Extended Vector Space Model with Semantic Relatedness on Java Archive Search Engine

Oscar Karnalim
Bachelor of Information Technology, Maranatha Christian University
Bandung, Indonesia
oscar.karnalim@itmaranatha.org

Abstract— **Byte code as information source is a novel approach which enable Java archive search engine to be built without relying on another resources except the Java archive itself [1]. Unfortunately, its effectiveness is not considerably high since some relevant documents may not be retrieved because of vocabulary mismatch. In this research, a vector space model (VSM) is extended with semantic relatedness to overcome vocabulary mismatch issue in Java archive search engine. Aiming the most effective retrieval model, some sort of equations in retrieval models are also proposed and evaluated such as sum up all related term, substituting non-existing term with most related term, logaritmic normalization, context-specific relatedness, and low-rank query-related retrieved documents. In general, semantic relatedness improves recall as a tradeoff of its precision reduction. A scheme to take the advantage of relatedness without affected by its disadvantage (VSM + considering non-retrieved documents as low-rank retrieved documents using semantic relatedness) is also proposed in this research. This scheme assures that relatedness score should be ranked lower than standard exact-match score. This scheme yields 1.754% higher effectiveness than standard VSM used in previous research.**

Keywords—extended vector space model; semantic relatedness; java archive; search engine

I. INTRODUCTION

Search engine effectiveness is measured by two main factor which are precision and recall. Precision is determined by how many retrieved documents are relevant whereas recall is determined by how many relevant documents are retrieved. Byte code as information source is a novel approach which enable Java archive search engine to be built without relying on another resources except the Java archive itself [1]. Unfortunately, its effectiveness is not considerably high (69.282 % mean average precision and 94.369 % recall) since its retrieval model still uses lexical exact matching VSM. One of the major problems in lexical exact matching is that a relevant document is considered irrelevant when there is no query term on it [2]. Some approaches have been developed to overcome this problem such as modifying its text acquisition and transformation. Although the variety of vocabulary word is reducible by these mechanism, synonymy and relatedness between two lexical terms are still not handled. Many well-developed semantic relatedness measurement algorithms have been developed in some researches (e.g. Hirst & St-Onge [3], Leacock & Chodorow [4], Banerjee & Pedersen [5], Wu & Palmer [6], Resnik [7], Jiang & Conrath [8], Lin [9], and Patwardhan & Pedersen [10]). Most of these algorithms rely on an ontology to measure term pair relatedness.

In this research, Retrieval model of Java archive search engine is extended using semantic relatedness to improve its effectiveness. Aiming the most effective retrieval model, some sort of equations in retrieval models are also proposed and evaluated All equations are based on semantic relatedness and Vector Space Model (which is Java archive search engine's default retrieval model). This research focuses on improving Java archive search engine effectiveness using semantic relatedness. Since many relatedness algorithms are ontology-based, WordNet is selected as base ontology in this research.

II. RELATED WORKS

A. The Variations of Vector Space Model

Vector space model (VSM) is a retrieval model that represents query and documents as vectors of identifiers. This model was first published in 1975 by Salton et al [11]. Although this model is considered outdated, this model is still frequently used because of its simplicity and wide-applicability [12]. Not all uses of vectors and matrices count as vector space models [12], Turney & Pantel states that the values of the elements in a VSM must be derived from event frequencies, such as word occurences.

There are currently three broad classes of Vector Space Model which are based on term–document, word–context, and pair–pattern matrix (Actually there are more VSM classes but most of VSM variations are classified to these classes) [12]. Term-document matrix VSM is the most conventional VSM which is used to calculate the similarity of document (in search engine task, query is considered as pseudo-document). The vectors of similar documents are assumed to be nearby againts each other. Word-context matrices VSM focuses on word similarity, instead of document similarity. Deerwester et al. claims word similarity can be achieved by looking at row vectors in the

term–document matrix, instead of column vectors [13]. A hypothesis called distributional hypothesis in linguistics claims that words that occur in similar contexts tend to have similar meanings. This hypothesis is the justification for applying the VSM to measuring word similarity. Elements in word vector are derived from word occurences which is obtained from various context. Lund et al. uses windows of word to calculate word occurences [14] whereas Lin and Pad´o & Lapata use grammatical dependencies [15] [16]. Richer contexts such as dependency links and selectional preferences on the argument positions are also used [17]. In a pair–pattern matrix VSM, row vectors correspond to pairs of words and column vectors correspond to the patterns in which the pairs co-occur. Lin & Pantel use pair-pattern VSM for measuring the semantic similarity of patterns [18] whereas Turney et al. use it for measuring the semantic similarity of relations between word pairs [19]. The latent relation hypothesis claims that pairs of words that co-occur in similar patterns tend to have similar semantic relations [20].

The wide-applicability of VSM enables it to solve many other semantic tasks. Sarkar uses VSM approach to merge gene disease knowledge inferred across Online Mendelian Inheritance in Man, GenBank, and Medline [21]. Rapp adopt a vector-based representation of word meaning to achieve greater score than average human score on multiple-choice synonym questions from the Test of English as a Foreign Language (TOEFL) [22]. Turney used vector-based representation of semantic relations to solve multiple-choice analogy questions from the SAT college entrance test [23]. Cunha et al. integrate VSM, Enertex systems, Yate term extractor, and the Disicosum system to build a hybrid summarizer [24]. Castells et al adopt vector space model for ontology-based information retrieval [25]. They combine conventional keyword-based retrieval with ontology-based retrieval to improve its tolerance against knowledge base incompleteness. Santos et al. enchance topic-based VSM for semantic-aware spam filtering [26] whereas Tous & Delgado uses VSM for semantic similarity calculation and OWL ontology alignment [27]. Bratsas et al. uses ontology-based VSM and fuzzy query expansion to retrieve knowledge on medical computational problem solutions [28]. In previous research VSM also used as retrieval model in Java archive search engine [1].

VSM are extended into several models such as Generalized Vector Space Model, Latent Semantic Analysis (Latent Semantic Indexing), and Random Indexing. Generalized Vector Space Model introduces a term to term correlations, which deprecate the pairwise orthogonality assumption [29]. Latent Semantic Analysis is a concept-based VSM which assumes that there is some latent structure in the word usage [30]. Random Indexing is a dimension reduction method and computational framework for distributional semantics [31].

Entending VSM does not always change its concept. Some VSM variations only extend its weighting scheme without changing the model itself [1] [32]. VSM weighting scheme is initially based on word occurrences / term frequencies (which has some variations such as binary, raw, log normalization, and double normalization K) [2]. However, some high-frequency common words can not distinguish relevant and non-relevant documents since it appears in almost all documents. This issue is solved using a weighting scheme called tf-idf (term frequency–inverse document frequency) [32]. Tf-idf reflects how important a word to a document in a collection through Idf component. There are many variations on tf-idf weighting which are the results of many researches. Term Discrimination is an extension of VSM which rank keywords in how useful they are [33]. It works quite similar to tf-idf but it deals with finding keywords suitable for information retrieval and ones that are not. Some weighting schemes are also embedded with context-specific domain scheme. For example, weighting scheme in previous research weight a term based on control flow graph in program [1]. Fautsch & Savoy add specificity measures derived from either information theory or corpus-based linguistics on tf-idf weighting scheme [34].

B. Semantic Relatedness and Similarity

Polyvyanyy states that there are several linguistic phenomena in natural languages which can be seen in Table I [35]. Most of them are tightly related with relatedness except composition, homography, and word-groups. Two terms are considered related when both of them have one or more relation between them (this includes is-a relation, part-of relation, affect relation, and many other relations). Similarity is just a subset of relatedness which is is-a relation. Many well-developed semantic relatedness measurement algorithms have been developed in some researches (e.g. Hirst & St-Onge [3], Leacock & Chodorow [4], Banerjee & Pedersen [5], Wu & Palmer [6], Resnik [7], Jiang & Conrath [8], Lin [9], and Patwardhan & Pedersen [10]). Pedersen states that semantic relatedness measurements are categorized based on how they measure the relatedness which can be seen in Table II [10]. Patwardhan & Pedersen proposed the latest semantic relatedness measurement which is based on definition and corpus measurement.

TABLE I
LINGUISTIC PHENOMENA

Linguistic Phenomena	***Description***	***Relatedness***
Synonyms	Two or more words are interchangeable because of their similar (or identical) meanings.	yes
Inflection	The modification or marking of a word to reflect information, such as gender, tense, number or person of a target word.	yes
Composition	The word forming process where formed word consists of more than one free morpheme.	no
Derivation	The process of creating new	yes

Linguistic Phenomena	*Description*	*Relatedness*
	lexemes from other lexemes. Derivational affixes can also modify the meaning.	
Hyponyms	Specific instances of a more general word.	yes
Meronymy	The process to denote a constituent part of, or a member of something relation.	yes
Homography	Words with the same orthography but different meaning.	no
Metonymy	The substitution of one word for another with which it is associated.	yes
Word-groups	Clusters of words that have particular semantic meanings when they are grouped together.	no

TABLE II
SEMANTIC RELATEDNESS MEASUREMENTS

Measurement Category	*Related researches*
Path-based	Rada et al., 1989 Hirst & St-Onge, 1998
Path + Depth	Wu & Palmer, 1994 Leacock & Chodorow, 1998
Path + Information Context	Resnik, 1995 Jiang & Conrath, 1997 Lin, 1998
Definition-based	Lesk, 1986 Banerjee & Pedersen, 2002
Definition + Corpus Measurement	Patwardhan & Pedersen, 2006

Since most semantic relatedness algorithms determine term pair relatedness based on their existence in a ontology, a complete ontology is needed for measuring semantic relatedness. Although semantic relatedness measurements aim similar goal, they yield different range of result (e.g. Wu & Palmer and Lin's result ranged from 0 to 1 [6] [9], Hirst & St-Onge's result ranged from 0 to 16 [3], and Banerjee & Pedersen and Jiang & Conrath's result limited as non-negative floating numbers [5] [8].). Because of their various result range, Their results cannot be compared to each other based on their values. They can only be compared based on dataset evaluation.

C. Research Contribution

One of the major problems in lexical exact matching is that a relevant document is considered irrelevant when it contains no query term [2]. Some modification on text acquisition and transformation (e.g. stemming and n-grams) may be applied to overcome this problem, but many linguistic phenomena are still not handled. Stemming enables Information Retrieval (IR) system to recognize inflected and derived words by converting them to their word stem, base or root form whereas n-grams enables IR system to recognize word-groups and homograph words. As mentioned in previous section, many linguistic phenomena are tightly related with relatedness. Tsatsaronie & Panagiotopoulou use semantic relatedness as embedded component in Generalized Vector Space Model and state that semantic information can boost retrieval model performance [36]. Their research became the major motive in this research to embed semantic relatedness on extended VSM. There are three reasons why VSM is still used as retrieval model instead of Generalized Vector Space Model or raw semantic relatedness measurement:

a) VSM is the benchmark of many retrieval model because of its simplicity and wide-applicability.
b) VSM can be easily extended. Many researches extend VSM and gain fairly good results.
c) VSM enables document similarity measurement without relying on an ontology. Many terms in Java archive are name entity or specific-context terms which semantic relatedness cannot be measured since they are not occur in common English ontology.

In this research, Semantic relatedness measurement is embedded in extended VSM to overcome linguistic phenomena on Java archive search engine. Choosing the best semantic relatedness measurement in extended VSM is not the focus of this research. This research focuses on developing the most effiective extended VSM. Some additional equation schemes in VSM are also proposed and evaluated such as sum up all related term, substituting non-existing term with most related term, logaritmic normalization, context-specific relatedness, and low-rank query-related retrieved documents.

III. DATASET AND EVALUATION SCHEME

Dataset used in this research is obtained from previous research. It consists of 552 Java archives and 1860 queries which has 146 MB size in total [1]. Evaluation is based on several schemes which are explained in this section.

As mentioned in previous section, most semantic relatedness measurement algorithms are ontology-based which lead us to use WordNet. WordNet is a large lexical database of English which categorize nouns, verbs, adjectives and adverbs into sets of cognitive synonyms [37]. Since comparing the best semantic relatedness measurement in extended VSM is not the focus of this research, a semantic relatedness measurement is chosen as a benchmark of semantic relatedness (Lin's algorithms). Lin states that two terms are similar if they share much commonality and few differences [9]. The maximum similarity between two terms occur when they are identical, no matter how much commonality they share. These statement yield its result to be ranged from 0 to 1 (0 = not related at all, 1 = identically related). Lin's measurement is chosen because of several factor such as:

a) Lin's measurement result range (0 to 1) is easier to be normalized and treated as percentage of relatedness between two terms.

b) Lin's measurement is newer than another 0-to-1-result measurement that is discussed in this research (Wu & Palmer) [6] [9].
c) Although Lin only measure about similarity, this measurement can be used as a benchmark of relatedness since similarity is a core subset of relatedness and many similar terms are strongly related. The result of involving only strong related term pairs and involving all related term pairs should be similar since relatedness in semantic term matching is mainly affected by strongly related term pairs.

Since semantic relatedness can only be measured if both terms are listed as ontology terms, all terms involved in these scheme must not be stemmed (document and query terms). However, semantic relatedness between all terms must be measured at indexing step to keep the search engine efficiency. Three scheme were proposed based on terms used in semantic relatedness measurement at indexing step which are:

a) Type I: All distinct terms in index
b) Type II: Type I + All distinct query terms which is not index terms.
c) Type III: Type II + All stop words

The first scheme measures semantic relatedness of all distinct term in index which is possible to be used in real search engine. The second and third schemes are not applicable in real search engine since query terms may be vary based on user. But for research purpose, these scheme are still used to measure semantic relatedness impact. Stop words used in type III consists of 95 terms which is divided into six categories. These stop words are obtained from previous research [1]. All term pairs measured in this step is limited to unigram terms since many related term pairs are unigram and measuring n-gram relatedness is not feasible in term of efficiency. When there are two or more relation between a term pair, the highest relatedness score is considered as its relatedness score since this research only focus on strong term pair relation.

The statistics of related term pairs in each scheme can be seen in Table III. The second row is measured using equation (1) which is based on the number of comparison between distinct terms (*n*). The number of related term pairs found are considerably small since most terms in Java archive are name entity or specific-context terms. Although few in number, related term pairs are expected to improve search engine effectiveness.

TABLE III
STATISTICS OF RELATED TERM PAIRS

Statistic Variable	***Type***		
	I	***II***	***III***
Distinct Terms *(n)*	40.978	41.055	41129
Related term pairs if all terms are related to each other	839.618.731	842.777.040	845.817.885
Related term pairs found	4.952.917	4.993.301	4.993.609
The percentage of related term pairs found	0,5899 %	0,5924 %	0,5903 %

$$totalRTP(n) = n + (n-1) + (n-2) +... + 1$$
$$= (n^2 + n) / 2 \quad (1)$$

In previous research, file name, class name, field name, method name, method content, method expansion constant, control flow weighting, loop constant, recursive method expansion constant, and maximum n-gram constant are selected as document features. These features are evaluated using the impact of feature absence scheme which can be seen in (2). The impact of feature *f* is measured by calculating the difference between two schemes. The first scheme combines all features with default parameter value (true for boolean–based feature and 1 for numerical-based feature) whereas the second scheme is quite similar with the first scheme except it excludes *f* (false if *f* is boolean-based feature and 0 if *f* is numerical-based feature). All positive-impact features are combined to gain best feature combination which yield the best mean average precision (MAP). Best feature combination consists of file name, class name, method content, control flow weighting, loop constant 1, and maximum n-gram constant 3. Default (type A) and the best feature combination (type B) are then used as benchmark for each equation scheme in this research.

$$impact(f) = result(default) – result(default – f) \quad (2)$$

There are six schemes in total used for each equation scheme which are the combination of semantic relatedness scheme (3 schemes) and feature selection scheme (2 schemes). For simplicity, these schemes are redeclared below:

a) Type IA: Default feature combination + semantic relatedness of all distinct terms in index.
b) Type IIA: Default feature combination + semantic relatedness of all distinct terms in index and query.
c) Type IIIA: Default feature combination + semantic relatedness of all distinct terms in index, query, and stop words. This scheme exclude stopping on its process since stop words are included.

d) Type IB: Best feature combination + semantic relatedness of all distinct terms in index.
e) Type IIB: Best feature combination + semantic relatedness of all distinct terms in index and query.
f) Type IIIB: Best feature combination + semantic relatedness of all distinct terms in index, query, and stop words. This scheme exclude stopping on its process since stop words are included.

IV. Research Roadmap

Aiming for the most effective extended VSM, this research is organized as follows:

a) Measuring the impact of stemming and stopping. They are measured as benchmark scheme of this research.
b) Measuring the effectiveness of VSM + sum up all related terms.
c) Measuring the effectiveness of VSM + substituting non-existing term with most related term
d) Measuring the effectiveness by reducing the impact of semantic relatedness in (b) and (c) using logarithmic normalization.
e) Measuring the effectiveness of VSM + substituting non-existing term with most context-specific query-related term.
f) Measuring the effectiveness of VSM + considering non-retrieved documents as low-rank retrieved documents using semantic relatedness.

A. The Impact of Stemming and Stopping

Since stemming and stopping are involved as parameters in this research, the impact of stemming and stopping are also evaluated (which can be seen in Table IV and Table V). Each type consists of two binary digit where the first digit represent stemming and the second digit represent stopping. 1-valued digit represent the feature existence and 0-valued digit represent the feature absence. For example, type 10 represent the use stemming without stopping. Stemming is conducted using Porter stemmer since it works well on English words. As seen in Table IV and Table V, Stemming improves precision by recognizing inflected and derived words as their word stem whereas stopping reduce recall since some terms are removed in search engine. Both of them reduce index memory size which improve its efficiency. Since non-stemmed terms are required in semantic relatedness measurement, type 01 and 00 are selected as benchmark in this research. For simplicity, this scheme is called as *S* where its type is depend on related term pairs used (see semantic relatedness measurement in dataset and evaluation scheme section). If term pairs used are type I and II, this scheme will used type 01 as benchmark. Type 00 will be used when term pairs used is type III.

B. VSM + Sum up All Related Terms (SA)

The first extended VSM scheme proposed in this research assumes that all related terms are unambiguously related and may strengthen the relevance of relevant documents. A document is considered relevant if and only if it contains query or query-related terms. The more query or query-related terms a document contains, the more relevant a document is. Under an assumption that relatedness between two terms represents how related they are, score of query-related terms are calculated by multiplying the percentage of relatedness with its tf-idf score. The similarity of query *q* and document *d* is measured using (3). *n* is the number of query terms and *m* is the number of document terms. *rel(q,t)* is semantic relatedness measurement between term *q* and *t* whereas *tfidf(t,d)* is tf-idf score of term *t* in document *d*.

TABLE IV
The Impact of Stemming and Stopping in Default Feature Combination

Measurement	Type			
	11	*01*	*10*	*00*
Mean Average Precision (%)	**61,184**	61,047	61,018	60,511
Recall (%)	93,314	91,795	**93,73**	92,319
Index Size (MB)	**4,887**	5,687	5,14	5,98

TABLE V
The impact of Stemming and Stopping in best Feature Combination

Measurement	Type			
	11	*01*	*10*	*00*
Mean Average Precision (%)	**69,282**	68,59	69,048	68,289
Recall (%)	94,369	93,137	**94,732**	93,661
Index Size (MB)	**36,136**	37,833	36,331	38,052

$$s(q,d) = \frac{\sum_{i=0}^{n}(tfidf(q_i,d)+\sum_{j=0}^{m} rel(q_i,t_j)*tfidf(t_j,d))}{\sqrt{\sum_{j=0}^{m} tfidf(t_j,d) +\sum_{i=0}^{n} tfidf(q_i,q)}} \quad (3)$$

This extended VSM scheme is evaluated using six schemes and its evaluation result can be seen in Table VI. MAP and recall difference is measured by substracting MAP or recall gained in this scheme with MAP or recall gained in *S* scheme (the benchmark scheme). Type I and type II are compared with type 01 whereas type III is compared with type 00. In general, precision of this scheme is still lower than the precision of *S* scheme. Many relevant documents are ranked lower than it should be since retrieval process is majorly affected by the sum of all query-related terms. The score of query-relevant terms is significantly higher than the score of exact-match query terms. This issue leads irrelevant documents which has many query-related terms to be ranked higher than relevant document which has fewer query-related terms. The score of exact-match query

terms should be higher than query-related terms since exact-match query term is more relevant than query-related term. Although exact-match or identical term pair always considered as "most related" in relatedness measurement, not all exact-match term pairs are considered as it is since a term pair is considered related if and only if both of them exist in ontology. VSM is used as core similarity measurement to handle this issue in Java archive search engine since many terms used in Java archives are uncommon and do not occur in English ontology. Recall of this scheme is higher than *S* scheme since this scheme expand retrieval rule by including query-related terms instead of exact-match query terms.

TABLE VI
EVALUATION RESULT OF VSM + SUM UP ALL RELATED TERMS

Factors	*Type*					
	IA	***IIA***	***IIIA***	***IB***	***IIB***	***IIIB***
MAP (%)	37,001	42,16	43,872	46,664	50,423	**50,423**
MAP Diff (%)	-24,04	-18,88	-16,63	-21,92	-18,16	-17,86
Recall (%)	96,078	96,24	96,347	96,764	96,871	**97,095**
Recall Diff (%)	4,283	4,445	4,028	3,627	3,734	3,318

The precision and recall of this scheme is improved when query terms are involved in relatedness measurement (type I to type II). From this statement, it can concluded that query-related terms may intensify the impact of query terms. Query-related terms may also be used as query term replacement which improves recall. Recall is improved proportionally as the number of related term pairs found (type I to III).

C. VSM + Substituting Non-existing Term with Most Related Term (M)

The second extended VSM scheme assumes that exact-match query document term must be prioritized to be used rather than query-related document term in each document. Query-related term is only used when the absence of exact-match query term occurs. This assumption resulted from conclusion gained on previous scheme that the absence of a term may be replaced by its related term based on its relatedness percentage. The similarity of query *q* and document *d* is measured using standard VSM except its tf-idf score of non-existing term are measured using (4). *qx* is the query term which is absent on document *d*, *m* is the number of document terms, $rel(q_x,t_{max})$ is semantic relatedness measurement between term *q* and most query-related term t_{max}, and $tfidf(t_{max},d)$ is tf-idf score of the most related term in document *d* with query term *qx*.

$$r(q_x,d) = rel(q_x,t_{max}) * tfidf(t_{max},d) \quad (4)$$

The evaluation result of extended VSM + substituting non-existing term with most related term scheme can be seen in Table VII. MAP and recall differences are measured using similar comparison mechanism in Table VI. This scheme yields greater result than previous scheme since the impact of relatedness is lowered. Related term is only used when a certain query term is absent and VSM score is only affected by the most related term (not all related terms).

Although it has higher precision than first scheme (*SA* scheme), its precision is still lower than the precision of type 01 and 00 in *S* scheme. Some relevant documents are ranked in lower position since the score of the most query-related term in irrelevant documents is sometimes higher than the score of exact-match query term in relevant documents. If there are a query term "ant" and two documents which are X (relevant) and Y (irrelevant). X has "ant" with a score of 2 and Y has "bug" with a score of 6. If "bug" and "ant" is half-related (50% of relatedness), the score of Y is still higher than X since "ant" score (2) is lower than "bug" score (6/2 = 3). According to this issue, type II yields lower precision than type I since type II involves more query terms in relatedness measurement. Precision comparison result between type I and II in this scheme is quite different with *SA* scheme. In *SA* scheme, type II has higher precision than type I since many relevant documents contains query and query-related term. Sum up all exact-match query term and all query-related terms intensify the score of many relevant documents (although its overall result is still low). In this scheme, the comparison result is reversed since relatedness is only measured on most related terms and does not rely on how many related term a document contains. This mechanism automatically removes semantic relatedness retrieval score intensification gained in *SA* scheme.

TABLE VII
EVALUATION RESULT OF VSM + SUBSTITUTING NON-EXISTING TERM WITH MOST RELATED TERM

Factors	*Type*					
	IA	***IIA***	***IIIA***	***IB***	***IIB***	***IIIB***
MAP (%)	53.824	53.717	53.032	**63.213**	63.168	63.181
MAP Diff (%)	-7,223	-7,33	-7,479	-5,377	-5,422	-5,108
Recall (%)	96.078	96.24	96.347	96.764	96.871	**97,095**
Recall Diff (%)	4,283	4,445	4,028	3,627	3,734	3,318

Type I has highest MAP since it involves less related term pairs than others. It prevents low-rank relevant document to be ranked down further. This issue is disguised by query-related terms score issue in *SA* scheme. Type III has lower MAP than type II since stop words in queries may ranked down some relevant documents. Recall resulted in this scheme evaluation is quite similar to first scheme which means that many non-retrieved relevant documents have at least one query-related term.

Stopping leads precision-recall tradeoff to occur (type II and type III). Type III removes stopping which improves recall in exchange of its precision reduction from type II. This phenomena is disguised by query-related terms score issue in *SA* scheme. Type IIB and IIIB is just a outlier case since stop words in type IIIB improve the rank of several relevant documents.

D. Log Normalization (LSA, LM, and LLM)

Since query-related terms score issue occurs in both of previous schemes (*LA* and *LM*), its impact is tried to be reduced by normalizing their score through log normalization. Log normalization used in this research can be seen in (5). *x* is the initial query-related term score that will be normalized and 10 is log normalization base constant. Initial relatedness value is added with 1 to ensure its result is positive. (6) and (7) are log normalized form of (3) and (4). (6) is log normalized form of (3) whereas (7) is log normalized form of (4). *tfidf(t,d)* is shortened to *tid(t,d)* to simplify this equations.

$$f(x) = {}^{10}log\ (x + 1) \quad (5)$$

$$s(q,d) = \frac{\sum_{i=0}^{n}(tid(q_i,d) + {}^{10}\log(\sum_{j=0}^{m} r(q_i,t_j) * tid(t_j,d)) + 1)}{\sqrt{\sum_{j=0}^{m} tfidf(t_j,d) + \sum_{i=0}^{n} tfidf(q_i,q)}} \quad (6)$$

$$r(q_x,d) = {}^{10}log(rel(q_x,t_{max}) * tid(t_{max},d) + 1) \quad (7)$$

The impact of query-related terms in *SA* scheme is reduced by applying log normalization on the sum of all tf-idf query-related terms. As mentioned in section IV.B, exact-match query terms should have greater impact than query-related terms. Improving the impact of exact-match query terms can be achieved by reducing the sum of all query-related terms score. Log normalization of *SA* (*LSA*) is evaluated and its result can be seen in Table VIII. MAP difference is measured by substracting MAP gained in *LSA* with MAP gained in *SA* scheme. As can be seen in Table VIII, MAP are greatly improved through log normalization and becomes quite similar with the result of type 01 and 00 in *S* scheme. Its MAP is still low since this scheme involves all related terms. The sum of many query-related terms in some irrelevant documents may yield them to be scored higher than low-rank relevant documents although it has been log normalized. It brings down the score of some low-rank relevant documents which lower MAP result. Recall gained in this scheme is similar with standard *SA* scheme since recall is not rank-dependant.

TABLE VIII
EVALUATION RESULT OF VSM + LOG NORMALIZED FORM OF SUM UP ALL RELATED TERMS

Factors	*Type*					
	IA	***IIA***	***IIIA***	***IB***	***IIB***	***IIIB***
MAP (%)	60,241	59,931	59,303	**67,875**	67,774	67,345
MAP Diff (%)	23,24	17,771	15,431	21,211	17,351	16,922
Recall (%)	96,078	96,24	96,347	96,764	96,871	**97,095**

The score of most query-related replacement term in *M* scheme is log normalized to reduce its impact. Its impact need to be reduced since the most query-related term in some irrelevant documents is scored higher than exact-match query term in relevant documents. Log normalized on most related term in *M* scheme (*LM*) is evaluated and its result can be seen in Table IX. MAP difference is measured by substracting MAP gained in this scheme with MAP gained in *M* scheme. Its MAP result is higher than MAP result of *LSA* scheme since the usage of semantic relatedness in *LM* scheme is slightly lower than *LSA* scheme. Semantic relatedness is only used as replacement of missing query term on documents.

Although both scheme (*SA* and *M* schemes) are log normalized, query-related terms score issue still occurs since log normalization is unable to ensure exact-match query terms to have higher score than query-related terms. Double log normalization is also tried on *M* scheme in order to diminish query-related terms score further. But, its MAP is still lower than type 01 and 00 in *S* scheme. Double log normalization form can be seen in (8). *x* is the initial query-related term score that need to be double log normalized. This mechanism had been tried on *M* scheme which result can be seen it Table X. MAP difference is measured by substracting this scheme's MAP with *LM* scheme's MAP. As seen in Table X, its MAP is only improved by 0.3% in general. Its recall is still similar with the other schemes since documents retrieved by these schemes are similar. For simplicity, double log normalization form of *M* scheme is called *LLM*.

TABLE IX
EVALUATION RESULT OF VSM + SUBSTITUTING NON-EXISTING TERM WITH LOG NORMALIZED MOST RELATED TERM

Factors	*Type*					
	IA	***IIA***	***IIIA***	***IB***	***IIB***	***IIIB***
MAP (%)	60,447	60,437	59,899	**67,984**	67,953	67,707
MAP Diff (%)	6,623	6,72	6,867	4,771	4,785	4,526
Recall (%)	96,078	96,24	96,347	96,764	96,871	**97,095**

$$f(x) = {}^{10}log\ ({}^{10}log(x+1) + 1) \quad (8)$$

TABLE X
EVALUATION RESULT OF DOUBLE LOG NORMALIZATION OF *M* SCHEME

Factors	*Type*					
	IA	***IIA***	***IIIA***	***IB***	***IIB***	***IIIB***
MAP (%)	60,765	60,758	60,316	**68,314**	68,268	68,041
MAP Diff (%)	0,318	0,321	0,417	0,33	0,315	0,334
Recall (%)	96,078	96,24	96,347	96,764	96,871	**97,095**

E. VSM + Substituting Non-existing Term with Most Context-specific Query-related Term (CS)

This scheme is quite similar with *LLM* scheme except it has additional query-related terms restriction. All query-related terms are only involved if they belong to certain context. Restriction applied in this scheme is expected to reduce the usage of semantic relatedness. Context is defined by measuring relatedness between document term with context terms which is selected based on search engine context. "computer" is used as context term since search engine used in this research is computer-related. Before applying double log normalization, replacement terms score in this scheme is measured using (9) instead of (4). x is a context term which represent the context used in this scheme and t_{max} is document term which has the highest concept score. Concept score are resulted from multiplication of relatedness between t_{max} and concept term, and relatedness between t_{max} and the missing query term. Multiplication sign is used since both relatedness affect each other. *tfidf(t,d)* is shortened to *tid(t,d)* to simplify the equation.

$$r(q_x,d) = \mathrm{rel}(q_x, t_{max}) * \mathrm{rel}(x, t_{max}) * tid(t_{max},d) \quad (9)$$

The statistic of "computer" term as context term in this research can be seen in Table XI. Type I, II, and III are the proposed schemes based on term pairs that included in semantic relatedness measurement (see Section III). "computer" related terms is just a small subset of the related term pairs which amount is slightly low (approximately 0.03% of all related term pairs). Since this amount is relatively small, many related term pairs that are out of context can be eliminated. Based on statistic, it can concluded that "computer" is quite relevant context term for Java archive search engine since the number of "computer" related term is increased proportionately as the number of related term pairs used in this research (type I to III)..

TABLE XI
THE STATISTIC OF CONTEXT TERM "COMPUTER"

Factors	*Type*		
	I	***II***	***III***
The number of related term pairs	4.952.917	4.993.301	4.993.609
The number of "computer" related terms	1.722	1.966	1.986
The percentage of "computer" related terms	0,0347	0,0393	0,0397

The evaluation result of this scheme can be seen in Table XII. MAP and recall difference is measured by substracting MAP or recall of this scheme with MAP or recall of *LLM* scheme. Effectiveness improvement is resulted as the sum of MAP and recall differences. Since many query-related terms are excluded, recall resulted in this scheme is lower than recall resulted in previous schemes. Although this scheme has lower recall, it has higher MAP than *LLM* scheme. Effectiveness improvement conducted by this scheme is near to zero which represent balace precision-recall tradeoff (or MAP-recall tradeoff). It can be concluded that this scheme's effectiveness is quite similar with *LLM* scheme except that this scheme reduces its recall in order to improve its precision.

TABLE XII
EVALUATION RESULT OF VSM + SUBSTITUTING NON-EXISTING TERM WITH MOST CONTEXT-SPECIFIC QUERY-RELATED TERM

Factors	Type					
	IA	**IIA**	**IIIA**	**IB**	**IIB**	**IIIB**
MAP (%)	60.967	60.963	60.501	**68.508**	68.502	68.264
MAP Diff (%)	0,202	0,205	0,185	0,194	0,234	0,223
Recall (%)	95.715	95.931	96.038	96.549	96.710	**96.818**
Recall Diff (%)	-0,363	-0,309	-0,309	-0,215	-0,161	-0,161
Effectiveness Improvement (%)	-0,161	-0,104	-0,124	-0,021	0,073	0,062

Although this scheme yields higher MAP, this scheme is only applicable to context-based search engine and cannot be applied on common search engine. Context terms used in this scheme must also be chosen carefully since some erroneous context term may yields faulty results. Context terms reduce the generality of term usage by excluding all out-of-context terms from semantic relatedness measurement. Erroneous context term may remove some relevant term pair relatedness which may reduce search engine effectiveness.

F. *VSM + Considering Non-retrieved Documents as Low-rank Retrieved Documents using Semantic Relatedness (LR)*

In previous schemes, some query-related terms may scored higher than exact-match terms which yields MAP reduction. Log normalization can minimize this issue but not solve it completely since it can't assure the score of query-related terms to be lower than the score of exact-match terms. Although context-based relatedness may improve its precision, it cannot be applied to general search engine and it will reduce its recall because of its specifity. Based on these statements, it can be concluded that this issue can only be removed by assuring that non-retrieved documents which is retrieved as the result of relatedness should be ranked lower than any exact-match retrieved documents regardless of their score. All documents which have at least one exact-match query term will be ranked first using standard VSM and the rest of documents will be ranked using relatedness and concatenated at the end of VSM retrieved documents list. This mechanism should remove query-related terms score issue since relatedness is only used on VSM non-retrieved documents and their rank always lower than original VSM retrieved documents.

VSM similarity measurement in this scheme can be seen in Figure 1. Low-rank retrieved documents are non-retrieved documents which are considered retrieved as a result of semantic relatedness. They are concatenated at the end of retrieved documents list to assure they are ranked lower than exact-match retrieved documents. This similarity measurement consists of three steps which are:

a) Standard exact-match VSM: All exact-match retrieved documents are filtered and ranked using exact-match similarity.
b) Query-relatedness-based VSM: All non-retrieved documents which has at least one query-related term in their document are ranked and merged as low-rank retrieved documents. Relatedness measurement is used instead of exact-match similarity. The missing query term is replaced by most query-related term which score is measured using (4). Log normalization is not required since their rank will always be lower than exact-match documents.
c) Result concatenator: Final retrieved documents are formed by concatenating low-rank retrieved documents at the end of retrieved documents.

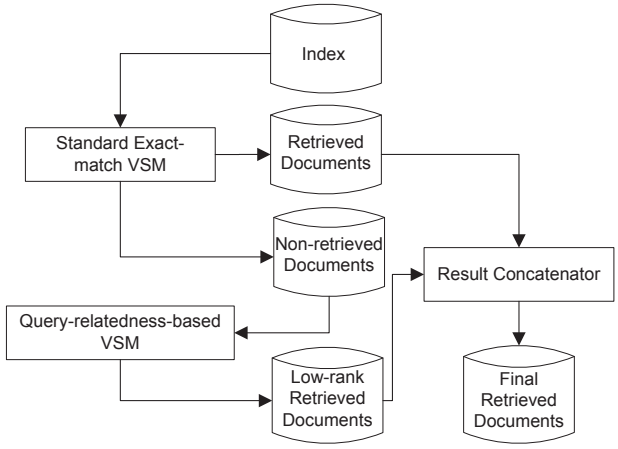

Figure 1 Modified VSM Similarity Measurement

The evaluation result of this scheme can be seen in Table XIII. MAP and recall difference is measured by substracting MAP or recall of this scheme with MAP or recall of *CS* scheme. This scheme yield excellent result which combines the advantage of semantic relatedness in recall with the advantage of exact-match VSM in MAP. Its recall is similar with most of previous schemes in this research (except concept-related scheme) which is improved since many non-retrieved relevant documents are retrieved as a result of query-related terms. Its MAP is the highest MAP among all MAP resulted in previous schemes since all query-related retrieved documents are concatenated at the end of exact-match retrieved documents (Query-related documents cannot distrupt exact-match document rank). When compared with *CS* scheme, this scheme yield greater MAP and recall. This scheme can also be applied to many common search engine since context is not involved in similarity measurement.

TABLE XIII
EVALUATION RESULT OF VSM + CONSIDERING NON-RETRIEVED DOCUMENTS AS LOW-RANK RETRIEVED DOCUMENTS USING SEMANTIC RELATEDNESS

Factors	Type					
	IA	**IIA**	**IIIA**	**IB**	**IIB**	**IIIB**
MAP (%)	61,079	61,0764	60,536	**68,613**	68,612	68,31
MAP Diff (%)	0,112	0,113	0,035	0,105	0,110	0,046
Recall (%)	96,078	96,24	96,347	96,764	96,871	**97,095**
Recall Diff (%)	0,363	0,309	0,309	0,215	0,161	0,277

V. COMPARISON OF ALL PROPOSED EXTENDED VSMS

In this research, many modified VSM scheme are proposed and evaluated to aim the most effective extended VSM using semantic relatedness. MAP and recall comparison of these scheme can be seen in Figure 2 and Figure 3 whereas their legend can be seen in Table XIV.

The comparison is limited in best feature combination scheme since default features scheme also produce similar result pattern. As can seen in these figures, involving semantic relatedness may yield higher recall and lower MAP.

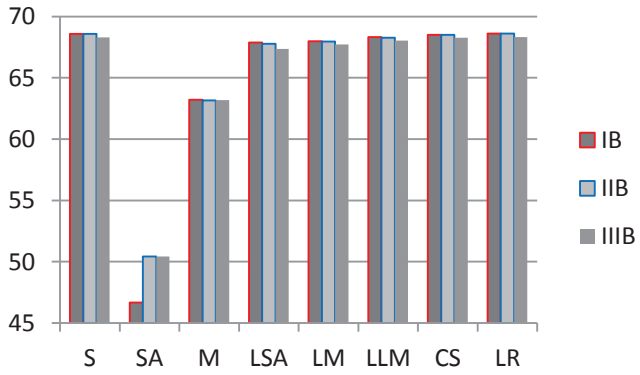

Figure 2 MAP Comparison betweenModified VSM Schemes

Lowest MAP is gained in *SA* scheme since this scheme involves all query-related terms by summing up their relatedness score using (3). Many schemes has lower MAP than standard VSM (S scheme) since query-related terms may disrupt retrieved documents rank. Highest MAP is gained in *LR* scheme since query-related documents are assured to be ranked lower rhan exact-match documents. This assumption also yield this scheme to be the only one scheme which has higher MAP than standard VSM.

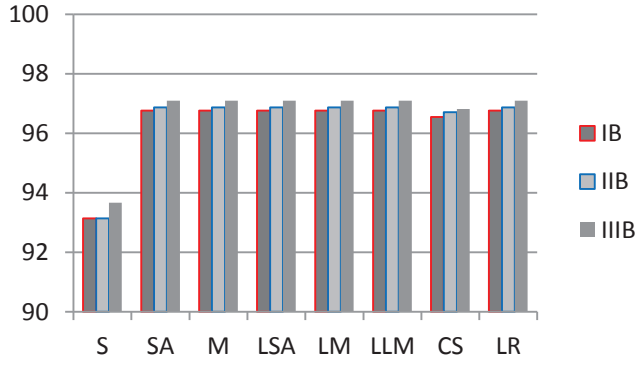

Figure 3Recall Comparison betweenModified VSM Schemes

TABLE XIV
LEGEND FOR FIGURE 2 AND 3

Symbol	*Scheme*
S	Default Scheme IB dan IIB: Non-stemmed standard VSM with best features combination IIIB: Non-stemmed standard VSM with best feature combination without removing stop words
SA	VSM + sum up all related terms
M	VSM + substituting non-existing term with most related term
LSA	VSM + log normalization of sum up all related terms
LM	Log normalization of VSM + substituting non-existing term with most related term
LLM	Double log normalization of VSM + substituting non-existing term with most related term
CS	VSM + substituting non-existing term with most context-specific query-related term
LR	VSM + considering non-retrieved documents as low-rank retrieved documents using semantic relatedness

Similarity measurement in these proposed schemes are expanded with relatedness which yield higher recall than standard VSM. Recall resulted in these proposed schemes is quite similar to each other since recall is not affected by documents rank and semantic relatedness involved in these schemes conduct similar retrieval rule. A document is retrieved if it has at least one exact-match query term or query-related term. Context-based relatedness used in *CS* scheme yields lower recall because of its specifity.

In each scheme, type I involves the least number of related term pairs which yields the highest MAP and lowest recall among all types. On the contrary, type III has the lowest MAP and highest recall since it involves the most number of related term pairs (including query-related stop words). It can be concluded that relatedness yields lower MAP by distrupting retrieved documents rank although it may yield higher recall since query-related documents are retrieved. Relatedness may yield higher MAP if and only if query-related terms score issue has been removed (which is applied in *LR* scheme).

Although *LR* is the most effective model among all the proposed model and non-stemmed standard VSM model, its MAP is still lower than stemmed standard VSM scheme (which is used as default retrieval model in previous research). *LR* has lower MAP since *LR* removes stemming on its text preprocessing. Stemming convert all derived terms to its stem / root form which may help retrieval model to see many derived terms as one similar root term. Comparison between these model can be seen in Table XV. ***SS*** represent **S**temmed **S**tandard VSM model whereas diff represent the difference between *SS* and *LR* in certain factor. However, *LR* has higher recall than stemmed standard VSM

since the rule of retrieving a document is generalized to "a document is retrieved if it has at least one query term or query-related term". Effectiveness improvement shown in Table XV shows that *LR* is more effective than stemmed standard VSM since it has 1.754% higher effectiveness.

TABLE XV
COMPARISON BETWEEN STEMMED STANDARD MODEL AND LR

Factors	*Type*		
	SS	***LR***	***Diff***
MAP	69,282	68,31	-0,972
Recall	94,369	97,095	2,726
Effectiveness improvement			1,754

VI. CONCLUSIONS

From research roadmap and evaluation results, several conclusion can be stated which are:

a) Relatedness may improve recall since more documents are retrieved. Retrieval rule is expanded which is not only limited to exact-match terms but also query-related terms. This statement solves exact-match term relevancy problem in VSM.
b) MAP reduction may occur if query-related term score is permitted to be higher than exact-match term score. Query-related term score should be lower than exact-match term score (this assumption is used in *LR* scheme).
c) Selecting a most query-related term is more query-relevant than involving all query-related term. It can be seen that *M* scheme has higher MAP than *SA*. This is caused by the sum of all related terms are bigger than the score of a most related term.
d) *LR* is the most effective scheme among proposed schemes since it utilize the advantage of relatedness without affected by its disadvantage (by assuming that relatedness score should be ranked lower than standard exact-match score). *LR* yields higher MAP than non-stemmed standard VSM although it has similar recall.
e) Although *LR* has lower MAP than standard stemmed VSM, its recall is much higher because of relatedness usage. By sum up MAP and recall improvement in *LR* scheme, it can be seen that *LR* is more effective than standard stemmed VSM.

VII. FUTURE WORK

In next research, document and query terms enrichment will be used and evaluated since terms in Java archive is limited. Certainly, this enrichment is applied to improve its effectiveness which lead a better Java archive search engine. Some feature weighting mechanism will also be applied since some Java archives features affect more than the others.

REFERENCES

[1] O. Karnalim, "Java Archives Search Engine Using Byte Code as Information Source," in International Conference on Data and Software Engineering (ICODSE), Bandung, 2014.
[2] B. Croft, D. Metzler and T. Strohman, Search Engine: Information Retrieval in Practice, Boston: Pearson Education.Inc, 2010.
[3] C. Leacock and M. Chodrow, "Combining local context and WordNet similarity for word sense identification," in WordNet: An Electronic Lexical Database, MIT Press, 1998.
[4] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," in WordNet: An Electronic Lexical Database, MIT Press, 1998.
[5] S. Banerjee and T. Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation using WordNet," in Appears in the Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, 2002.
[6] Z. Wu and M. Palmer, "Verb semantics and lexical selection," in ACL '94 Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Stroudsburg, 1994.
[7] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995.
[8] J. Jiang and D. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in Proceedings of International Conference Research on Computational Linguistics (ROCLING X), Taiwan, 1997.
[9] D. Lin, "An information-theoretic definition of similarity," in Proceedings of the15th ICM, Madison, 1998.
[10] S. Patwardhan and T. Pedersen, "Using WordNet-based context vectors to estimate the semantic relatedness of concepts," in Proceedings of the EACL, 2006.
[11] G. Salton, A. Wong and C.-S. Yang, "A vector space model for automatic indexing," in Communications of the ACM, 1975.
[12] P. D. Turney and P. Pantel, "From Frequency to Meaning: Vector Space Models of Semantics," Journal of Artificial Intelligence Research, vol. 37, pp. 141-188, 2010.
[13] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas and R. A. Harshman, "Indexing by latent semantic analysis," Journal of the American Society for Information Science (JASIS), vol. 41, p. 391–407, 1990.
[14] K. Lund, C. Burgess and R. A. Atchley, "Semantic and associative priming in high-dimensional semantic space," in Proceedings of the 17th Annual Conference of the Cognitive Science Society, 1995.
[15] D. Lin, "Automatic retrieval and clustering of similar words," in Proceedings of the 17th international conference on Computational linguistics, Stroudsburg, 1998.
[16] S. Pado and M. Lapata, "Constructing semantic space models from parsed corpora," in Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, 2003.
[17] K. Erk and S. Pado, "A structured vector space model for word meaning in context," in Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08), Honolulu, 2008.
[18] D. Lin and P. Pantel, "DIRT – discovery of inference rules from text," in Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2001, 2001.
[19] P. D. Turney, M. L. Littman, J. Bigham and V. Shnayder, "Combining independent modules to solve multiple-choice synonym and analogy problems," in Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03), Borovets, 2003.
[20] P. D. Turney, "The latent relation mapping engine: Algorithm and experiments," Journal of Artificial Intelligence Research, vol. 33, p. 615–655, 2008.
[21] I. Sarkar, "A vector space model approach to identify genetically related diseases," Journal of the American Medical Informatics Association, vol. 19, no. 2, pp. 249-254, 2012.

[22] R. Rapp, "Word sense discovery based on sense descriptor dissimilarity," in Proceedings of the Ninth Machine Translation Summit, 2003.
[23] P. D. Turney, "Similarity of semantic relations," Computational Linguistics, vol. 32, no. 3, p. 379–416, 2006.
[24] I. D. Cunha, S. Fernandez, P. V. Moralez, J. Vivaldi, E. Sanjuan and J. M. Torres-moreno, "A new hybrid summarizer based on Vector Space model, Statistical Physics and Linguistics," in MICAI'07 Proceedings of the artificial intelligence 6th Mexican international conference on Advances in artificial intelligence, Berlin, 2007.
[25] P. Castells, M. Fernandez and D. Vallet, "An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval," IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 2, pp. 161-272, 2007.
[26] I. Santos, C. Laorden, B. Sanz and P. G. Bringas, "Enhanced Topic-based Vector Space Model for semantics-aware spam filtering," Expert Systems with Applications: An International Journal, vol. 39, no. 1, pp. 437-444, 2012.
[27] R. Tous and J. Delgado, "A vector space model for semantic similarity calculation and owl ontology alignment," Database and expert systems applications. Lecture notes in computer science, vol. 4080, p. 307–316, 2006.
[28] C. Bratsas, V. Koutkias, E. Kaimakamis, P. Bamidis and N. Maglaveras, "Ontology-based vector space model and fuzzy query expansion to retrieve knowledge on medical computational problem solutions," in Proceedings of the 29th Annual International Conference of the IEEE EMBS Cité Internationale, Lyon, 2007.
[29] S. K. Wong, W. Ziarko and P. C. Wong, "Generalized vector spaces model in information retrieval," in SIGIR '85 Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval, New York, 1985.
[30] T. K. Landauer, "Latent Semantic Analysis," in Encyclopedia of Cognitive Science, John Wiley & Sons, Ltd, 2006.
[31] M. Sahlgren, "An introduction to random indexing," in Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering (TKE), Copenhagen, 2005.
[32] C. D. Manning, P. Raghavan and H. Schutze, Introduction to Information Retrieval, Cambridge: Cambridge University Press, 2009.
[33] F. Can and E. A. Ozkarahan, "Computation of term/document discrimination values by use of the cover coefficient concept," Journal of the American Society for Information Science, vol. 38, no. 3, pp. 171-183, 1987.
[34] C. Fautsch and J. Savoy, "Adapting the tf idf vector-space model to domain specific information retrieval," in SAC '10 Proceedings of the 2010 ACM Symposium on Applied Computing, New York, 2010.
[35] A. Polyvyanyy, "Evaluation of a novel information retrieval model: eTVSM," in Master's thesis, Hasso Plattner Institut, 2007.
[36] G. Tsatsaronis and V. Panagiotopoulou, "A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness," in EACL '09 Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, Stroudsburg, 2009.
[37] "WordNet," [Online]. Available: https://wordnet.princeton.edu/. [Accessed 17 March 2015].