามเอกสารแนบ

4. วิธีทดลอง

ตามเอกสารแนบ

5. สรุปและวิจารณ์ผลการทดลอง และข้อเสนอแนะสำหรับงานวิจัยในอนาคต

ตามเอกสารแนบ

6. ภาคผนวก

Output จากโครงการวิจัยที่ได้รับทุนจาก สกว.

1. ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ (ระบุชื่อผู้แต่ง ชื่อเรื่อง ชื่อวารสาร ปี เล่มที่ เลขที่ และหน้า) หรือผลงานตามที่คาดไว้ในสัญญาโครงการ

   P. Aimmanee, T. Theeramunkong, Improving the Retrieval Performance by Using the Distance-Based Bigrams, ECTI Transaction EEC Vol 8, 1 (2010), 106 – 112

2. การนำผลงานวิจัยไปใช้ประโยชน์

   ไม่มี

3. ผลงานตีพิมพ์ในที่ประชุมวิชาการ

   Pakinee Aimmanee and Thanaruk Theeramunkong, Multimodal $s_{n,k}$-grams: A Skipping-based Similarity Model in Information Retrieval, to be appeared in the preceedings of the 2nd Asian Conference on Intelligent Information and database systems, 24-26 March 2010, Hue, Vietnam and series Lecture Notes in Artificial Intelligence LNCS/LNAI.

   Pakinee Aimmanee, Thanaruk Theeramunkong, Improving IR performance using s-skip n-gram term modeling, in Proceedings of the 6th International Joint Conference on Computer Science and Software Engineering (JCSSE 2009), Phuket, Thailand

   Pakinee Aimmanee and Thannaruk Theeramunkong, Improving the Retrieval Performance by Using Distance-Based Bigram, in Proceedings of ECTI-CON 2009 is the sixth annual international conference organized by Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association (ECTICON 2009), Bangkok, Thailand

4. ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติที่กำลังรอผลการตอบรับ

   Pakinee Aimmanee, Thanaruk Theeramunkong, IR Enhancement using Classified Multimodal s-gram Similarity, submitted to IEEE Transactions on Information Technology in biomedicine.

Retrieval Enhancement using Multimodal s-gram Similarity Combination

Pakinee Aimmanee*, **Thanaruk Theeramunkong**

131 Moo 5 Tivanont Rd., Bangkadi, Muang, Pratumthani, 12000

*Corresponding author, e-mail: pakinee@siit.tu.ac.th

ABSTRACT: The s-gram or $s_{n,k}$-gram is a generalization of n-gram term modeling obtained by allowing k-term skipping in the n-gram representation. This paper presents a framework of a multi-modal $s_{n,k}$-gram similarity combination, a combination of similarities between a document and a query encoded with several $s_{n,k}$-grams with various n and k. Adjusting weights in the similarity aggregation enables us to create a suitable approximate matching model between a relevant document and a query although such document does not include any exact terms as in the query or vice versa. In the experiments, three different types of weightings are used and compared in the combination of similarities between a document and a query each of which is encoded with a multi-modal $s_{n,k}$-gram. Two collections of medical documents that are alike in context but different in written languages (English and Thai) are the testing domain. The result shows that the proposed approach significantly outperforms the conventional approaches such as the unigram and bigram models.

KEYWORDS: $s_{n,k}$-gram Multi-modal $s_{n,k}$-gram similarity combination, Information retrieval, Similarity vectors, approximate matching

INTRODUCTION

Due to individual writing style, synonym usage, phrasal, and compound word variation, most of conventional keyword-based information retrieval approaches faced a problem of handling phrases or a sequence of words which are semantically identical but have different expressions. One of the remedies is to apply latent semantic indexing (LSI) to map the word-based level on to the semantic-based level[1,2]. However, LSI has a major drawback on its high computational cost in both space and time in the process of dimension/rank reduction of a term-by-document matrix[3]. As an alternative, one simple and widely used approach is approximate matching. In principle, this approach assumes that two sequences of units such as characters, phonemes, words, phrases and sounds, containing similar sequence order are likely to be equivalent in semantics. A few popular approximate matching techniques are Soundex[2,4], Edit distance(ED)[5], Longest Common Subsequence (LCS)[6], n-gram matching[7,6,8], and s-gram matching[5,9,10,11,12,13,14,15,16,17]. Among these techniques, the n-gram matching is a simple and straightforward technique that uses a window size of n contiguous strings as a unit when the similarity between two sequences is investigated. The more common units two sequences share, the more likely they are semantically similar. For instance, consider two sequences of words, *dogs love Jim* and *these dogs love Jim*. They have a common bigram (2-gram) set of {*dogs-love, love-Jim*}), and as a consequence, we may conclude that they are somehow semantically related. Recently the n-gram approximate matching schemes have been experimentally proved to be effective in the area of information retrieval in several languages[18]. It was claimed to perform well in term of accuracy on languages that are rich of compound words such as Asian languages[8]. Although these previous works showed that n-gram particularly helps improving especially precision in IR field, it may suffer drastically from low recall due to its inflexible and fixed patterns. For example, given two expressions; '*a dog quickly bites a boy*' and '*a big dog just bites a tiny boy*'. Unfortunately there is no common bigrams (2-gram) among them and thus they are incorrectly interpreted to be semantically unrelated. To solve this problem, the s-gram approximate matching which is a generalization of n-gram allows some units of the original texts to be skipped. In the character level, s-grams were employed to solve sequence order variation in applications of cross-lingual spelling between two similar languages such as English and French[9], in similar sequence-pattern findings in biological and genetic IR[19] and in our previous work[20,21,22] to improve the search performance. Inspired by the flexibility of s-gram ($s_{n,k}$-gram), this paper introduces a multi-modal $s_{n,k}$-gram, an aggregation of $s_{n,k}$-gram and provides a framework for similarity calculation, where similarity between a document and a query can be measured by encoding a document as well as a query with various n and k, and then aggregating the similarities derived from these encoding combinations. To explore the

advantages of the proposed method, a set of documents from two different languages, Thai and English, in the domain of medicine are employed since they usually include adjectives and modifiers or subordinating phrases to describe symptoms of diseases and treatments. The outline of the paper is as follows. Section 2 provides the formal description of n-gram and s-gram (or more specific as $s_{n,k}$-gram). Section 3 introduces the terminalogies used for defining similarity, how a document and a query are encoded with multi-modal $s_{n,k}$-gram, and how the similairties are aggregated. Section 4 defines the experimental settings, and evaluation methods. Section 5 illustrates the experimental results and discussions. Finally, Section 6 summarizes our work in conclusion.

S-GRAM TERM MODELING

This section describes the n-gram term model and its extension, the s-gram model. Originally invented by Shannon in 1948, the n-gram term model is formulated by a sub-sequence of n tokens from a given string and applied in many applications of IR and data mining[7 5 18]. Differ from field to field, tokens may be defined as either characters, strings, words or chunks depending on focused applications. As an example of word-level tokens, unigrams (1-grams), bigrams (2-grams), and trigrams (3-grams) generated from a given input, '*I just bought a brand new car*' are {*I , just, bought, a, brand, new, car*}, {*(I,just), (just,bought), (bought, a), (a,brand), (brand,new), (new,car)*}, and {*(I, just, bought), (just, bought, a), (bought,a, brand), (brand,new, car)*}, respectively. In several works on information retrieval and other natural language processing applications, it was evidenced that a method with a larger n usually achieves higher precision but sacrifices recall due to its strict contiguity constraint[19 18]. Phrases with modifiers, such as 'round table', 'round brown table', and 'round brown wood table' can not be retrieved by the naïve n-grams. Towards this, it is possible to apply the concept of s-gram to link these slightly different terms. In the past, s-gram was implicitly applied for approximate matching in several domains. As a practical application, s-grams were used by Califano and Rigoutsos[19] in 1993 to find string homology in DNA strings, with up to 40 different random s-gram. Exploiting an analogous concept to s-gram, Pevzner and Waterman[23] presented an algorithm for finding all locations of m-tuples in the text and in the query that differ by at most k mismatches while Lehtinen et al.[24] applied the q-gram (i.e., s-gram) for indexing achieves containing text of a highly inflected language of many languages.

While the concept of s-gram was used in various applications as described above, it was never formally defined. In 2002, Pirkola et al.[9] originally gave a definition of s-gram and applied it to tackle cross- and mono-lingual morphological variants at a character level with mainly s-grams with a two-character skip. Independently in 2003, Burkhardt and Kärkkäinen[12] invented a gaped q-gram whose concept is identical to s-gram for filtering process to speed up the approximate string matching. The gaped q-gram refers to a subset of q characters of a fixed non-contiguous pattern called *shape*. For example, s-grams with shape #### of a string *ABCDEFGHI* are AB*DE, BC*EF, CD*FG, DE*GH, and EF*HI, where # is a character to be matched and * is any ignorable character, expressing a gap or a hole in the string. Besides applications to string matching, the s-gram or gaped q-gram concept was introduced to facilitate the out-of-vocabulary word translation[10]. Unfortunately, there is no report on applying $s_{n,k}$-grams in a word-level application for text retrieval. For more detail, an s-gram can be expressed in terms of $s_{n,k}$-gram where n is the number of tokens that constitute the s-gram and k is the number of tokens in the original string that are skipped in a finite interval. A formalism of $s_{n,k}$-gram can be given as follow.

Definition 1 **($s_{n,k}$ gram of a string):** *An $s_{n,k}$-gram of a string S is a sequence of n tokens generated from S with order preserving within an interval of $n + k$ tokens in S, i.e. k tokens are missing this interval.*

Note that the $s_{n,0}$-gram corresponds to the conventional n-gram.

$s_{n,k}$-GRAM SIMILARITY FORMULATION

Exploiting s-gram models, each document and query can be expressed variously using different values of n and k in $s_{n,k}$-grams. This section introduces a formulation of a profile for a document and a query using $s_{n,k}$-grams. The profile is used for calculating the similarity between a document collection and a query.

Definition 2 **$s_{n,k}$-gram profile** *An $s_{n,k}$-gram profile of a text is a set of $(t, w(t))$ pairs where t is $s_{n,k}$-gram term generated from the text and w is a weight corresponding to t and is governed by a predefined weighting function.*

Throughout this paper, the subscript $\{n, k\}$ is called *model*. When $\{n, k\}$ is used as a subscript of d and q, it means the document and query using $\{n, k\}$ model, respectively. For ease, we call the document model and query model as d-model and q-model, respectively. For instance, given a text '*I know that they know that I know*' , the $s_{2,1}$-gram profile generated from the text using term frequency weighting is {((I-that, 1), (know-they,1), (that-know, 2), (they-that, 1), (know, I, 1)}. The similarity calculation in this work follows the definition that defined in the vector space model[3].

Similarity between profiles

The similarity between two profiles X and Y whose members are shown as $X = \{(a_1, b_1), (a_2, b_2), \ldots, (a_{n_1}, b_{n_1})\}$ and $Y = \{(c_1, d_1), (c_2, d_2), \ldots, (c_{n_2}, d_{n_2})\}$ is defined as below.

$$sim(X, Y) = \frac{\sum_{j=1}^{n_2} \sum_{i=1}^{n_1} \rho((a_i, b_i), (c_j, d_j))}{\omega(X) * \omega(Y)} \tag{1}$$

$$\rho((a_i, b_i), (c_j, d_j)) = \begin{cases} b_i * d_j & a_i = c_j, \\ 0 & otherwise \end{cases} \tag{2}$$

where $\omega(X)$, and $\omega(Y)$ are the Euclidean norms of the weights of the profiles X and Y and ρ is a function that returns the similarity score between two profiles. Semantically, this formula implies the cosine similarity between vectors generated by X and Y profiles.

Similarity Aggregation

In this subsection, we define the notation of a similarity model obtained from d-model and q-model at a fixed number of n and also define how we integrate each individual similarity model in the similarity combination.

Definition 3 A similarity between a document profile $d_{n,x}$ and a query profile $q_{n,y}$
Given a fixed gram number n, the similarity between a document with d-model $\{n, x\}$ and q-model $\{n, y\}$ can be denoted as $sim(d_{n,x}, q_{n,y})$ or for short as $sim_{n,x,y}(d, q)$. For simplicity, we call the subscript of the similarity (n, x, y) as the similarity model.

Definition 4 Combination of model similarities:
Given a set of gram numbers $N = \{1, 2, \ldots, n_{|N|}\}$, a set of skipping numbers for documents $K_x = \{x_1, x_2, \ldots, x_{|K_x|}\}$ and a set of skipping numbers for queries $K_y = \{y_1, y_2, \ldots, y_{|K_y|}\}$, the combination of similarity between a document d and query q, denoted by $sim(d, q)$, can be computed as follows.

$$sim_{n,K_x,K_y}(C, q) = \alpha_{1,0,0} sim_{1,0,0}(C, q) + \sum_{n=2}^{|N|} \sum_{i=1}^{|K_x|} \sum_{j=1}^{|K_y|} \alpha_{n,x_i,y_j} sim_{n,x_i,y_j}(C, q) \tag{3}$$

Note that the first term is similarity obtained from the unigram while the second term specifies the other model similarities from s-gram . The $\alpha_{1,\{0\},\{0\}}$ and α_{n,x_i,y_j} are the corresponding weights for $sim_{1,\{0\},\{0\}}$ and sim_{n,x_i,y_j}, respectively. Our similarity formation when the skipping value of the query side is zero and that of the document side is a nonzero N coincides with the concept of *the ordered window of the query words (*$ODN(w_1 w_2 ... w_n)$*)* defined by the Callen 1992[25]**? ?** where N in its abbreviation is the number of words skipped in the original text in the document and $w_1 w_2 \ldots w_n$ is a sequence of words in the query. Our similarity formation generalizes the concept of the ordered window to allow the skipping to be in both the document side and the query side to increase the possibility to find more matches between a document and a query than that of the ODN concept. Furthurmore, integratting the results from all model similarities with weights allows us to improve the searching performance efficiently.

EXPERIMENTAL SETUP

This section describes details of how we setup our experiments. The settings concerned are characteristics of document and query collection, models used for the document and query representation, weighting functions used for the document and query profiles and similarity functions used for computing similarity between document and query profiles. Finally the experimental models and evaluation methods are discussed.

Document Collections and Queries

In our experiment, two document collections are applied with different language environments; one for English and the other for Thai. The first collection is selected from MEDLINE and contains 1,033 English health and medical abstracts relating to disease, anatomy, and pharmaceutical. For each collection, 30 queries are created for testing the query relevance. The average lengths of the queries is 26 words per queries. As a groundtruth, a set of related documents is provided for each query. The average number of relevant documents per query is 22. The second collection contains 1,000 documents sampled from a Thai Medical corpus, originally containing 10,567 documents collected from several major Thai medical web sites. Later called MD1000, this collection of documents comprises general information about diseases, causes, incurrent disease, symptoms, cautions, prevention, treatments and herb information. The average length of the queries for this collection is 11 words, respectively. Their related documents are collected manually by judging the relevance between the queries and the documents in the collection. On average, there are 6 relevant documents per query queries. These related documents are used as corresponding answers for evaluating the accuracy.

D-Models and Q-Models

As our preliminary test shows that the performance of the model similarity when the number of grams n and the number of skips k go beyond three is usually poor, thus we limit the maximum gram number (n) and the maximum skipping number (k) to three in our experiment. According to the similarity combination equation defined in Definition 4, $n_{|N|}$ is 3 and the skipping number sets K_x and K_y are all set to $\{1, 2, 3\}$.

Term weighting and model weighting

Besides the document and query encoding models, we also explore term weighting and model weighting. In term weighting, each term in a document and a query is given a weight, indicating how much it contributes to represent the document or the query. In contrast, the model weighting concerns a weight given to the each model's similarity value between a document and a query when $s_{n,k}$-grams are combined. For the term weighting, two common weightings namely *the term frequency(TF)* and *term frequency-times-inverse document frequency (TFIDF)* are used as the term weighting $w(t)$ in each $s_{n,k}$ gram profile. For the model weighting, as the weighting factors α's in eq 3, we employ two different weighting schemes. Namely equal weight combination (EWC), the first scheme naïvely assigns the weight one to all models. As the second scheme named performance-based weight combination (PWC), performance of a model is preliminary tested and its performance is used as a weight of the model. Intuitively, a model with higher performance is given a higher weight. In this work, two kinds of performance used for model weighting are mean average precision (for short, P-PWC) and performance voting (for short, V-PWC).

Given a model, the mean average precision is the mean value of the average precisions each of which is obtained when a test query is used for testing the model. On the other hand, the performance voting of a model is the number of queries that the model yields the highest mean average precision, compared among all possible models, over the total number of queries. For example, if there are ten queries and the similarity model {2,1,0} yields the maximum average precision for two out of these ten queries. The performance voting of this model is 0.2.

Evaluation Methods

As evaluation criteria, we use two standard measures, namely the mean average precision and the maximum recall. The definition of the mean average precision $\bar{P}_V(Q)$ and the mean maximum recall $\bar{R}_M(Q)$ are given below. Here, Q is a set of queries in consideration.

Definition 5 (The rank of a document): *Given a query q and a document collection D, the rank of $d \in D$, denoted as $rank_q(d)$, is the position of d in sorted decreasing order of the similarity to q.*

Definition 6 (The recall of a ranked document): *Given a query q and a document collection D, the recall of $d \in D$ denoted as $R_q(d)$ is a ratio of the number of retrieved relevant documents whose ranks are higher or equal to $rank_q(d)$ over the total number of documents relevent to q in D.*

Note that a document cannot be retrieved if its similarity to q equals to zero. Therefore, the recall of this document is set to zero.

Definition 7 (The precision of a ranked document): *Given a query q and a document collection D, the precisionl of $d \in D$ denoted as $P_q(d)$ is a ratio of the number of retrieved relevant documents whose ranks are higher or equal to $rank_q(d)$ over the $rank_q(d)$ itself.*

Definition 8 (A maximum recall): *Given a query q and a document collection relevent to the query q, $Rel_q = \{r_1, r_2, \ldots, r_p\}$, the maximum recall can be defined as $R_M(q) = max\{R_q(r_1),\ R_q(r_2),\ \ldots,\ R_q(r_p)\}$.*

Note that if no relevant document can be retrieved for a query q, the maximum recall $R_M(q)$ is zero.

Definition 9 (A precision of a ranked document): *Given a query q and a document collection relevant to the query q, $Rel_q = \{r_1, r_2, \ldots, r_p\}$, the precision of $r_i \in Rel$ denoted as $P_q(r_i)$ is the ratio of the number of retrieved relevant documents whose ranks are higher than or equal to that of r_i over the total number of retrieved documents. If no relevant document is not retrieved, its precision is zero.*

Definition 10 (An average precision): *Given a query q and a set of relevant documents to the query q, $Rel_q = \{r_1, r_2, \ldots, r_p\}$, the average precision, denoted by P_V, is $\frac{1}{n}\Sigma_{i=1}^{n} P_q(r_i)$. Remark that when there are few relevant documents retrieved, P_V can be low even though the precisions for those relevant documents retrieved are high.*

Definition 11 (A mean of the maximum recalls): *Given a set of queries $Q = \{q_1,\ q_2,\ \ldots,\ q_{|Q|}\}$, the mean of the maximum recall over all queries in the query set Q denoted as $\bar{R}_M(Q)$ can be expressed as*

$$\bar{R}_M(Q) = \frac{1}{|Q|}\sum_{i=1}^{|Q|} R_M(q_i),$$

Definition 12 (A mean of the average precisions): *Given query collection $Q = \{q_1,\ q_2,\ \ldots,\ q_{|Q|}\}$, the mean of average precision over all queries in the query set Q denoted as $\bar{P}_V$ can be expressed as*

$$\bar{P}_V(Q) = \frac{1}{|Q|}\sum_{i=1}^{|Q|} P_V(q_i).$$

EXPERIMENTAL RESULTS

To evaluate our approach, we have performed three sets of experiments. In the first experiment, we examine the performance of a single $s_{n,k}$-gram model in order to find out which model is suitable for the task. In the second experiments, we investigate the performance of the multi-modal $s_{n,k}$-gram similarity combination.

Results of Single Models

Table 1 shows $\bar{R}_M$'s and $\bar{P}_V$'s of each single model when MEDLINE and MD1000 corpora are used, respectively. With regards to the results in the tables, the following observations can be made. The non-skipping models, i.e., the unigram model (1, 0, 0) and the bigram model (2, 0, 0) obtain considerably high recalls and precisions for both MEDLINE (English Collection) and MD1000 (Thai collection) compared to the skipping models. This result shows that terms obtained from the non-skipping models are more effective than the skipping models in terms of representing a document. Another observation on these non-skipping models is that for the English collection, the unigram model obtains higher recalls and precisions than the bigram model. However, both unigram and bigram models gain comparative performance for the Thai collection. In other words, the bigram model can represent the documents in the MD1000 collection well while it may not be useful for expressing the documents in the MEDLINE collection.

For the performance of skipping models, we discover that the higher the skipping number k is, the lower recalls and precisions we obtain. Naturally two terms that locate far away from each other, may have less semantic connection. In addition, the skipping models gain higher performance in the MD1000 collection than in the MEDLINE collection. This result signifies that there are more synonym expressions, that are in the form of adding or removing terms from an original expressions in the Thai collection than in the English collection.

Table 1 Mean of maximum recalls and mean of average precisions for MEDLINE and MD1000 of each s_{n,x_i,y_j}-gram models when TF and TFIDF are used.

	MEDLINE				MD1000			
	TF		TFIDF		TF		TFIDF	
$\{n, x_i, y_j\}$	$\bar{R}_M$	$\bar{P}_V$	$\bar{R}_M$	$\bar{P}_V$	$\bar{R}_M$	$\bar{P}_V$	$\bar{R}_M$	$\bar{P}_V$
(1,0,0)	0.90	0.47	0.90	0.51	1.00	0.42	1.00	0.55
(2,0,0)	0.35	0.25	0.35	0.25	0.85	0.41	0.85	0.49
(2,0,1)	0.05	0.05	0.05	0.05	0.42	0.17	0.42	0.19
(2,0,2)	0.04	0.03	0.04	0.03	0.28	0.08	0.28	0.09
(2,0,3)	0.02	0.01	0.02	0.01	0.26	0.09	0.26	0.11
(2,1,0)	0.07	0.05	0.07	0.05	0.18	0.03	0.18	0.04
(2,1,1)	0.10	0.08	0.10	0.08	0.55	0.25	0.56	0.28
(2,1,2)	0.05	0.04	0.05	0.04	0.39	0.11	0.39	0.14
(2,1,3)	0.01	0.01	0.01	0.01	0.28	0.06	0.28	0.08
(2,2,0)	0.04	0.03	0.04	0.03	0.21	0.04	0.22	0.05
(2,2,1)	0.04	0.03	0.05	0.04	0.22	0.04	0.22	0.05
(2,2,2)	0.05	0.04	0.05	0.04	0.35	0.13	0.35	0.13
(2,2,3)	0.02	0.02	0.02	0.02	0.24	0.09	0.24	0.10
(2,3,0)	0.04	0.03	0.04	0.03	0.09	0.05	0.08	0.05
(2,3,1)	0.03	0.03	0.03	0.03	0.14	0.03	0.14	0.03
(2,3,2)	0.03	0.02	0.03	0.02	0.20	0.05	0.20	0.05
(2,3,3)	0.02	0.02	0.02	0.02	0.18	0.07	0.18	0.07
(3,0,0)	0.08	0.07	0.08	0.07	0.54	0.26	0.54	0.30
(3,0,1)	0.02	0.02	0.02	0.02	0.26	0.13	0.26	0.14
(3,0,2)	0.01	0.01	0.01	0.01	0.21	0.06	0.21	0.07
(3,0,3)	0.01	0.01	0.01	0.01	0.11	0.06	0.11	0.06
(3,1,0)	0.02	0.02	0.02	0.02	0.09	0.02	0.09	0.03
(3,1,1)	0.01	0.01	0.01	0.01	0.29	0.16	0.29	0.17
(3,1,2)	0.01	0.01	0.01	0.01	0.17	0.13	0.17	0.13
(3,1,3)	0.01	0.01	0.01	0.01	0.14	0.07	0.14	0.07
(3,2,0)	0.01	0.01	0.01	0.01	0.09	0.05	0.09	0.05
(3,2,1)	0.01	0.01	0.01	0.01	0.12	0.05	0.12	0.06
(3,2,2)	0.01	0.01	0.01	0.01	0.13	0.07	0.13	0.08
(3,2,3)	0.01	0.01	0.01	0.01	0.10	0.08	0.10	0.08
(3,3,0)	0.02	0.01	0.02	0.01	0.05	0.02	0.05	0.03
(3,3,1)	0.00	0.00	0.00	0.00	0.06	0.04	0.06	0.04
(3,3,2)	0.01	0.01	0.01	0.01	0.09	0.05	0.09	0.05
(3,3,3)	0.01	0.01	0.01	0.01	0.07	0.06	0.06	0.05

Results of the multi-modal $s_{n,k}$-gram similarity combination

In this section, results of multi-modal $s_{n,k}$-gram similarity combination are presented. To evaluate the performance, two schemes, called average precision evaluation and Top-K rank evaluation, are considered. The first scheme evaluates a model by focusing on its precisions obtained at all distinct recalls while the latter evaluates the model by looking at precision and recall of its K top ranks. Along with the results, the observations and explanations are also provided.

Table 2 shows respectively the mean average maximum recalls and mean average precisions of MEDLINE and MD1000 when three different weightings are used in the combination for the multi-modal $s_{n,k}$-gram similarity combination. From the results of the profile-based model shown in Table 2, the following observations can be made. Among our proposed multi-modal $s_{n,k}$-gram similarity combination, the mean of average precisions of PWCs are higher than that of EWC. The results are more obvious in MD1000 than MEDLINE. With PWCs, we can get many more relevant documents that are not retrieved by EWCs. This result indicates that the proposed multi-modal $s_{n,k}$ similarity combination can enhance the performance of unigram

Table 2 Mean of maximum recalls and that of average precisions when TF and TFIDF are used as term weights of MEDLINE and MD1000 of the multi-modal $s_{n,k}$-gram similarity combination. The EWC, $P-PWC$ and $V-PWC$ combinations are compared to unigram and bigram.

	MEDLINE				MD1000			
	TF		TFIDF		TF		TFIDF	
$\{n, x_i, y_j\}$	$\bar{R}_M$	$\bar{P}_V$	$\bar{R}_M$	$\bar{P}_V$	$\bar{R}_M$	$\bar{P}_V$	$\bar{R}_M$	$\bar{P}_V$
EWC	0.91	0.48	0.91	0.53	1.00	0.44	1.00	0.51
P-PWC	0.92	0.49	0.92	0.53	1.00	0.47	1.00	0.56
V-PWC	0.90	0.47	0.91	0.51	0.89	0.44	1.00	0.55
$\{1, 0, 0\}$	0.90	0.47	0.90	0.51	1.00	0.42	1.00	0.55
$\{2, 0, 0\}$	0.35	0.25	0.35	0.26	0.85	0.40	0.85	0.49

model and bigram model, especially precision in the Thai collection. For MEDLINE, the maximum relative improvement on precision of our proposed method compared to the unigram model is up to 4.2% with TF weighting while for MD1000, it is up to 11.9 % with also TF weighting .

Consider the performance at the K top ranks, the results of MEDLINE with TF term weighting, MEDLINE with TFIDF term weighting, MD1000 with TF term weighting, and MD1000 with TFIDF term weighting are presented in Fig. 1, 2, 3, and 4, respectively. The observations and explanations are summarized below. The model with TFIDF performs slightly better than that with TF weighting in both recall and precision measures. The recalls and precisions of the MD1000 are greater than those of the MEDLINE in regard with the same type of queries and term weighting.

For the evaluation when only the first top K are considered, we discovered that regardless of types of the term weightings used in MEDLINE, the PWC model yields almost the same with or very slightly better recalls and precisions than the unigram model. In comparision, for MD1000 the improvements of PWC models over the unigram model is clearly noticable especially when TF weighting is used An improvement of PWCs over the unigram model occurs when the term weighting is TF more than when it is TFIDF especially at the early recalls (top $K < 10$). For MEDLINE, the maximum relative improvement of our proposed model with P-PWC over the unigram model is up to 10.28% on recall and 12.50% on precision whereas for MD1000, 30.65% on recall and 25.02% on precision with TF weighting.

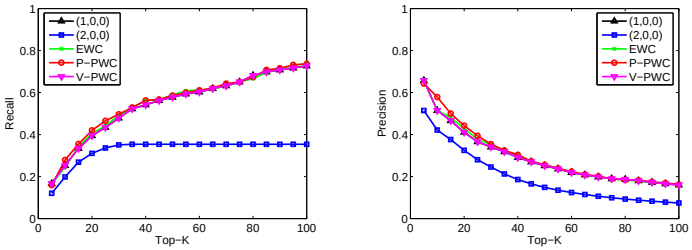

Fig. 1 Recalls (left) and precisions (right) of MEDLINE when multimodal s-gram combination with a local weighting TF

Discussion

This section provides some discussions related to the findings gained from the experiments for multi-modal similarity combination models. In contrast with the observations stated in the literature, the mean average precision of our bigram model ($(2, 0, 0)$) is no better than the unigram model ($(1, 0, 0)$). This outcome can be explained as follows. The MEDLINE collection is constructed intentionally to illustrate that Latent Semantic Indexing (LSI) is helpful for solving synonymy and polysemy problems, therefore most phrases in queries do not appear the same as in their corresponding relevant documents. As some examples, consider one of the queries from MEDLINE *the crystalline lens in vertebrates, including humans.* The relevant documents for this

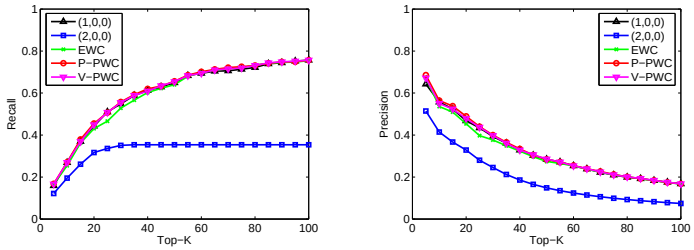

Fig. 2 Recalls and Precisions of MEDLINE with a local weighting TFIDF queries

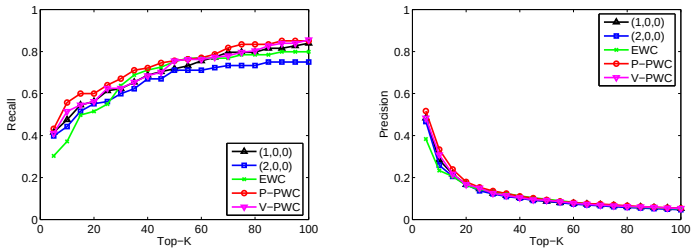

Fig. 3 Recalls(left) and Precisions(right) of MD1000 with a local weighting TF

query are shown below.

Query: *The crystalline lens in vertebrates, including humans.*

Some of its relevant documents are listed below.

1. Analysis of mammalian lens proteins by electrophoresis. Lens proteins of different mammalian species were analyzed by two-dimensional starch gel electrophoresis. The number of fractions detected by this means varied from 11-20. A-crystallin was resolved into two to three components, b-crystallin into 5-11, and y-crystallin into three to five components. This technique provides a sensitive method for the fractionation of lens proteins and for analyzing species differences.

2. Histological research on the lens in condition of hypoxia, changes in the mitotic activity of the epithelium. The effect of hypoxia on the mitotic activity of the cells of the lens epithelium was studied in 24 rats of the same strain and weight. The hypoxia was obtained in the decompression chamber. The results show that the mitotic activity of the lens epithelium is depressed at any of the examined altitudes (6.500, 8.000, 9.500 m). In particular, a marked reduction in the number of the prophases and an accumulation in metaphase was observed. The results were examined from the statistical standpoint and discussed

From this example, there are few common bigram terms in the query and each of its relevant documents. This makes the bigram model not retrieve as many documents as it should be, resulting in a low recall and a low precision.

For the faucet of MD1000, the queries are constructed manually by modifying selected texts, phrases, or sentences obtained from the document collection. As a consequence, the results from the bigram model is also not as good as those of the unigram model. In addition, the performances of the bigram model are noticeably low in regard with other multi-modal $s-gram$ similarity combination. This result elaborates that a multi-modal s-gram similarity combination tries to find the best parts of the solutions from unigram models, bigram models, and all other models. Thus it is no surprise that the multi-modal s-gram similarity combinations achieve better performance than all other models. This phenomenon is obvious when the term weighting TF is used.

When TFIDF is used for term weighting instead of TF, the inverse document frequency (IDF) adjusts the weights of the terms that are globally common to have less weight resulting the promotion of important terms and demotion of frequently occurring terms. That means the weights of terms that appear often in a document but not often in the collection are usually promoted whereas those that appear infrequently in a document but appear often in the whole collection are demoted. This characteristic makes recalls and precisions obtained

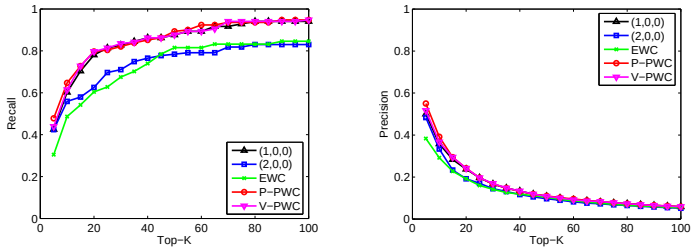

Fig. 4 Recalls(left) and precision(right) of MD1000 with a local weighting TFIDF

from TFIDF better than those of TF for all models. As our document collections contain many common terms like organ or body-part names such as skin, blood, cells, vein and many common symptoms such as cold, rash, headache, and stomach ache, these words appear frequently in describing diseases; they have larger weights due to high occurrence frequency (TF). When we search for one disease, its description may include some common terms related to symptoms and body parts, which may also appear in a description of another disease, causing several irrelevant documents retrieved at high ranks. On the other hand, when TFIDF is used, the weights of these common terms are adjusted with a factor of the inverse document frequency (IDF). Consequently the common terms will get lower weights and then these irrelevant documents get retrieved at lower ranks. With this background, we can observe that the results of all models with TFIDF weighting are usually better than those with TF weighting.

The absolute improvement on recalls and precisions of our PWCs for MD1000 over the two baseline models ($\{1,0,0\}$ and $\{2,0,0\}$) is apparently huge compared to the recall and precision improvement for MEDLINE because, as observed in Fig. 5, the recall and precision of individual skipping models such as $\{2,0,1\}$, $\{2,1,0\}$, $\{2,1,1\}$, $\{2,0,2\}$ of MD1000 are considerably higher than those of MEDLINE especially at a very low recall. This implies that there are many expressions in MD1000 that use modifier phrases and then the skipping models can help us ignore these excessive modifiers with skips to improve retrieval performance.

As for the error analysis of EWC and PWCs (the multi-modal $s_{n,k}$-gram similarity combination), the documents that these models incorrectly retrieve usually contain a set of tokens that appear exactly the same or closely as those of tokens of the query but they are semantically unrelated. For example, if the common phrase *is caused by the bacteria* is included in a query, the unigrams, bigrams or skipping grams may not include enough information to retrieve a set of relevant documents as there are many overlapping. Our multi-modal $s_{n,k}$-gram similarity combination also faces with this issue but it becomes even more serious than the case of unigram or bigram since our s-gram model puts more weights, due to multiple models, for documents that include this phrase, even they may not relate to the query.

CONCLUSION

This paper proposed a multi-modal s-gram similarity combination constructed from the $s_{n,k}$-grams, a generalization of n-gram to allow k-gram to be skipped in the n-gram where $k \in K$, to enhance the performance of information retrieval. By varying weights for each similarity model, we form two types of model combinations in the multi-modal $s_{n,k}$-gram, the equal weight combination (EWC), and two of the performance-based weight combinations (PWC). The EWC gives a weight of one to each model's similarity vector while the PWC gives a weight to each model according to each model's performance. Two medical collections, one is written in English and another is in Thai, which are similar in size and contexts are tested. By experiments, the result shows that the multi-modal $s_{n,k}$-gram similarity combination significantly outperforms the conventional approaches, unigram and bigram especially on the Thai collection. When only the first few top ranks of the similarity result are considered, the relative improvement on recall of our combination models compared to a conventional unigram model are up to 10.28% on the English collection and 30.65% on the Thai collection, respectively. For the side of precision, it is up to 12.50% and 25.02% on the English collection and the Thai collection, respectively. Our proposed multi-modal $s_{n,k}$-gram similarity combination models finds more new relevant documents than the unigram and bigram model because they consider all possibilities of

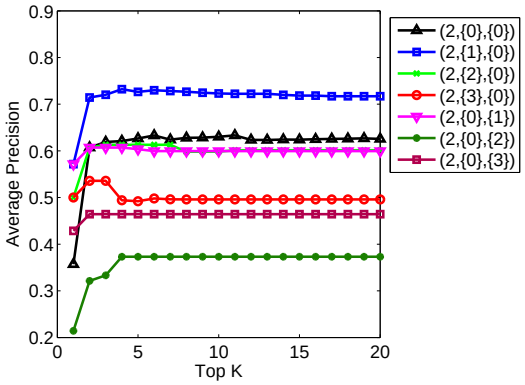

Fig. 5 The top K of skipping models

term combination including all non-contiguous ones while the order is still preserved; consequently our models increase a chance for terms in a query to match up with terms in the relevant documents even though a relevant document omits some terms from and/or adds some terms to the expressions in the query. Our work is useful for a search engine that are dealing with collections that have a wide variety of writing styles and synonym expressions.

Acknowledgements: This project (MRG5080273) is supported by the Thailand Reserarch Fund(TRF) and the Commission of Higher Education (CHE). We would like to pay gratitute to TRF and CHE for their support.

REFERENCES

1. Scott D, Susan DT, Landauer TK, Furnas GW, Laura B (1988) Improving information retrieval with latent semantic indexing. In: Proceedings of the 51^{st} Annual Meeting of the American Society for Information Science, vol 25, pp 36–40.
2. Berry MW, Dumais ST, OBrien GW (1995) Using linear algebra for intelligent information retrieval. *SIAM Review* **37**, 573–595.
3. Baeza Yates R, Ribeiro Neto B (1999) *Modern Information Retrieval*, Addison WesLey.
4. Zobel J, Dart P (1996) Phonetic string matching: lesson from information retrieval. In: Proceedings of the SIGIR conference on research and development in information retrieval, pp 166–172.
5. Jarvelin A, Jarvelin A, Jarvelin K (2007) s-grams: Defining generalized n-grams for information retrieval. *Information Process Management* **43**, 1005–1019.
6. Patrick HA, Dowling GR (1980) Approximate string matching. *ACM Computting Surveys* **12**, 381–402.
7. Robertson AM, Willett P (1998) Applications of n-grams in textual information systems. *Journal of Documentation* **54**, 48–69.
8. Lee JH, Ahn JS (1996) Using n-grams for korean text retrieval. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, pp 216 – 224.
9. Pirkola A, Keskustalo H, Leppanen E, Kansala AP, Jarvelin K (2002) Targeted s-gram matching: a novel n-gram matching technique for cross- and mono-lingual form variants. *Information Process Management* **4**, 231–255.
10. Jarvelin A, Jarvelin A (2008) Comparison of s-gram proximity measures in out-of-vocabulary word translation. *Lecture Notes in Computer Science* **5280**, 75–86.
11. Ukkonen E (1992) Approximate string matching with q-grams and maximal matches. *Theoretical Computer Science* **92**, 191–192.
12. Burkhardt S, Karkkainen J (2003) Better filtering with a gapped q-grams. *Fundamenta Informaticae* **56**, 51–70.
13. Heikki K, Ari P, Kari V, Erkka L, Kalervo J (2003) Non-adjacent digrams improve mathcing of cross lingual spelling variants. In: Proceedings of the 10th international Symposium on String Processing and Information Retrieval (SPIRE). Lecture Notes in Computer Science 2857, pp 252–265.
14. Pfieiffer Ulrich FN Poersch Thomas (1996) Retrieval effectiveness of proper name search methods. *Information Processing and Management* **32**, 667–679.

15. Julian UR (1977) A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words. *The Computer Journal* **20**, 141–147.
16. David GA, Ophir F (2004) *Information Retrieval: Algorithms and Heuristics*, Springer.
17. Mahleko B WAFP (2005) Process annotated service discovery facilitates by an n-gram based index. In: In IEEE International COnference on e-Technology, e-commerce and e-service, pp 2–8.
18. William CB (1995) Using an n-gram based document representation with a vector processing retrieval model. *NIST special publication* **225**, 269–277.
19. Califano A, Rigoutsos I (1993) Flash: A fast look up algorithm for string homology. In: Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, pp 56–64.
20. Aimmanee P, Theeramunkong T (2009) Improving the retrieval performance by using distance-based bigram. In: in proceedings of the 6th the sixth annual international conference organized by Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association.
21. Aimmanee P, Theeramunkong T (2009) Improving the performance of ir using s-skip n-gram term modeling. In: in proceedings of the 6th International Joint Conference on Computer Science and Software Engineering (JCSSE2009).
22. Aimmanee P, Theeramunkong T (2010) Multimodal $s_{n,k}$ grams: A skipping based similarity model in information retrieval. In: To be appeared in proceedings of the 2nd Asian Conference on Intelligent Information and Database Systems and Lecture note in Artificial Intelligence LNCS.
23. Pevzner PA, Waterman MS (1995) Multiple filtration and approximate pattern matching. *Algorithmica* **13**, 135–154.
24. Lehtinen O, Sutinen E, Tarhio J (1996) Experiments on block indexing. In: Proceedings of the 3rd South American Workshop on String Processing(WSP96), pp 183–193.
25. James P Callan JB Bruce W Croft (1995) Trec and tipster experiments with inquery. *Information Processing and Management* **31**, 327–343.