May 2020

The Ensemble MESH-Term Query Expansion Models Using Multiple LDA Topic Models and ANN Classifiers in Health Information Retrieval

Sukjin You
University of Wisconsin-Milwaukee

THE ENSEMBLE MESH-TERM QUERY EXPANSION MODELS USING MULTIPLE LDA TOPIC MODELS AND ANN CLASSIFIERS IN HEALTH INFORMATION RETRIEVAL

by

Sukjin You

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Information Studies

at

The University of Wisconsin -Milwaukee

May 2020

ABSTRACT

THE ENSEMBLE MESH-TERM QUERY EXPANSION MODELS USING MULTIPLE LDA TOPIC MODELS AND ANN CLASSIFIERS IN HEALTH INFORMATION RETRIEVAL

by

Sukjin You

The University of Wisconsin-Milwaukee, 2020
Under the Supervision of Professor Xiangming (Simon) Mu

Information retrieval in the health field has several challenges. Health information terminology is difficult for consumers (laypeople) to understand. Formulating a query with professional terms is not easy for consumers because health-related terms are more familiar to health professionals. If health terms related to a query are automatically added, it would help consumers to find relevant information. The proposed query expansion (QE) models show how to expand a query using MeSH (Medical Subject Headings) terms. The documents were represented by MeSH terms (i.e. Bag-of-MeSH), which were included in the full-text articles. And then the MeSH terms were used to generate LDA (Latent Dirichlet Analysis) topic models. A query and the top *k* retrieved documents were used to find MeSH terms as topic words related to the query.

LDA topic words were filtered by 1) threshold values of topic probability (TP) and word probability (WP) or 2) an ANN (Artificial Neural Network) classifier. Threshold values were effective in an LDA model with a specific number of topics to increase IR performance in terms of infAP (inferred Average Precision) and infNDCG (inferred Normalized Discounted Cumulative Gain), which are common IR metrics for large data collections with incomplete judgments. The top *k* words were chosen by the word score based on (TP *WP) and retrieved document ranking in an LDA model with specific thresholds. The QE model with specific thresholds for TP and WP showed improved mean infAP and infNDCG scores in an LDA model, comparing with the baseline result. However, the threshold values optimized for a particular LDA model did not perform well in other LDA models with different numbers of topics.

An ANN classifier was employed to overcome the weakness of the QE model depending on LDA thresholds by automatically categorizing MeSH terms (positive/negative/neutral) for QE. ANN classifiers were trained on word features related to the LDA model and collection. Two types of QE models (WSW & PWS) using an LDA model and an ANN classifier were proposed: 1) Word Score Weighting (WSW) where the probability of being a positive/negative/neutral word was used to weight the original word score, and 2) Positive Word Selection (PWS) where positive words were identified by the ANN classifier. Forty WSW models showed better average mean infAP and infNDCG scores than the PWS models when the top 7 words were selected for QE. Both approaches based on a binary ANN classifier were effective in increasing infAP and infNDCG, statistically, significantly, compared with the scores of the baseline run. A 3-class classifier performed worse than the binary classifier.

The proposed ensemble QE models integrated multiple ANN classifiers with multiple LDA models. Ensemble QE models combined multiple WSW/PWS models and one or multiple classifiers. Multiple classifiers were more effective in selecting relevant words for QE than one classifier. In ensemble QE (WSW/PWS) models, the top *k* words added to the original queries were effective to increase infAP and infNDCG scores. The ensemble QE model (WSW) using three classifiers showed statistically significant improvements for infAP and infNDCG in the mean scores for 30 queries when the top 3 words were added. The ensemble QE model (PWS) using four classifiers showed statistically significant improvements for 30 queries in the mean infAP and infNDCG scores.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

There are lots of people helping me complete my dissertation: my advisor, Dr. Mu, other committee members, HPC (High-Performance Computing) Center at UWM, and other staff in my department, SOIS.

I cannot thank you enough, Dr. Mu, for your kindness and openness. He is always open to any ideas I have, encouraging to apply my ideas to the projects, which helped me enjoy my research. I could argue anything, although sometimes I am not knowledgeable in a specific area. He encouraged me to discuss anything I am interested in and gave novel ideas and instructions. His suggestions to solve the potential issues that may occur in the research motivated me by boosting my interest in the research. His support for upgrading existing computing resources (e.g. RAM and Hard disks) helped me to set up computing resources for the study. Dr. Wolfram, one of the scholars I respect, gave me a clear direction for productive results in publication. The fairness and generosity in treating project members, colleagues, and students would be reasons why he has led successful projects. His knowledge and instructions showed me who a role model of a scholar is. Dr. Zhang’s advice for the research method and design was valuable to complete this dissertation. Detailed comments and guidelines about research methodology were helpful to redesign my study. I cannot forget the time working with Dr. Joo. I realized how enjoyable collaboration with another scholar can be when I worked with Dr. Joo. His insight, passion, and attitude for research were qualifications a true scholar ought to have. I learned useful research skills, such as writing, methods, co-work skill through the projects. I appreciate his sincere advice/tips/help. Even though I had no chance to work with Dr. Lu, he was willing to serve as a committee member. His comments about mathematic formulas helped cover my short knowledge about math. Without help from the UWM HPC (high-performance computing) service, I could not finish my dissertation. Shawn, Jason, and Daniel helped me to use the cluster so that I can access lots of CPUs for multiple tasks. Also, I appreciate Dean Lipinski who helped to get granted to use the high-performance computing resources. Lastly, I appreciate the IT staff of SOIS, who always fixed issues that occurred in several computers used for this research promptly.

Chapter 1 INTRODUCTION

Information retrieval (IR) is the process and activity to find information matching a user's information need occurring in everyday life. Differently from the past, a huge amount of information is being created and shared in electronic formats on the Web in the world every day. It is getting challenging to find relevant information today. IR strategies may vary according to users' information needs. Information can be provided by a form of ranked results or categorized groups. More recently, advances in computational and statistical methods have made it possible to design IR systems that implement IR strategies using various techniques based on data/text mining and machine learning. The integration of these techniques could be more effective than depending on only one technique.

1.1 IR Paradigms

Two typical prevalent IR paradigms are query-based retrieval and browsing. Most classic models are query-based models (Belkin & Croft, 1987). A query is input, and a search engine displays retrieved results. Although query-based IR is still dominant in web-based search as well as database-based (collection-based) search, it has shown some weaknesses, such as difficulties of query formulation, empty or too many results retrieved, the dependency on text information (Cox, 1992). Browsing is another type of IR depending on a user's cognitive processing, including pattern recognition. Browsing reflects users' real information-seeking behavior as introduced in the berry-picking model (Bates, 1989). Users browse documents categorized by discipline, topic, journal, date, etc., or refer to taxonomies or ontologies to find more information about a topic, which can be more helpful in finding related information.

Classic IR models were developed for query/text/system-based IR. On the other hand, modern IR models are focused on browsing, filtering, recommendation, and multimedia/web/user-based IR. Quantitative approaches, such as machine learning, data mining, and natural language processing, have demonstrated better performance based on a large amount of data in IR.

Popular techniques applied to one side (e.g. browsing) can be effective on the other side (e.g. query-based IR), too. Ideas and techniques, which are originated from each IR approach have been integrated to make synergy in IR. Ensemble models based on various types of data and mechanisms have shown powerful IR performance by combining the strength of each model and complementing the weakness of each model.

1.2 The Ensemble of Topic Modeling & Classification in IR

Topics in a document collection are one of the critical elements in both IR approaches (query-based retrieval and browsing) because topics represent documents with key terms. Topic modeling is a useful technique to extract latent topics of a collection including a huge number of documents. Topic detection can be applied to not only the classification/clustering of documents for browsing but also query-based IR.

1.2.1 Topic modeling for query-based search

In query-based search, one popular way to improve IR performance is to add meaningful words following the original query. LDA (Latent Dirichlet Allocation, Blei, Ng, & Jordan, 2003) is the most common algorithm for topic modeling these days. An LDA model generates meaningful topic words representing the collection. Latent topic words generated by LDA can be candidate terms for query expansion (QE).

One weakness of the QE model using LDA topic words is that topic words might be too general to represent topics. Those words might not be helpful to retrieve a relevant document. Therefore, some dictionaries would be useful to filter out general words and identify key terms appropriate for a specific field. In the health domain, a health-related special terminology, such as MeSH (Medical Subject Headings) terms can provide more effective terms for QE.

1.2.2 The classifier for word selection

Another challenge in QE is how to select relevant terms from candidate terms. An LDA model identifies topics related to a given text, such as a query or a retrieved document and generates topic words related to the topics. Although the LDA model is a good tool to collect candidate words for QE, the selection of appropriate words is the following concern. Some topic words might be relevant for QE, but others are not. If there is a recommendation system for identifying relevant words for QE, it would be used to select relevant words for QE.

Classification is a common method to categorize information into existing sections useful for browsing. A classifier can be employed as a word recommendation system in query-based IR as well as browsing. The performance of a classifier is decided by an amount of training data and the effectiveness of used features. A classifier must be trained on enough data and effective features showing differences between relevant words and irrelevant words. Word features for classification can include dynamic features generated by LDA models given a query and static feature related to a collection or document. To collect enough data for training, LDA topic words can be generated based on a query and the top *k* (e.g. 10) documents retrieved by the query.

The ANN (Artificial Neural Network) has been popularly used in machine learning (supervised/unsupervised/reinforcement) including classification and regression. ANN models have shown superior performance to traditional machine learning techniques. The application of ANN classifiers to QE can contribute to increasing IR performance.

1.2.3 The ensemble of multiple LDA models and classifiers for QE

Integrating LDA models and classifiers might be effective to identify words for QE. The IR performance would depend on the performance of each LDA model and classifier—how well the LDA model generates relevant MeSH terms as topic words related to the query and how well the classifier can identify relevant MeSH terms for QE. If the performance of the LDA model or classifier is poor, the word, which is generated by the LDA model and selected by the classifier, would not be effective for QE.

The ensemble of multiple LDA models and classifiers would guarantee stable IR performance not depending on an individual model and classifier. Ensemble models would generate synergy from multiple models. Assuming even only a relevant word is recommended by each LDA model and 40 LDA models are used, 40 candidate words can be collected. The 40 candidate words would be more relevant than 40 words generated by only one LDA model. The candidate words can be evaluated by a classifier (positive/negative/neutral) or multiple classifiers. If a word is identified as a positive word for QE by multiple classifiers, the word would be more likely to be a positive word than when the word is classified into the positive word group by only one classifier.

1.3 Research Questions

In this paper, three research questions were answered about the effectiveness of LDA topic models and ANN classifiers in query-based IR. Effectiveness was assessed by comparisons with the baseline results without LDA models or ANN classifiers in terms of infAP and infNDCG. Two-sample t-tests or paired t-tests were conducted to see significant differences in mean scores.

1.3.1 RQ1) How effective is the application of LDA topic words based on MeSH terms for QE in health IR?

To address the RQ1, topic words comprising MeSH terms, which are related to a query and documents retrieved by the query (by an LM-based search engine, Terrier) were generated by an LDA topic model.

Referring to the word probability and the topic probability for a topic word is one way to measure the extent to which the word is related to the query or the top *k* retrieved document by the query. If the topic for the word is highly related to the query and retrieved documents and the word is highly related to the topic generated by LDA models, the word would be likely to be related to the query. For effective word filtering, thresholds for topic probability (TP), word probability (WP), and the values of (TP * WP) were set up. It was assumed that the LDA topic words contribute to achieving better performance in terms of infAP and infNDCG, compared with the original query, especially when topic words are selected with thresholds. Mean infAP and infNDCG scores of 40 LDA models were compared with the result of the baseline run by two-sample t-test.

1.3.2 RQ2) How effective is the application of LDA MeSH terms to QE in health IR when LDA topic words are weighted or selected by an ANN classifier?

An ANN classifier is designed with some dynamic (LDA-related) and static (corpus-related) features to judge if the selected topic words are relevant for QE. The binary or 3-class ANN classifiers categorized a word into two or three groups: positive/negative/(neutral) word groups.

Words for QE in an LDA model were selected in two ways: 1) the Word Score Weighting (WSW) model where a word score based on (TP*WP) and document rank was weighted by the probability estimated for the positive/negative/neutral word group using the ANN classifier and 2) the Positive Word Selection (PWS) model where a positive word was identified by the ANN classifier.

The top 7 or top 10 words by the weighted word score and positive words were added to the original queries. The effectiveness of the two models was examined and compared in creating query expansion terms over 40 LDA models. Two-sample t-tests were conducted to compare mean infAP and infNDCG scores of 40 WSW/PWS models with the result of the baseline run.

1.3.3 RQ3) How effective are the ensembles of multiple LDA models and ANN classifiers in selecting MeSH terms for QE in health IR?

Ensemble QE models collect words for QE from multiple LDA models (not by one LDA model) in two proposed ways (WSW & PWS). The ranking score of a word was calculated by one classifier or multiple classifiers. Of candidate words, the top *k* words by the ranking score were extracted for QE. The IR performance of the new query was evaluated by infAP and infNDCG over 30 queries. Mean infAP and infNDCG scores were compared between the best runs of the ensemble QE models and the baseline run for 30 queries by paired t-test.

1.4 Significance

1.4.1 Theoretical significance

Proposed QE models would be a foundation to establish general QE models based on LDA models and classifiers in IR. QE models using an LDA model and an ANN classifier are intended to reflect the dynamic and static nature of a word in selecting a word for QE by integrating the dynamic word features generated by an LDA model with static word features related to a corpus. The process of the proposed QE model is similar to the pseudo relevance feedback model in that the top *k* retrieved documents along with a query can be used to find related topic words using LDA topic models already generated based on a collection, but the process is a little different in that LDA topic models and ANN classifiers are involved in the word selection for QE.

Ensemble QE models using multiple LDA topic models and ANN classifiers introduce how the ensemble QE models can generate more effective IR results than the QE models using an individual LDA model and classifier.

The QE model including an LDA model and a classifier has two types: Word Score Weighting model (WSW) and Positive Word Selection (PWS). The ensemble QE models integrating multiple LDA models with ANN classifiers showed that ensemble models based on different types of machine learning techniques can increase IR performance in the health domain.

1.4.2 Methodological significance

Proposed QE models show how MeSH terms were generated from a query and retrieved results by an LDA model. Also, this study introduces how data for training a classifier can be generated on the evaluation scheme of the TREC CDS track.

Data (PMC) was collected through NCBI (the National Center for Biotechnology Information—part of the United States National Library of Medicine) FTP, which provides a huge number of full-text documents in open-access journals in the health field. The findings from the extensive set of data collection would be stable and general. The assessment of the IR tasks was conducted based on the evaluation scheme of the TREC CDS (Clinical Decision Support) track, which is known as a stable, trendy and verified scheme.

Topic words consisting of general terms, which are generated by LDA, might not be appropriate for QE. As a dictionary for representing the documents for IR, the MeSH terminology was used to choose the effective terms included in full-text documents.

LDA topics are generated by the developed module (g*ensim*) in Python. *gensim* is efficient in controlling the memory for a large amount of data by an LDA algorithm in a stochastic manner where parameters are estimated dramatically faster than batch algorithms on large datasets (Hoffman, Bach, & Blei, 2010).

ANN classifiers played a role in identifying relevant words for QE given a query It is critical to collect enough data in machine learning. This study proposes a method to collect data consisting of dynamic/static features from the corpus and LDA models. For creating training data, the word features values are created by multiple (40) LDA models and other collection-related python libraries (e.g. *gensim*) using the queries and the top *k* (*k*=10) retrieved documents.

1.4.3 Practical significance

The proposed QE models can be applied to IR systems to select relevant words as query expansion terms given a query. Existing PRF (Pseudo Relevance Feedback) algorithms would be integrated with the proposed QE models.

The proposed QE models are not limited to the static nature (e.g. TF, IDF, and CF) of a word related to a corpus because word features include dynamic features generated by LDA models given a query. Therefore, a word can have different features values according to a query. This study investigates if LDA models can perform better for QE when the ANN classifier was incorporated. A binary ANN classifier would be compared with a 3-class classifier in terms of infAP and infNDCG. If the classifier is trained on sufficient data and effective features with optimized tuning parameters for ANN, the top *k* words of the words recommended by multiple LDA models would contribute to improving IR performance. The ensemble QE models supported by multiple LDA models and ANN classifiers may be effective in complementing the weakness caused by an individual LDA model or classifier.

Chapter 2 LITERATURE REVIEW

2.1 Standard IR Models

Classic IR models can be classified into four types even though it might be argumentative: the Boolean model, Vector Space Model, Probabilistic Model, and Language Model (Hiemstra, 2009). The IR models can be compared with respect to the IR process. Classic models were optimized for classic IR systems, which are oriented to system-based and full-text IR. They have been varied to suit current IR systems. Some IR evaluations based on full-text articles including health information have been conducted by the system-based approach (e.g. TREC, Text REtrieval Conference, https://trec.nist.gov/). The Bag-of-Words Model and the Vector Space Model affect the representation of documents and the IR process on classic IR models.

IR models and topic models share a similar process and concepts. Text processing based on the Bag-of-Words Model and vector representation of a document is also required in topic modeling. The Language Model estimates the probability distribution of words to find documents related to a query. The probability distribution of words for a topic is an output in topic modeling. Factors for ranking in classic IR models, such as TF (Term Frequency), CF (Collection Frequency), and IDF (Inverse Document Frequency), give inspiration for IR improvement regarding similar concepts in topic modeling, including TP (Topic probability), WP (Word Probability), CTD (Collection Topic Density), and CTF (Collection Topic Frequency), which are explained later.

2.1.1 System-based and full-text IR

IR research can be separated into two approaches according to the IR procedure: 1) the system-based model and 2) the user-based approach (Moghadasi, Ravana, & Raman, 2013). In the user-based retrieval systems, interactions with users, including feedback/actions/behavior, are used in order to enhance retrieval performance. User participation is necessary for receiving feedback because an individual user's satisfaction level is a critical evaluation criterion. System-based retrieval approaches depend on their retrieval algorithms and methods and systematic IR performance evaluation rather than the interaction with users and their satisfaction level measurements. The similarity and overlapping between the retrieved documents and a set of assessed relevant documents is measured to evaluate search results including retrieved documents and ranking. Document corpus, queries, and judgment sets are the main components in the system-based IR system evaluation.

Full-text retrieval refers to the retrieval of full-text documents. Beall (2008) compared full-text searching with metadata-enabled searching. Full-text searching is a type of query-based IR used to obtain ranked results based on keyword/algorithmic/stochastic/probabilistic searching, while metadata-enabled searching (i.e. deterministic searching) is a more sophisticated type of browsing by matching search terms with terms in structured metadata. Full-text is the unstructured text based on natural language. The characteristic of the unstructured full-text might be one reason to make the IR system more complicated, which requires pre-processing of the text and interpreting natural language into a machine-understandable form. Limitations of classic IR models are revealed due to the full-text nature.

Manning, Raghavan, and Schütze (2008) made the distinction among three types of IR: RDB (Relational DataBase) search, unstructured retrieval, and structured (e.g. tree/hierarchy structure) retrieval. RDB search is the fastest but has difficulty for ranking. The semi-structured or structured text, such as the metadata or XML (eXtensible Markup Language) / SGML (Standard Generalized Markup Language), make IR efficient and effective because the structure (e.g. a tree or hierarchy) can be used for IR, while there are challenges in constructing the structure consistently for different types of documents. Unstructured retrieval is based on free-text queries, so convenient for users, compared with SQL (Structured Query Language) in RDB search. Unstructured retrieval is more appropriate for the (unstructured) full-text documents.

2.1.2 The Bag-of-Words Model

The Bag-of-Words model (Harris, 1954) is a basic assumption underlying in Boolean/Vector Space models, and even simple Probabilistic/Language models. In the Bag-of-Words model, a document is represented as a group of words. Syntactical structure and semantic implications are ignored, but only the lexicon is considered. The order of words in a document is meaningless. Only the term (word) is an element for representation.

Information needs are represented by a set of assigned keywords (a query) and matched to index terms representing documents. A query ignoring the order or proximity between words is likely to miss an exact meaning that can be captured from multi-gram words. The specification of information needs or problems is limited in IR models based on the Bag-of-Words model, although the representation process can be finished easily in a short time. Although IR systems based on n-gram words or semantics (e.g. context and situations) cannot be easily implemented in the Bag-of-Words model, those IR systems are effective to catch users' information needs.

2.1.3 The Boolean Model

The Boolean model (Lancaster & Fayen, 1973) is based on the notion of sets. Search results retrieved by Boolean operations cannot be ranked, unlike other IR models. Each term in a Boolean query statement is interrelated by three Boolean operators—AND, OR, and NOT. Each operation delimits search scope; OR is used in connecting synonymous terms into a broader scope, while AND is used to narrow the scope. NOT can be used for filtering undesired results out (Salton, Fox, & Voorhees 1985). It is difficult to search for phrases including more than two terms, unless new operators are not added such as the proximity operator, /, (e.g. "/3", within 3 words) because Boolean statements examine one term as a search unit. Although most Online Public Access Catalogs (OPAC) systems have adopted the Boolean IR model, Boolean queries consisting of Boolean operators and query terms are still unfamiliar to users.

Boolean operations based on "True" or "False", disclose some weaknesses. Two judgment values including true and false, make the ranking of results impossible, which only creates two result categories:

relevant or not. The number of results is unpredictable. There might be no results or a huge number of results. Users might be in trouble when the number of results is large. "No ranking" implies that it is difficult to refine an initial query after the initial search results because the feedback for query reformulation based on top-ranked results, is not feasible. However, if a reasonable number of relevant documents can be defined in the Boolean model, it is not impossible as Salton, Fox, and Voorhees (1985) proposed a feedback model on the Boolean model.

Representation: term-document incidence matrix vs. inverted index matrix. In the Boolean model, documents are represented by term vectors. A term-incident table includes values showing the relationship between a term and a document. The relationship is represented as "0" or "1". The cell for a term and a document is filled with "1" when the term is included in the document. Because most cells are filled with many '0's meaning "not included", the term-document matrix is very sparse. It is wasteful to save all the values of the matrix in the memory.

The inverted index matrix was designed to overcome this weakness. It links terms to related documents for the case of '1's in the term incidence matrix. It only saves meaningful values. Inverted index files include terms and corresponding document lists for each term with a linked list or array structure.

2.1.4 The Vector Space Model (VSM)

VSM (Salton, Wong, & Yang, 1975) was proposed for a ranked retrieval model; a query and documents are represented as vectors, and retrieval results are ranked by similarity score. In the VSM, terms included in a query can be weighted by a user. A query is considered as a short document containing a series of terms. Although a query is a short document that includes several terms, the order of terms does not affect the results (the Bag-of-Words model).

The relationship between documents and a query is represented by a similarity score that can be measured in several ways: distance measures such as Euclidian/Jaccard distances (Jaccard, 1901), cosine similarity using term weighting (Salton & Buckley, 1988), and Dice similarity (Dice, 1945). While are the

Boolean model allows binary values for the relation between a document and a query, the VSM allows other values, such as Term Frequency (TF), Document Frequency (DF), and Collection Term Frequency (CF) for the calculation of the similarity score. Scores are normalized by document length for the relative comparison.

2.1.5 The Probabilistic IR Model

In the probabilistic model, an IR system estimates the probability that a document (d) is relevant to a user query (q): P (R=1 | d, q) = P (R=1 | $\vec{x}$, $\vec{q}$). In the Binary Independence Model (BIM), which is the simplest probabilistic IR model introduced by Yu & Salton (1976), d is represented by the term incidence vector: $\vec{x} = (x_1, \ldots, x_M)$ where M is the number of terms. q is represented as a vector ($\vec{q}$) like a document. Theoretically, a partial set of judged documents is required for the calculation of the probability, which is a weakness of the probabilistic model.

A basic theory for the probabilistic model is Bayes' rule:

$$\mathrm{P}\ (\mathrm{A} \mid \mathrm{B}) = \mathrm{P}\ (\mathrm{B} \mid \mathrm{A}) \times \mathrm{P}\ (\mathrm{A})\ /\ \mathrm{P}\ (\mathrm{B}) \tag{2.1}$$

where P (A | B) = the posterior probability, P (B | A) = the likelihood, P (A) = the prior probability, and P (B) = the marginal likelihood (the total probability of observing the evidence). In Bayes' rule,

$$\mathrm{P}\ (R \mid \vec{x}, \vec{q}) = \mathrm{P}\ (\vec{x} \mid R, \vec{q}) \cdot \mathrm{P}\ (R \mid \vec{q})\ /\ \mathrm{P}\ (\vec{x} \mid \vec{q}). \tag{2.2}$$

P (R | $\vec{x}$, $\vec{q}$) is the probability of being a relevant/non-relevant document for a query. P ($\vec{x}$ | R, $\vec{q}$) is the probability that $\vec{x}$ is the relevant/non-relevant document for a query. P (R | $\vec{q}$) is the total probability of being a relevant/non-relevant document (e.g. no. relevant or non-relevant documents divided by no. all documents). P ($\vec{x}$ | $\vec{q}$) is the probability of the being a document (e.g. no. documents with $\vec{x}$ / no. all documents). Prior knowledge, such as a prior probability, P (R | $\vec{q}$), must be known in advance of IR task, however, it is assumed that the number of relevant documents is very small than the number of non-relevant documents.

Ranking is determined by the odds (O) of the probability (Manning et al., 2008, p. 224):

$$\mathrm{O}\ (\mathrm{R} \mid \vec{x},\ \vec{q}) = \mathrm{P}\ (R = 1 \mid \vec{x},\ \vec{q})\ /\ \mathrm{P}\ (R = 0 \mid \vec{x},\ \vec{q}) = \mathrm{O}\ (\mathrm{R} \mid \vec{q}) \cdot \prod_{t=1}^{M} \frac{\mathrm{P}\ (x_t \mid R = 1,\ \vec{q})}{\mathrm{P}\ (x_t \mid R = 0,\ \vec{q})} \tag{2.3}$$

The Probability Ranking Principle (PRP, Rijsbergen, 1979, as cited in Manning et al., 2008, p. 221) is implemented by the *Bayes Optimal Decision Rule* where the expected loss (Bayes Risk) is minimized by retrieving more likely relevant documents than non-relevant documents. In practice, the initial probability of term *t* appearing in a nonrelevant document is calculated by df_t (document frequency for the term *t*) / *N* where *N* = no. all documents (Manning et al., 2008, p. 227). The initial probability of term *t* in a relevant document can be set up as a specific value for the term: e.g. 0.5 (Croft and Harper, 1979, as cited in Manning et al., 2008, p. 227) or $(1/3 + (2 \cdot \mathrm{df}_t\ )\ /\ (3 \cdot N))$ (Greiff, 1998, as cited in Manning et al., 2008, p. 227). Those initial probabilities are updated in pseudo relevance feedback where a relevant document set including the top *k* retrieved documents can be obtained.

In the probabilistic IR model, ranking is calculated based on the probability theory instead of similarity measures used in the VSM. The Okapi BM25 model (Robertson, Walker, Jones, Hancock-Beaulieu, & Gatford, 1995) is another probabilistic IR model where term frequency and document length were applied to the ranking algorithm, which is more effective on the full-text document collection.

2.1.6 The Language Model (LM)

Ponte and Croft (1998) showed that the 11-point average precision of LM is better than the TF-IDF model on the TREC4 dataset. A document is usually represented by the sequences of terms in a language model. The probability that a language model generates the query is calculated given a document. Documents are ranked by the probability, P (q | M_d), where M_d is a language model giving probability estimate for a sequence of words (i.e. a query). The Language Model differs from the Probabilistic Model in that modeling the language does not need a judgment set of documents and many assumptions. The LM calculates the probability of generating a query (query likelihood) rather than the probability of predicting relevance.

The LM is not dependent on the Bag-of-Words model. Although the unigram Language Model is based on the Bag-of-Words Model, the multi-gram Language Model reflects the order of words on the matching process.

For the unigram LM, P (q | M_d) is calculated using MLE (Maximum Likelihood Estimate):

$$\hat{P}_{\text{mle}}(q \mid M_d) = \prod_{t \in q} \frac{tf_{t,d}}{L_d} \qquad (2.4)$$

where L_d is the length of a document (i.e. the number of words). Smoothing methods are applied to solve the issue with zero term frequency.

The LM provides a general scheme where language models can be variously implemented. Different LMs might be applied to several types of documents differently according to the nature of the groups. Some types may be more sensitive to bigram, while others might not. Different Smoothing and normalized methods might be applied according to collection domain / discipline / document nature (e.g. web document / academic paper), query nature (e.g. query length), etc.

2.2 The IR Process

Indexing, searching, and feedback are critical processes in IR. Information needs are reflected in a query via the process for query formulation. Documents are indexed and represented as indexed terms. The query is formulated at the point of IR, while document representation is completed using terms before the IR point (time). A query and a set of indexed documents are compared through the matching process to retrieve relevant documents. Pseudo relevance feedback using ranked results is effective for query reformulation to improve IR performance.

2.2.1 Indexing

The indexing process aims at representing documents and queries with meaningful terms that can summarize users' information needs as well as content (Sy et al., 2012). Indexing comprises three steps: tokenization, normalization, and building an index table. Tokenization is a process to extract meaningful terms from documents by dividing text into token, such as n-gram terms, punctuations, multi-word lexemes (e.g. phrases). Normalization standardizes the form of a term by case-folding, stemming, and removing stop words. Tokenization and normalization processes are not different among IR models if the unit of analysis is the same, while there is a difference between unigram and multi-gram IR models.

Document representation affects the indexing process in terms of storage size. An indexed file is created and stored in a file (generally on a hard disk). Most IR models construct the inverted indexing structure to save storage.

In the Boolean model, queries are represented by Boolean statements including Boolean operators and documents are represented by term incidence vectors whose values are 0 or 1. In other models, a query is perceived as a document, which is represented as a term vector. The values of the term vector, generally, have weighted values (e.g. TF-IDF) and are normalized by document length or other smoothing values.

Vector Space/Probabilistic/Language models need more dynamic memory space for information for the similarity/probability calculation than the Boolean model. In the Boolean model, storing each value (0 and 1) for the term incidence table needs only one bit. The other models store more information for term weightings, such as IDF, TF, and CF, which can be more efficiently stored in the inverted file by ignoring meaningless values (0s), compared with the term incidence vector table (Witten, Moffat, & Bell, 1994). Additional information might be stored according to the IR model: 1) the probabilistic model—relevance value, and 2) the LM—mean term frequency, mean probability of a term in documents containing the term, the probability that the term t is generated by the language model regarding the document, P (t | M_d), etc.

2.2.2 Search

Search is the core process of IR, which must match a query to relevant documents effectively. Salton (1988) pointed out that Boolean operations can produce unreasonable results. For example, in an OR operation, it is impossible to prioritize documents including all terms to documents only including a term. In an AND operation, documents containing all terms except one are assessed as the non-relevant documents like the documents not containing all terms. Closely matching documents, which do not include all query terms, but include several terms, cannot be retrieved in Boolean operations. In other models, a ranking is available for the retrieved results. In the VSM, each retrieved item has a similarity score. The odds ratio of the probability in the probabilistic model and the probability of query generation in the LM can be used as a ranking score. The ranking score enables best (i.e. partial) matches instead of the exact match in the Boolean model.

The Boolean model might be appropriate for IR systems not requiring a ranking result but time-sensitive searches, while other IR models with ranking might be deployed for inference searches (best matching).

2.2.3 Feedback (query reformulation)

In IR models with ranking, search results make query reformulation possible with the feedback process. Query reformulation/expansion is an intermediate process to improve the quality of the search results. Pseudo Relevance Feedback (PRF) makes IR systems dynamic using top-ranked retrieved documents. PRF automates the manual feedback of the relevance so that users obtain improved retrieval results without an extended interaction.

2.3 Comparison of Classic IR Models

To sum up, classic IR models were compared in terms of document/query representation, IR process, strength/weakness, and application (Table 1).

Table 1. Comparison of four classic IR models

Class IR Model	Boolean (1970’s)	Vector space (1970’s)	Probabilistic (1970’s)	Language (1990’s)
Doc./query Representation	term-document incidence matrix or inverted index matrix (Bag-of-Words model): 0, 1	vector notation of unigram term vector (Bag-of-Words model) term weighting	n-gram notation (unigram and multi-grams) based on term weighting	n-gram notation term weighting
Indexing/ preprocessing (tokenization, normalization)/ storage	terms, no frequency, no position, bit - binary value	term frequency, weighted value (non-binary), document length or other smoothing values, TF, CF, IDF	relevance judgment value, n-gram features, single/phrase term indexing, weighted value, document length, smoothing values	position, proximity, mean term frequency, the mean probability of a term in documents containing the term, document length, smoothing values, the probability that the term *t* is generated by the language model of document
Search/ Ranking	binary operation, exact match, deterministic	best or partial match, ad hoc operation, vector operation based on weighting model e.g. TF-IDF, geometric/distance measures: cosine similarity, Euclidian/Jaccard/dice distance, length normalization rather than probability	the Probability Ranking Principle value, inductive approach, comparison of the probability that a relevant/non-relevant document is retrieved based on the Bayesian rule—the odds of the probability, retrieval status values (RSV)	the probability that a language model for a document generates the query, smoothing/normalized methods, query/document likelihood (e.g. KL-Divergence), likelihood ratio, Divergence of query and document models
Strengths	low cost for the IR system design, fast retrieval based on simple algorithms, familiar operations with DBMS	popular search engine model with good performance, practical (easy to implement), term-weighting improves the quality of the retrieval result set,	good theoretical background - modeling the uncertainty in the IR process, the explicit assumption, the flexibility of combining with other statistic algorithms, PRF	generative probabilistic model, conceptually simple and explanatory, formal mathematical model, flexible in developing IR system—e.g.

		partial matching, the similarity measurement is relatively simple and fast, ranking by weighting models, PRF, application of text classification based on document similarity		different language models for different types of IR/documents, or disciplines, PRF, the flexibility of combining with other statistic algorithms (Dirichlet smoothing methods, EM: expectation-maximization algorithm)
Limitations	no notion of partial matching, no weighting of terms, information need must be translated into a Boolean expression, too few or too many results, not suitable for complex queries, difficult to rank results, difficult proximity/n-gram/semantic/concept-based search, memory waste by sparse values, not proper for natural language, no feedback mechanism	do not consider term dependency—difficult for proximity/n-gram/semantic search, lack of statistical foundation	judgment sets of relevance needed, arguments about initial values (estimation) of the probability of the term t appearing in a document relevant to the query in practice (without a judgment set): e.g. 0.5 or $(1/3 + (2 \cdot \mathrm{df}_t) / (3 \cdot N))$	the high cost of indexing/pre-processing in terms of storage and processing, complex algorithms, how to develop a language model, large storage needed
Application	OPAC, DBMS, a partial-match system: the set-based, extended Boolean, Fuzzy set	SMART, Generalized Vector Model	Network Inference Model, BIM (Binary Independence Model), Okapi BM25, Neural Network models	Latent Semantic Indexing

2.4 Other IR Models and Variations

There are several IR models not introduced. In cluster-based IR (Jardine & Rijsbergen, 1971, as cited in Liu, & Croft, 2004), a query is compared with the clusters of documents rather than individual documents under the assumption that similar documents would match a query. In the Network Inference Model for IR (Turtle & Croft, 1989), the inference network detects the probabilistic dependencies between nodes included in two kinds of networks. The networks consist of a query network and a document network. A document is mapped to terms and the terms are mapped to concepts existing in a thesaurus before IR. A query network is activated at the IR point. A query term is connected to some concepts so that documents related to the concepts can be retrieved. As a variation, tree-structured dependencies between terms have been applied in the probabilistic model proposed by Rijsbergen (1979, as cited in Manning et al., 2008, p. 221). This assumption of the dependency between terms contrasts with the basic assumption of the Bag-of-Words model.

2.5 Advanced IR Models

Classic models have given inspirations in developing modern IR models such as the Set-Based model, the extended Boolean model, the Fuzzy Set Model, the Generalized Vector Model, Latent Semantic Indexing, and the Neural Network model. Classic models had appeared before the Internet/Web era. Although classic models were not designed for web IR systems, they have been applied to web IR engines through integration with modern IR models and other techniques, such as machine learning and data mining.

2.6 Ensemble IR Models

Ensemble models have shown secure performance based on multiple diverse models that are implemented on different methods/parameters/algorithms/techniques. In IR based on classification or ranking, each model has its own strength and weakness. Integrating individual models can enhance the strength and complement the weakness to design a general system that would show stable performance.

Tuarob, Tucker, Salathe, and Ram (2014) proposed ensemble heterogeneous classifiers outperforming an individual classifier in health-related information classification. Five classification algorithms were employed: Random Forest (RF), Support Vector Machine (SVM), Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Bernoulli Naïve Bayes (NB), and Multinomial Naïve Bayes (MNB). Each classifier was trained on heterogeneous types of features: N-gram terms, c-feature (compound feature based on the union of n-gram words, Figueiredo, Rocha, Couto, Salles, Gonçalves, & Meira Jr, 2011), LDA features (topic and word distribution), and sentiment features. One of the ensemble types, Weighted Probability Averaging (WPA), where the average of the probability estimates for a positive class in five individual classifiers, were used for classification, which outperformed basic classifiers on Twitter and Facebook data sets in terms of precision and F1 (F-measure) scores.

Wang, Rastegar-Mojarad, Elayavilli, Liu, and Liu (2016) introduced an ensemble model integrating 1) a Part-of-Speech based query term weighting model (POSBoW), 2) a Markov Random Field model leveraging clinical information extraction (IE-MRF), and 3) a Relevance Pseudo Feedback (RPF) model for QE. POSBoW was used to weight words by assigning trained weights to POS tags, while a query was expanded by RPF. IE-MRF generated weighted medical concepts.

An ensemble IR model incorporating an RNN (recurrent neural network) model into an IR system has been introduced by Song, Yan, Li, Zhao, and Zhang (2016). An RNN model was trained based on a dataset from various resources in public websites comprising 1,606,741 query-reply pairs. Given a query, the IR system matched related query-reply pairs from a knowledge base using the query and identified the most relevant reply by scoring the relevance using a classifier. The query and the most relevant reply were used to generate another reply by the RNN model. Of these two replies, one was selected by a ranker. Compared with the reply generated from either the IR system or the RNN model, the reply generated from the ensemble system showed better evaluation scores on a human and automatic evaluation system (bilingual evaluation understudy: BLEU, Papineni, Roukos, Ward, & Zhu, 2002).

2.7 Query Expansion in IR

QE is the process to reformulate query for finding relevant documents. A query can be expanded manually/automatically/interactively (between a user and a system, Efthimiadis, 1996). Differently from manual QE, automatic QE consists of several steps before query reformulation (Carpineto & Romano, 2012; Azad & Deepak, 2019). In automatic QE, query reformulation is the final step of QE, where unnecessary terms are removed and new terms are added. For query reformulation, meaningful terms are extracted from internal/external collections, hand-built data sources (dictionaries, thesaurus, ontologies). Of the terms, terms related to the query are selected, weighted, ranked for term selection.

QE approaches have been categorized into two types of techniques: global analysis and local analysis (Azad & Deepak, 2019). For term selection, global analysis has employed various techniques according to data type: 1) linguistic techniques including syntactic/semantic/contextual analyses on external data sources (e.g. WordNet, Miller, 1995; ConceptNet, Liu & Singh, 2004), 2) concept extraction using term clustering, co-relation analysis between terms, term feature extraction using mutual information on an internal resource (corpus), 3) query-document relationship analysis on search logs (e.g. user/query/search logs), and 4) query enrichment using semantic annotations and hyper/linked text on web-based resources (e.g. Wikipedia, anchor texts, and FAQs). Meanwhile, QE terms in the local analysis are selected from retrieved documents based on (pseudo) relevance feedback.

A concept is a group of related nouns. Not only an individual term but also a concept can help increase IR performance when it is used for QE. A concept can be extracted from corpora or retrieved documents based on data mining or machine learning techniques. A concept is a group of clustered terms, which may include not only synonyms but also adjacent (co-occurred) in term of context in the collection. Concepts can be generated by. For example, a concept might correspond to a topic in topic modeling. Generated concepts can be named by concept lexicons (e.g. LSCOM, Yanagawa, Chang, Kennedy, & Hsu, 2007). Natsev, Haubold, Tešić, Xie, and Yan (2007) expanded queries by mapping text, visual queries, and initially retrieved results to LSCOM-Lite 39 concepts. The presence of concepts related to a query was used to re-rank initial results in multimedia retrieval. Terms frequently occurred in concepts related to a query or the terms with high probability in a topic are likely to be appropriate terms for QE. Those concept/topic-based QEs has shown improvements in IR performance in terms of precision, recall, or F-measure (Chang, Ounis, & Kim, 2006; Zeng, Redd, Rindflesch, & Nebeker, 2012; Xu & Croft, 2017).

Today, QE using word embedding (Mikolov, Chen, Corrado, & Dean, 2013) has been a popular trend. Word embedding is a representation technique of words with multiple dimensions. A word can be represented with multiple features as a vector. In word selection for QE, word embedding values for words can be used to measure the similarity between words. In other words, the similarity between a word and a query consisting of words can be calculated. Kuzi, Shtok, & Kurland (2016) showed that QE using word embedding or integrating QE with pseudo-feedback outperformed using only a query in terms of MAP, p@5, and RI (Robustness Index). Diaz, Mitra, & Craswell (2016) proposed a QE model using local embeddings showing improvements in recall, comparing with QE based on a global embedding. While the global embedding was trained on a whole corpus, which local embeddings were trained on topically-constrained corpora (e.g. results retrieved by query or query-related topics).

2.8 Topicality

In the document-based assessment, information quality is measured in impersonal ways. It is related to document features including content, presentation, format/type, and information about the document (metadata). This approach is similar to the system-based IR evaluation that is focused on topics of documents rather than the user's information needs. The comparison of the topic scopes between a document and the collection might be used for measuring novelty and topicality.

Topics consisting of the mixture of words are recognized by the relationships of words that appeared in the same document. The mixture of topics for a document also can show the relationship among topics. The similarity of topic distribution among documents can help to identify relationships among documents, which was a goal of analyses using citations and hyperlinks. Citation analysis has been a popular way of identifying relationships among documents or authors. H-index is an example of citation analysis reflecting recency, which was developed by Hirsch (2005) to measure a citation impact based on the number of recent publications and the number of citations to publications. PageRank (Page, Brin, Motwani & Winograd, 1999) had been developed for IR based on the Internet. PageRank scored the degree of relationships among web documents by the number of incoming links (web pages) and the degree of importance of incoming links in the Internet environment. On the other hand, conventional citation analyses have been focused on the relationship among authors or documents. High popularity based on many relationships might imply high authority, credibility or trustworthiness.

2.8.1 Topicality and cohesiveness

Zhou and Croft (2005) showed that a quality-based retrieval model was overall effective in IR in terms of precision, MRR (Mean Reciprocal Rank), and MAP (Mean Average Precision) on WT2G, WT10G, and GOV2 collections (Hawking, 2000). As a quality metric, collection-document distances were calculated by the difference of term distributions between the collection and the document. They assumed that misspelled words, the relatively high frequency of some terms, and words of tables/lists, would be the reasons for high distance. One limitation is that the application of information quality was not effective for some topics that could be presented effectively by tables and lists. Zhu and Gauch (2000) measured cohesiveness by calculating cosine similarities between vectors (representing a webpage) and a reference ontology. The more a webpage is closely related to the top 20 topics of the reference ontology, the higher cohesiveness the webpage has. The cohesiveness metric was based on the weight distribution of topics. The IR using cohesiveness showed significant improvement in terms of precision, comparing with the baseline result (paired-samples t-test, alpha = 0.05) in finding relevant sites routing a query to the sites that potentially seemed to have answers (distributed search) as well as in the centralized search (direct search). There might be challenges according to different domains: 1) how to process the large set of data, 2) how to select appropriate ontologies, and 3) how to justify the cut-off of ranked topics. Comparing topics for a document/collection/site/ontology can be a good approach to measure the comprehensiveness of a topic.

2.8.2 Topicality and relevance

A topic of the document is related to the user preference/interest. IR systems match information needs to relevant documents. Kagolovsky and Mohr (2001) discussed various perspectives of relevance. In the system-based IR, relevance has been discussed on the relationship of topics between a document and a query. Topic relevance was the main concern of system-based IR without considering users. User-based approaches have been studied in different ways. Utility was specified by Cooper (1973) as a concept to evaluate relevance from the user perspective through the search process. During the search process, a user's knowledge status and information needs are changed by the interactions between the user and documents. Utility (or system-utility) was measured by the average of search-utilities. The search-utility totals the document-utility for each document. The usefulness of documents is an important criterion for relevance in the user perspective.

Relevance has been elaborated by interactions between various factors in Saracevic's stratified model (1976; 2007) of relevance interactions. Of computer levels, the algorithmic level was involved in the IR system (specifically, query-and-system-based IR) or algorithms that match the document to a query. Topicality was the main criterion of relevance in the algorithmic level. The topic of a document must correspond to the topic of a query. Meanwhile, documents are examined at the content level in terms of accuracy, correctness, and completeness. The relationship between a document and a user's knowledge status, context, a user's emotional factors are measured in the cognitive, situational, and affective levels.

2.9 Machine Learning Technologies for Data/Text Mining and IR

Data mining is the science of discovering, extracting, and re-creating meaningful knowledge from a huge number of datasets. In the Internet world, the number of documents on the Web is growing exponentially. It became necessary to organize the knowledge from the massive data automatically using machines instead of humans. Text mining looks like a sub area of data mining in case those data are limited to text. However, the text is the main means for communication, not just a subtype of data. Text mining has been developed to some extent separately from data mining. Text mining is not a simple extension of data mining because the text is a complicated type of data, which has been a popular context or topic in several multidisciplinary areas, such as natural language processing, Artificial Intelligence (AI), IR, as well as data mining and machine learning. Topic modeling is used in text mining for discovering hidden topics. Machine learning is a subfield of computer science, which provides powerful algorithms for data mining. Clustering and classification have been popular machine learning techniques applied in IR. LDA is one technique of unsupervised learning and text mining for topic modeling.

2.9.1 Data Mining

Data mining was defined as knowledge discovery in databases (Christopher, 2010) and discovering models for data (Leskovec, Rajaraman, & Ullman, 2011). Han, Pei, and Kamber (2011) introduced seven processing steps of knowledge discovery: data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation, and knowledge presentation (visualization). Data mining was defined as a step to discover knowledge, “an essential process where intelligent methods are applied to extract data patterns” (p. 8). The first four steps are pre-processing steps that are related to data mining, while the last three steps might be seen as the preparation stages for the interaction between users and a knowledge base.

Fayyad, Piatetsky-Shapiro, and Smyth (1996) divided data mining methods into six categories: classification, regression, clustering, summarization, dependency modeling (association rule), and change/deviation detection. These are helpful techniques that can be applied to design an IR system. For example, summarization might be helpful for indexing, while classification, regression, clustering, and dependency modeling may be used for browsing. Change/deviation detection techniques can filter or rank retrieved results.

2.9.2 Text (Data) Mining (TDM)

Text mining algorithms operate on features to represent a document. The design of feature selection is a critical process that affects the performance of the IR algorithm. Four types of features are generally employed: characters, words, terms, and concepts by the order of semantic richness (Feldman & Sanger, 2007). For the character or word – level features, the order of features might be important in n-gram features, but not in bag-of-features (e.g. bag-of-characters or bag-of-words). Those features can be directly extracted from the text of the document, while the term or concept-level features are extracted from external resources, such as other domain knowledge, ontology, taxonomy, and thesaurus. Extracted features are used in identifying each document.

In the case of full-text documents, the text is unstructured because the language is natural. Like data pre-processing is needed before data mining, natural language text needs to be pre-processed before text mining using natural language processing techniques. TDM includes sentiment analysis, part-of-speech tagging, parsing (grammatical analysis), topic segmentation and recognition, automatic summarization, and so on.

There is confusion about the boundary between data mining and text mining. Both are also associated with the other areas, such as IR, AI (Artificial Intelligence), knowledge discovery, co-citation analysis, and so on. Kroeze, Matthee, and Bothma, (2003) summarized the types of text mining according to the novelty of investigation. Non-novel investigation means the retrieval of existing metadata or full-text data. Semi-novel investigation is to discover standard knowledge of existing patterns in the data, but yet unknown explicitly (e.g. co-citation analysis, summarization, lexical or syntactic analysis based on computational linguistics, discovery or classification of themes or trends). Novel investigation is the intelligent creation of new knowledge about something outside the data or collection. In text mining, AI-based on machine learning techniques is the main area to implement an intelligent system predicting the trend and influence of the data on society or other fields. Topic modeling may be a type of semi-novel/novel investigation in TDM.

Data mining and text mining adopt many techniques in machine learning and natural language processing. Those techniques show remarkable effectiveness based on the huge size of data. The human-like intelligent IR system is providing system-initiative assistant services, such as information recommendation systems.

2.9.3 Machine learning

Machine learning has evolved from the study of pattern recognition and the computational learning theory in AI. The term, Machine Learning, was originated from Samuel's paper (1959), which was defined as *"Field of study that gives computers the ability to learn without being explicitly programmed".* Machine learning is oriented on the prediction, while the focus of data mining is discovering. In machine learning, a machine is learning independently by itself based on examining a given dataset automatically. A machine is programmed by data (not by humans).

There are three main types of machine learning: 1) supervised learning, 2) unsupervised learning, and 3) reinforcement learning. In the IR perspective, (un)supervised learning aims to design effective clustering and classification schemes based on sizable training data: topic generation, spam filtering, topic categorization/spotting/segmentation/recognition, etc. The goal of reinforcement learning is to take action hoping to get the most reward (i.e. maximize a reward function), while (un)supervised learning algorithms minimize a loss (error) function for prediction. In robotics, reinforcement learning is applied to design a robot's optimal behaviors and actions. Topic modeling based on LDA is an unsupervised learning technique. In clustering, one cluster groups a set of documents, while in the LDA topic model, a topic is represented by a distribution of words.

Supervised learning. Supervised learning focuses on finding effective rules that assign the most desirable pre-defined (labeled) output for an input through learning based on a dataset (training data). Classification is a representative type of supervised learning.

In supervised learning, pre-defined categories are used in classifying documents. Documents in training data have class labels. New test documents are labeled by a classifier that is optimized by the training process. There are many applications of supervised learning; classification of news articles/emails/web pages/journal articles, spam filtering, word sense disambiguation, language/author identification, tagging, and so on.

There are a variety of algorithms for classification, such as Naive Bayes, logistic regression, decision tree, LDA (Linear Discriminant Analysis), PCA (Principal Component Analysis), SVM (Support Vector Machine), Perceptron, K nearest neighbors, Rocchio, etc. After selecting the features representing the documents, various methods for feature scoring and weighting are applied. For example, Mutual Information is used as a measurement for dependencies between variables (i.e. between a class and a term) and TF-IDF for top-ranked words can be a feature.

Several classification methods can be used and their performance might be compared. Prediction accuracy is affected by the supervised learning technique used. For instance, in a multi-layer perceptron, the neural network is trained to predict a probability distribution over the developed features, such as vocabulary weighting and Part-of-Speech tags for a text. The Perceptron based on the network structure can be designed using standard neural net training algorithms such as with backpropagation (Hagan, Demuth, Beale, & de Jesús, 1996) that includes stochastic gradient descent process for weight update between nodes. Results of other classifiers, such as SVM are compared with the results based on the artificial neural network in terms of an error rate or accuracy.

Unsupervised learning. In case there are data without labeled outputs, a hidden pattern or structure of the data can be found through an unsupervised learning process like clustering. There are several types of clustering: centroid-based clustering, hierarchical clustering, distribution-based clustering, density-based clustering, model-based clustering (decision tree & neural networks), grid-based clustering, and so on.

In the centroid-based clustering, clusters are represented by a central vector, which may be a mean value (K-means), median value (K-median), or central member (K-medoids) of the data. K-means algorithm was proposed by MacQueen (1967). K clusters must be as apart as possible. Each item is included in the nearest centroid. Centroids are recalculated until the centroids are no longer changed.

The assumption of the hierarchical clustering is that an entity is more related to nearby entities. The goal of the hierarchical clustering is to create a hierarchical tree among entities or clusters (e.g. documents or words). In the Hierarchical Agglomerative Clustering (HAC), two close clusters are merged repeatedly until there is only one cluster. The hierarchy is a form of a binary tree. The dendrogram is a tree diagram frequently used to illustrate the structure of the clusters. Three types of common algorithms are single-link clustering based on the minimum of object distances, complete-link clustering based on the maximum of object distances, and average-link clustering.

2.10 Topic Modeling

Topic clustering is one of the main interests in IR. A topic model is a statistical model to find abstract topics from a set of documents. Identification of the topics related to a query (or a tag) might be helpful for query reformulation or clustering of the related documents. The Topic model is focused on the identification of related topics by calculating the probability distributions over words, while a classical clustering algorithm (like K-means or hierarchical clustering) matches only one label per document rather than multiple matches.

2.10.1 Latent Semantic Analysis / Indexing (LSA, LSI)

Deerwester, Dumais, Furnas, Landauer, and Harshman (1990) developed an algorithm to find latent topics. A term-document count matrix is created and analyzed by Singular Value Decomposition (SVD). SVD is used to identify strong relationships between terms and documents with the least information.

The relationships of documents and terms are represented in a latent semantic space so that cluster related documents and words, which can be used in IR. As limitations, it is hard to determine the number of topics in LSA (Alghamdi & Alfalqi, 2015) and it does not allow for polysemy, implying that words with multiple meanings cannot match different topics (Bergamaschi, Po, & Sorrentino, 2014).

2.10.2 Probabilistic Latent Semantic Analysis / Indexing (PLSA, PLSI)

Hofmann (1999) introduced an automatic document indexing model based on EM (Expectation-Maximization) algorithm and KL (Kullback Leibler) Projection. Conceptual similarity and difference to LSA/LDI were discussed: Mixture Decomposition vs. Singular Value Decomposition and Kullback Leibler Projection vs. Orthogonal Projection. PLSA shows a similar process to LDA based on the variational Bayesian inference. The disadvantages of PLSA including an overfitting issue depending on a training set (Blei, Ng, & Jordan, 2003) and slow convergence when the corpus is large (Griffiths & Steyvers, 2004).

2.10.3 Latent Dirichlet Allocation (LDA)

Blei, Ng, and Jordan (2003) developed a topic modeling method, LDA, which is one of the unsupervised learning techniques. The LDA algorithm categorizes the document into a mixed group of multiple topics. Under several topic categories, each topic word is distributed with a probability that shows how much the topic word presents the corresponding topic category. A new inference technique was introduced based on variational methods and an EM (Expectation-Maximization, Hofmann, 1999) algorithm for Bayes parameter estimation, which is an optimization approach (Figure 1).

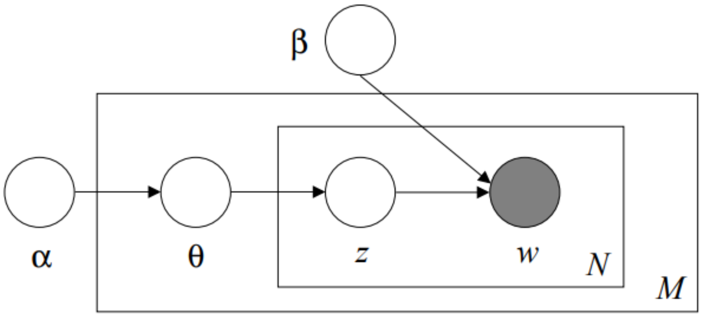

Figure １. LDA plate notation (Blei, Ng, & Jordan, 2003)

A word is the basic unit in a vocabulary indexed by {1..., *V*}. A document is a sequence of *N* words: **w** = (w_1, w_2, ..., w_n), where w_n is the *n*th word in the sequence. A corpus is a collection of *M* documents: *D* = {w_1, w_2, ..., w_m}. Dirichlet prior **α** is given for the topic distributions (θ_d) for each document *d*. For the observed words (***w***) and the number of topics (*k*), topic z_n is assigned for w_n over the multinomial variable (θ). the word probabilities are parameterized by a *k* × *V* matrix, **β**. The LDA problem is to solve the probabilities of topic-document and topic-word for a document.

$$p\ (\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{p(\boldsymbol{\theta},\ \mathbf{z},\ \mathbf{w} \mid \boldsymbol{\alpha},\ \boldsymbol{\beta})}{p(\mathbf{w} \mid \boldsymbol{\alpha},\ \boldsymbol{\beta})} \tag{2.5}$$

Two free variational parameters, φ and γ, which are used to estimate the topic-word distributions (**z**) and the topic distributions (**θ**) for each document, are updated by stochastic iteration process - minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior. Original Variational Bayes (VB) is not practical for a collection including large amounts of documents because it is based on batch processing needing all documents in a collection. Hoffman, Bach, and Blei (2010) developed the Online VB inference for LDA to overcome this weakness of the batch VB inference by just looking at parts of documents.

The Gibbs sampling, a type of MCMC (Markov Chain Monte Carlo) is another way to implement LDA models (Griffiths & Steyvers, 2004; Steyvers & Griffiths, 2007). In the initial step, the Gibbs sampling algorithm assigns each word token in a document to a random topic and updates the topic of the words using word-topic and document-topic count matrices and Dirichlet priors (**α** and **β** as hyper-parameters).

$$p\ (\mathrm{z}_i = j \mid \mathrm{z}_{-i}\ , w_i\ , d_i, \cdot\ ) \propto \varphi_i'^{\,(j)}\ \theta_j'^{\,(d)} \tag{2.6}$$

where,

$$\varphi_i'^{(j)} = \frac{C_{w_i j}^{WT} + \beta}{\sum_{w=1}^{W} C_{wj}^{WT} + W\beta} \tag{2.7}$$

$$\theta\ _j'^{(d)} = \frac{C_{d_i j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d_i t}^{DT} + T\alpha} \tag{2.8}$$

“ · ” means all other known or observed information, all other word and document indices $\boldsymbol{w}_{-i}$ and $\boldsymbol{d}_{-i}$. $\mathbf{C}^{WT}$ and $\mathbf{C}^{DT}$ are *W(ord)* x *T(opic)* and *D*(*ocument)* x *T* count matrices. C_{wj}^{WT} is the frequency of the word (*w*) assigned to the topic (*j*), not including the current word (w_i). $C_{d_i t}^{DT}$ is the frequency of the topic (*t*) assigned to the word tokens in the document (d_i), not including the current topic in the document ($d_i j$).

Krestel, Fankhauser, and Nejdl (2009) showed that the LDA achieves better accuracy in eliciting a shared topical structure from collaborative tags than the approach using TF-IDF based on the dataset from Delicious, which includes 75,000 users, 500,000 tags and 3,200,000 resources connected via 17,000,000 tag assignments of users. Wei and Croft (2006) showed that LDA application to a standard language model was effective in improving ad-hoc retrieval on several kinds of TREC collections (Associated Press newswire, Financial Times, San Jose Mercury News, and Wall Street Journal).

Goodwin and Harabagiu (2014) applied LDA in the TREC 2014 Clinical Decision Support track. The distributions of LDA topic words representing a query and a document were used to calculate the cosine similarity between two LDA topic representations for a query and a document. The IR performance was not effective, compared with other methods using medical knowledge bases (e.g. the Unified Medical Language System (UMLS), the Systemized Nomenclature of Medicine – Clinical Terms (SNOMED-CT), and Wikipedia and unsupervised distributional semantics based on Google’s Word2Vec deep learning architecture. It implies that the only LDA application integrated with no other IR algorithms might not be effective on IR performance.

Chen, Zhang, Song, and Wang (2015) showed that QE using LDA topic words outperformed a classic language model and widely used QE approaches (Lavrenko & Croft, 2001) in terms of Mean Average Precision (MAP) on the TREC AP8890 collection. Top M ranked words of top N topics (total M*N words) were used for QE.

Lu and Wolfram (2012) applied an LDA model to understand the relatedness between authors by similarity based on topic words. A Twitter-LDA model that modified the original LDA model was introduced by Zhao et al. (2011) to design a topic discovery system by classifying the topics according to news categories such as art, sports, business, etc. Joo, Choi, and Choi (2018) showed the trend of the research domain of knowledge organization based on the LDA topic model and term frequency over time.

2.10.4 LDA topic model evaluation

Existing tools, indicators, and indexes can be used or referred to decide the measurement. Reliability and validity are critical evaluation factors in justifying the selection of the measurement. Measurement should be consistent regardless of internal (e.g. instrument and tools) and external factors (e.g. time, place, and researcher). It is related to whether the measurement is valid. The validity of the measurement selection might be justified by experts or literature.

Measurement reliability refers to the repeatability of measurements. If an instrument showed different values of measurement, the measurement would be unreliable, which may generate different results and conclusions.

Validity is the degree to which a variable can be measured by measurement procedures and instruments (internal validity) or the extent to which a variable can be generalized by external factors, such as discipline, media, situation, time, space, etc. Even if the measurement were reliable, if validity is not guaranteed, the measurement of the variable is meaningless in the research.

Measurement validity is the extent to which a measure can be explained by the measurement. A measure can be explained by several indicators or indexes. A model (formula) can be generated by factor analysis or a researcher's design. If there were existing scores or classes representing a measure (dependent variable), statistics analyses (e.g. regression analysis, correlation analysis, or multi-collinearity test: the degree to which variables affect each other) can be conducted to evaluate the coefficient of each indicator (independent variable) as well as the overall performance of the new model (e.g. R-squared). Those analyses are helpful to re-design the model. Valid models might be used as indexes or tools.

2.8.4.1 The reliability of topic modeling (LDA)

Machine learning depends on the iterative inference algorithm. In the LDA model, topic reliability (consistency) is hardly guaranteed. In LDA, variables are β_k (distribution over vocabulary for topic *k*) and $\theta_{d,k}$ (topic proportion for the topic, *k*, in the document, *d*). Reliability for LDA modeling can be related to the variables affected by LDA parameters, such as Dirichlet parameters (α and β), the number of topics, and the number of iterations. Algorithms including Bayes variational inference, Gibbs sampling, and EM (Expectation-Maximization) can affect results. LDA topics are generated differently, although the difference might be slight. The iteration process based on unsupervised machine learning does not guarantee the same topics when the model is regenerated. Just giving a lot of iterations might be helpful (Wei & Croft, 2006) in the LDA topics model. Wallach, Mimno, and McCallum (2009) showed that it is effective for topic consistency to take an asymmetric Dirichlet prior as the hyperparameter for the document-topic distribution and a symmetric Dirichlet prior for the topic-word distribution in terms of the variance of information (VI) in the LDA model using Gibbs sampling. Meanwhile, Rehurek (2013) did not find much difference in terms of topic quality when he applied the asymmetric Dirichlet prior for the document-topic distribution in the LDA model based on Bayes variational inference. Showing topic consistency among LDA models with the same number of topics is another way to secure the reliability of the LDA model. Heo, Kang, Song, and Lee (2017) used Pearson correlation coefficients between topics (0.13 ~ 0.18, weakly positive) for 10 runs (generating the LDA model 10 times) to show there was consistency in generating topics. Hellinger distance (Hellinger, 1909), which is used to quantify the difference between two distributions, is another measure to compare topic distributions.

2.10.4.2 The validity of topic modeling (LDA)

There have been several discussions on the validity evaluation of the topic model. Measuring perplexity is a common way to see whether a topic model predicts well on a test set. The lower perplexity means better prediction. Blei, Ng, and Jordan (2003) used perplexity to find the best number of topics.

$$\text{perplexity}\ (\boldsymbol{D}_{test}) = \exp\left\{ -\frac{\sum_{d=1}^{M} \log\ p(\mathbf{w}_d \mid \boldsymbol{\alpha}, \boldsymbol{\beta})}{\sum_{d=1}^{M} N_d} \right\} \tag{2.9}$$

$\boldsymbol{D}_{test}$ is a test collection of documents. M is the number of documents. N_d is the number of words in the document. **α** is the Dirichlet prior for distribution over topics θ_d, and **β** is a multinomial distribution over the vocabulary, which is driven from Dirichlet prior.

Jacobi, van Atteveldt, and Welbers (2016) suggested that a formal internal validity evaluation for a topic model should be by checking automatically coded articles or by comparing it to manually coded articles. Quinn, Monroe, Colaresi, Crespin, and Radev (2010) discussed basic types of external or criterion-based concepts of validity for a topic model; 1) semantic validity (the extent to which each category or document has a coherent meaning and is related to one another in a meaningful way), 2) construct validity (convergent construct—the extent to which the new measure matches existing measures that it should match, discriminant—the extent to which the measure departs from existing dissimilar measures), predictive validity (the extent to which the measure corresponds to external events), and hypothesis validity (the extent to which the measure can be used effectively to test hypotheses). Grimmer and Stewart (2013) introduced convergent validity in order to validate the LDA topic model, which is based on unsupervised learning, using supervised methods.

2.10.5 Software / Tools for the LDA topic model

Several tools for topic modeling have been developed and updated in various computer programming languages. As variational LDA models and related models are proposed, the packages for the related models have been implemented, added and integrated into existing modules.

BigARTM (Vorontsov, 2014) has been developed based on ARTM (Additive Regularization for Topic Models, Vorontsov, 2014). ARTM was designed to overcome two limitations of LDA regarding sparsity: 1) most topics have zero probability in a document and 2) most words have zero probability in a topic (Vorontsov & Potapenko, 2014).

Mallet (McCallum, 2002) is a Java-based software package for machine learning applications, which includes topic modeling algorithms sampling-based implementations of LDA, Pachinko Allocation (PAM – including correlations between topics, Li & McCallum, 2006), and Hierarchical LDA (Griffiths, Jordan, Tenenbaum, & Blei, 2004). Stanford Topic Modeling Toolkit (Ramage & Rosen, 2009) includes several types of LDA implementations, such as collapsed Gibbs sampler (Griffiths & Steyvers, 2004) and the collapsed variational Bayes approximation to the LDA objective (Asuncion, Welling, Smyth, & Teh, 2009).

gensim ("Generate Similar", Rehurek & Sojka, 2010, http://radimrehurek.com/gensim/) is an open Python library, which includes text-preprocessing modules for generating vector space representation based on a corpus/dictionary. The LDA module was developed based on online learning for LDA (Hoffman, Bach, & Blei, 2010). It also includes a Mallet wrapper for compatibility.

Hornik, K., and Grün, B. (2011) implemented an R package consisting of the Bayesian mixture model for discrete data where topics are assumed to be uncorrelated (Blei, Ng, & Jordan, 2003), Correlated topic models (CTM, Lafferty & Blei, 2006) and Gibbs sampling (Phan, Nguyen, & Horiguchi, 2008).

jLDADMM was introduced as a Java-based package by Nguyen (2015), which was designed for topic modeling on normal or short texts using collapsed Gibbs sampling.

TopicModelsVB.jl (Proffitt, 2016) is a Julia package for the variational Bayesian topic modeling (Blei, Ng, & Jordan, 2003). It includes variations of LDA models, like filtered latent Dirichlet allocation model, Correlated Topic Model (CTM, Lafferty & Blei, 2006), filtered correlated topic model, Dynamic Topic Model (DTM, Blei, & Lafferty, 2006), and Collaborative Topic Poisson Factorization model (CTPF, Gopalan, Charlin, & Blei, 2014).

2.10.6 Topic model applications to IR

Topic modeling has been used to discover or identify topics in a collection of health information. Relationships between words or topics are a popular theme of research. Karami (2015) proposed a topic modeling for medical Corpora based on the fuzzy set theory (Zadeh, 1973) and compared its accuracy of classification to that of another topic modeling (LDA); subsets of medical abstracts, such as MuchMore Springer Bilingual Corpus (http://muchmore.dfki.de/resources1.htm) and Ohsumed Collection (www.disi.unitn.it/moschitti/corpora/ohsumed-first-20000-docs.tar.gz), were used to generate to classification models. Paul and Dredze (2014) tried to catch the health-related topics in social media using topic models based on 144 million Twitter messages using a variant of LDA, which considers whether the word is related to an ailment or just a common word. Topic words were categorized into general/symptoms/treatments words. Zhang et al. (2011) Proposed a Symptom-Herb-Diagnosis topic (SHDT) model to identify relationships among symptoms, herbs, and diagnoses, which is an extension of Author-topic model (Rosen-Zvi, Griffiths, Steyvers, & Smyth, 2004), a variation of LDA.

Topic distributions might be described to identify influential topics for IR, such as collection topic distribution. There have been many tries to weight a term for IR. Sparck Jones (1972) introduced the concept of the inverse document frequency. Salton and Yang (1973) introduced a logarithmic form for term weighting using the inverse document frequency. Zhang and Nguyen (2005) introduced a term significance weighting model by combining the frequency characteristics (the range and the middle value) with the term distribution characteristics (the width – the ratio of term frequency in the collection to that in the document and the depth – the ratio of all terms in all documents including the term to the number of terms in the collection).

To select topic words for QE, the predicted topic probability can be weighted in several ways. As a similar concept to TF-IDF, topic and term probability were introduced in previous studies regarding IR: TF-ITP (Term-Frequency · Inverse Term Probability, Ferilli, 2011), TP·ITP (Term Probability · Inverse Topic Probability, Brisebois, Abran, Nadembega, & N’techobo, 2017).

2.11 ANN (Artificial Neural Network) - based Supervised Leaning for IR

The idea for the artificial neural network (ANN) has been introduced several decades ago, being inspired by the human brain mechanism (McCulloch & Pitts, 1943). As computation power to handle a huge amount of data has been growing, ANN applications have proven the powerful prediction performance in the machine learning areas including unsupervised/reinforcement learning as well as supervised learning (e.g. regression/classification). Supervised learning has been popularly used for not only IR but also other types of information services related to IR, such as Q&A service, the recommender system, browsing, filtering, and so on. Applications of ANN-based machine learning techniques to IR have been increasing the potential by being utilized for other information services.

2.11.1 The Backpropagation Model for supervised learning

Backpropagation (Hagan, Demuth, Beale, & de Jesús, 1996) is the algorithm to update the weights between nodes included in an ANN classifier. The predicted target label and the actual label must be the same if the ANN classifier works correctly, but there would be many cases wherein the prediction is not correct. To minimize the error, Backpropagation is used to adjust weight values.

An ANN model for classification can comprise many nodes (neurons) with multiple layers. A simple artificial neuron model was illustrated in Figure 2 with two layers (input and hidden/output). The number of input (*x*) nodes, *n*, is the same as the number of input features. Lots of nodes and several hidden layers can be included in the ANN model. The number of nodes and hidden layers affects accuracy. The more nodes in the hidden layer and the more hidden layers show better accuracy for training data, generally, however, which can make an overfit meaning that the ANN model may not work well on a test set. A node/neuron (*j*) in the hidden layer is connected with all input nodes with a weight value (e.g. w_{2j} – the weight value between the 2nd input node and the *j*th node in the hidden layer).

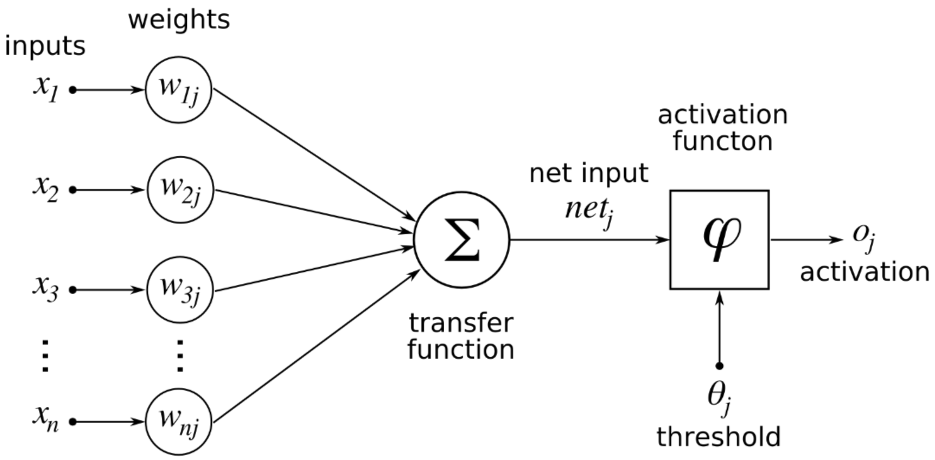

Figure 2. The artificial neuron model (Chrislb, 2005)

The transfer function generates an output value from a node in the hidden layer. All the weights between the node and input nodes are summed:

$$net_j = w_{1j} + w_{2j} + w_{3j} \ldots w_{nj} \tag{2.10}$$

Activation functions reside in the nodes in hidden layers and the output layer. The summation value is transformed into a value for the input of the next layer (if it exists) or the output layer through the activation function. The Relu (Rectified linear units) function is used widely for the hidden layer. Relu showed better performance comparing with binary units in face recognition (Nair & Hinton, 2010). Sigmoid functions are used for the output layer in the binary classification, which can be multi-labeled classification. Meanwhile, the softmax function, which is formalized by Gibbs (1902), can be employed in not only binary but also multiclass classification. The softmax function (σ), normalized exponential function, generates a probability to be included in the class (i) for each element z_i , where i = 1, 2… , K and $\mathbf{z} = (z_1, z_2, \ldots, z_K) \in$ R^{K} :

$$\sigma(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.11}$$

Probabilities might be used for more sophisticated weighting rather than using class values.

A loss (cost) function (***Loss***) is required to update weighting values by minimizing the loss so that the ANN classifier predicts classes accurately. The cross-entropy (Good, 1956) function is popularly used as a loss function in an ANN classifier, while the square error is a general loss function in a regression model. The cross-entropy is calculated using discrete probability distributions (p and q), for a random variable with a set of possibilities, $\{x_{1, \ldots}, x_{\mathrm{n}}\}$ in *X*:

$$H\,(p, q) = -\sum_{x\,\in\,X} p(x) \log q(x) \tag{2.12}$$

In backpropagation the partial derivative of the loss function concerning a weight, w_{ij} is computed: $\frac{\partial \boldsymbol{Loss}}{\partial w}$. By being multiplied by a learning rate, η, the weight is updated:

$$\Delta w = \eta \frac{\partial \boldsymbol{Loss}}{\partial w} \tag{2.13}$$

2.11.2 ANN-based supervised learning applications in IR

As data size is growing as much as we cannot manually control, automatic classification and information retrieval based on topics related to a query are getting attention. The ANN is widely employed to design effective IR systems, which are trained on big data. In image recognition, CNN (Convolutional Neural Network)-based systems have outperformed humans. Falagas et al. (2017) showed that the accuracy of a CNN learning system was better than the average of 21 demonologists in the classification of skin cancers for photographic and dermoscopic images.

ANN or DNN (deep neural network) has shown excellent performance in the prediction of both classification and regression. ANN classifiers can be used in not only classification but also a scoring/weighting function for IR. In text IR, word features play a critical role to decide the performance of classification/regression. High dimensional features for a word can be are generated by ANN classifiers using context related to the word. Those features showed a powerful potential in measuring similarity among words and documents as well as classification. Hughes, Li, Kotoulas, and Suzumura (2017) designed a deep learning network based on a CNN to categorize medical text over 26 categories. The (Word2Vec +

CNN) model showed better accuracy (0.68) than the (Doc2Vec + logistic regression classifier) model (0.28) in the categorization of the medical text.

Huang et al. (2013) applied a DNN model to create low-dimensional semantic features from a document. The dimensions for semantic (i.e. concept) features were created by an LSA (Latent Semantic Analysis) model. A term vector for a document/query comprised word frequencies for a 500K-word vocabulary. Term vectors were transformed into letter-trigram vectors using a word hashing method. A word was represented by a vector with 30,621 dimensions. The letter-trigram vectors were inputted to generate semantic features with 128 dimensions as outputs using an ANN classifier. Candidate documents for a query were ranked by measuring cosine similarity of the semantic features between a query and candidate documents. The integrated ranking model by the DNN and LSA models using the word hashing method showed better nDCG scores comparing with other ranking models, such as TF-IDF, BM25, LSA, PLSA, and so on. The DNN model also showed better performance in classifying queries (e.g. Restaurant/ /Hotel/Flight/Nightlife) than SVM (Liu, Gao, He, Deng, Duh, & Wang, 2015).

Yan, Song, and Wu (2016) introduced a DNN-based scoring system, which is a responding system selecting a relevant reply for a query (question). Candidate replies were extracted from existing Q&A web data by a scoring system. The scoring system employed a DNN model including word embedding, LSTM (long short-term memory), CNN and multiple fully connected neural network layers. The texts of an original query, a reformulated query including context, an antecedent posting, and a candidate reply were inputted to generate three scores: 1) the relatedness between a reply and the reformulated query, 2) the similarity between the associated posting and the query, 3) the correlation of the reformulated query and the original query. The final score of the candidate reply is weighted by three scores for ranking. The Deep Learning-to-Respond model outperformed other models, such as Random Match, Okapi BM25, Deep Match (Lu & Li, 2013) based on LDA topics in terms of nDCG and MAP.

Word embeddings (Mikolov, Chen, Corrado, & Dean, 2013) are vector representations (Word2Vec) for a word, which is created by an ANN classifier. High dimensional features are generated for a word based on the relationship between the word and context (i.e. other words around the word). Le and Mikolov (2014) proposed a sentence embedding model based on paragraph/document vector (Doc2Vec), which is constructed similarly to Word2Vec. Zuccon, Koopman, Bruza, and Azzopardi (2015) introduced a neural translation language model wherein the relevance between a document and a query was measured by the cosine similarity of the words in a query and a document. The neural translation language model showed better MAP and P@10 scores in several datasets including newswire articles and Wikipedia articles, comparing with the Dirichlet Language Model and the Translation Language Model (Karimzadehgan & Zhai, 2010).

2.12 Health Information & IR

Health information covers several areas, such as general health information for patients, drugs and supplements, health information for specific populations, genetics, environmental health & toxicology, clinical trials, biomedical literature, and so on (NIH, 2018). Although health information is getting more accessible to the public, health terminology is a change for consumers in IR.

2.12.1 Data retrieval

Data and IR systems are affected by the nature of disciplines. Health information includes bibliographic contents based on the Database Management System (DBMS), web catalogs, and specialized registries including the combination of heterogeneous information (Lopes, 2008). Those data consist of structured/semi-structured data. Data in public health information systems are sets of individual health records and administrative records of health institutions (e.g. police records of accidents or violent deaths, occupational reports of work-related injuries, and food/agricultural records of food production and distribution). Those data are generated from the public health practice (World Health Organization, 2008). PubMed (Public/Publisher MEDLINE) is a public search engine consisting of MEDLINE (Medical

Literature Analysis and Retrieval System Online) data. MEDLINE is a type of OPAC (Online Public Access Catalog) based on databases, whose data type is a structured form of metadata including references and abstracts. For structured data, clarity and conciseness are critical factors in organizing data with a limited length. DBMS-based operations look efficient on a (data) retrieved system for the structured data. However, in case the record includes a long text abstract, a retrieval system needs to adopt more complicated algorithms.

2.12.2 Information retrieval

Health information can contain full-text contents (included in a collection) or news. For instance, PubMed Central data, which are provided by the US National Library of Medicine National Institutes of Health, has been widely used in research fields. PMC (PubMed Central) data consists of full-text-based biomedical literature. Text REtrieval Conference (TREC) is a conventional text retrieval conference. PMC snapshots have been employed for TREC. Typical health information systems including PubMed are implemented based on the integration of two types of information (bibliography and full-text articles) to provide enough information for consumer information needs. Also, those systems encourage user participation, through comment/review. Provision of metadata along with original articles helps users' information search.

2.12.3 Thesaurus & ontology

External information sources have been used to improve the performance in health IR. External sources, such as thesauri (e.g. the Medical Subject Headings, MeSH), and ontologies (e.g. GALEN: Generalized Architecture for Languages, Encyclopedias, and Nomenclatures in medicine), increase understandability of information by providing concepts related to a term. Those concepts are used to improve IR performance. For example, the Unified Medical Language System (UMLS) integrating external sources have been popularly used in health IR research such as TREC (Bedrick, Edinger, Cohen, & Hersh, 2012; Leaman, Khare, & Lu, 2013). MeSH is the controlled vocabulary used to index the MEDLINE

(Medical Literature Analysis and Retrieval System Online) articles. Mu, Lu, and Ryu, (2014) showed that the IR system including a tree browser and a term browser based on MeSH was effective to improve user-perceived topic familiarity and Q&A performance. SNOMED-CT (Systematized Nomenclature of Medicine - Clinical Terms, (https://searchhealthit.techtarget.com/definition/SNOMED-CT**)** is another standard vocabulary including clinical terms based on 300,000 medical concepts for clinical health information regarding medical symptoms and conditions**.** SNOMED-CT terms have been mapped into MeSH terms through UMLS (Merabti, Letord, Abdoune, Lecroq, Joubert, & Darmoni, 2009), which enable more extensive related terminology for IR.

2.12.4 Datasets in IR

Much quantitative research is conducted for IR. System-based IR research requires collections while many classification algorithms based on machine learning for IR, prefer a large amount of training data (classified data). Although various corpora have been introduced for research, there might be concerns about how well a corpus is sampled to represent the whole of documents (population) over research interests, such as topics, disciplines, time, and regions.

Choudhury, Lin, Sundaram, Candan, Xie, and Kelliher (2010) introduced a Twitter sampling method of data for research, which has been collected from 2006 to 2009 based on a wide range of topics (a.k.a.Choudhury dataset). The sampling size has been justified by showing what size of samples is appropriate in representing various topics relatively fairly. Sampling based on topology and user context (e.g. location and activity) showed a lower error in terms of information diffusion at the granularity of topics.

The Reuters-21578 has been used for natural language processing, text-based IR, and machine learning. Reuters newswire stories (before and after 1987) are included in the collection: 21,578 documents, 37,926 word-types, 9,603 categorizations, 3,299 training documents, and 8,676 test documents.

The OHSUMED collection consists of medical abstracts from MeSH categories of the year 1991: 34,389 cardiovascular diseases-related abstracts out of 50,216 medical abstracts. PubMed Central (PMC) is a free full-text archive (about 4 million articles in 2016) of the biomedical and life sciences journal literature at the U.S. National Institutes of Health's National Library of Medicine (NIH/NLM). Snapshots of PMC have been deployed for the TREC CDS (Clinical Decision Support, http://www.trec-cds.org/) track for health IR tasks.

2.12.5 IR Evaluation

IR Evaluation is deeply related to a judgment set. The sample size (quantity) and validity (quality) of data are important factors in building a judgment set. It is ideal that all documents in the collection can be accessed for given queries, however, it would be impossible to have complete judgments in practice. Large amounts of data are better for generality, but the cost is expensive. Valid data sampling can be decided according to the evaluation purpose. In principle, valid data for general documents must be sampled over various areas, times, disciplines, etc., while sampling valid data for a specific type of documents in a certain area, would be limited to the characteristic of the document type or the domain area.

TREC evaluation standards have been used often in many studies. Gold standards for assessment can be set up on manually evaluated (e.g. relevant/non-relevant) documents by information professionals. For instance, in the TREC 2016 Clinical Decision Support track, the evaluation was conducted by physicians, who were either biomedical informatics students (in the Department of Medical Informatics and Clinical Epidemiology at Oregon Health & Science University) or postdoctoral fellows (at the Lister Hill National Center for Biomedical Communications at the U.S. National Library of Medicine). Three categories, such as “Definitely Relevant”, “Possibly Relevant”, or “Not Relevant” were applied for judgment (Roberts, Demner-Fushman, Voorhees, & Hersh, 2016).

Evaluation measures commonly used, such as average precision, R-precision, and precision-at cutoff *k*, are not robust to incomplete relevance judgments. *bpref* (Buckley & Voorhees, 2004) was more effective to incomplete relevance than R-precision and precision at 10 (P@10) in terms of Kendall's correlation between the system ranking evaluated by the original judgment set and the system ranking produced using the reduced judgment set. Yilmaz and Aslam (2006) proposed three evaluation measures for an incomplete judgment set: *induced* AP (Average Precision), *subcollection* AP, and *inferred* AP. Kendall's τ, linear correlation coefficient ρ, and root mean squared (RMS) error were calculated to see the changes as the judgment set is reduced. Compared with *bpref*, three evaluation measures were robust to the reduced judgment set. Similarly, *inferred* NDCG (Normalized Discounted Cumulative Gain) consistently outperformed infAP and nDCG on random judgments in terms of Kendall's τ and root mean squared (RMS) error (Yilmaz, Kanoulas, & Aslam, 2008).

Two inferred measures, including inferred AP and inferred NDCG, have become popular measures for large data collections with incomplete judgments (Bompada, Chang, Chen, Kumar, & Shenoy, 2007; Voorhees, 2014)—especially, in TREC (Lupu et al. 2011; Roberts, Simpson, Voorhees, & Hersh, 2015; Roberts, Demner-Fushman, Voorhees, Hersh, Bedrick, Lazar, & Pant, 2017).

Chapter 3 METHODOLOGY

QE models using LDA topic models and artificial neural network (ANN) classifiers were proposed. The PMC 2016 (the OA subset, 12/04/2016) snapshot including 1,451,661 documents in the public domain was used to generate topic words based on LDA models. An ANN classifier was used to weight the scores for the topic words or to select suitable words for QE. The TREC Evaluation scheme was chosen for evaluation by two IR evaluation metrics, infAP and infNDCG for 30 queries (http://www.trec-cds.org/topics2016.xml) because infAP and infNDCG are robust measures for collections with incomplete judgments (Yilmaz & Aslam, 2006; Yilmaz, Kanoulas, & Aslam, 2008).

3.1 Data Collection

Setting up a dataset is a costly process in quantitative research. The TREC CDS (Clinical Decision Support) track has provided several sets of data collections (e.g. PubMed Central) with a gold standard for a judgment set to participants for IR tasks. 108, 012 documents were assessed with three categories (“Definitely Relevant”, “Possibly Relevant”, or “Not Relevant”, Roberts, Demner-Fushman, Voorhees, & Hersh, 2016). Two categories (“Definitely/Possibly Relevant” and “Not Relevant”) were deployed for infAP, while three categories (“Definitely Relevant”, “Possibly Relevant”, and “Not Relevant”) were employed for infNDCG.

The cost of data collection and evaluation is influential in data selection. Even if data are closely associated with the research purpose, if it is very costly in collecting them, alternate data sets and other data domains can be adopted. For example, although the Web of sciences and Google Scholar includes convenient APIs for data collections, there are limitations to a normal scholar regarding the use of APIs; the number of the API usage is limited, or usage fees are required for the API use. Some websites restrict users from crawling web pages. On the other hand, PMC-related sites provide various open APIs so that public users collect data in convenient ways.

Evaluation cost occurs when the data should be assessed by assessors. A researcher should develop evaluation methods, sometimes a manual assessment is conducted. In quantitative research, the evaluation can be costly for large data. Research might be willing to collect more data because, generally, results are generalized on adequate data.

3.1.1 Dataset for indexing

Using the TREC data and evaluation scheme is an easy way to save the evaluation cost as well as the data collection cost. The 2016 CDS track dataset (http://www.trec-cds.org/2016.html) was indexed by the search engine, Terrier (http://terrier.org/). Terrier was used to generate search results. For the 2016 dataset, it includes 6,970 folders—journals, and 1,495,289 files—full-text articles (52G). The indexing was conducted before this study. Indexing a huge number of documents takes much time (e.g. a few weeks). Even though the data for indexing are slightly different from the data for LDA modeling, assuming the difference would not be critical to this study, the indexing data previously generated was used.

3.1.2 Dataset for LDA topic models

In this study, a PMC snapshot (12/04/2016, ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/) was used. There are 6966 OA (open access, https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist) journals included in PMC, which were categorized into three types: 1) full participation—depositing the complete contents of each volume and issues, starting with a specific volume and issue, 2) NIH Portfolio —depositing all NHI-funded articles, and 3) selective deposit—including a selected set of articles by publishers (NIH, 2015). The number of OA journals might be slightly different by the time when data are collected. This dataset was used for the LDA model generation. Of 1,451,661 text files, 1,451,651 (50.3 GB) documents were represented by MeSH terms to create LDA topic models. Ten documents did not include any MeSH terms, therefore, ignored.

3.2 Conceptual Framework for Query Expansion

The LDA model was employed to predict the topics of a new document because the LDA model overcomes several limitations occurring in the other topic models including LSA and PLSA, such as slow convergence and overfitting on a large amount of data, the polysemy issue and so on (Rao & Li, 2012; Bergamaschi, Po, & Sorrentino, 2014).

Some topic words generated by LDA models may be relevant for QE, but others may not. A classifier can play a role in selecting relevant words for QE. An ANN classifier has shown excellent performance comparing with other traditional classifiers, such as SVM, decision tree, logistic regression, k-means, and so on (Hughes, Li, Kotoulas, and Suzumura, 2017; Liu, Gao, He, Deng, Duh, & Wang, 2015; Ibrahim & Rusli, 2007; Hruschka, & Natter, 1999). Also, an ANN classifier can handle multiclass classification simply. When there are many features and large size of a dataset, the classification accuracy is improved by the deep level structure (using numerous nodes and layers) and various learning techniques (deep learning techniques including CNN & RNN). For these reasons, an ANN classifier was applied in this study.

3.2.1 The Bag of MeSH

For IR in health information, terms more related to health can contribute to IR performance. The National Library of Medicine publishes a controlled vocabulary thesaurus called MeSH (Medical Subject Headings). MeSH data consist of three types of data: 1) MeSH descriptor, 2) MeSH qualifier, and 3) MeSH Supplemental Concept Records (SCR). Generally, LDA topic models are constructed based on all words included in a collection. When including all the words in a document, an LDA model includes many general terms in topics. MeSH can be a more effective terminology than general terms in health IR. (Mu, Lu, & Ryu, 2014; Merabti, Letord, Abdoune, Lecroq, Joubert, & Darmoni, 2009; Díaz-Galiano, García-Cumbreras, Martín-Valdivia, Montejo-Ráez, & Ureña-López, 2007; Lu, Kim, & Wilbur, 2009). Another benefit of using MeSH is that preprocessing of documents represented by only MeSH terms is more efficient than using all words included in documents. For these reasons, a document was represented by MeSH terms, which are included in the full-text article. MeSH terms (n-gram) used in this study were extracted from the descriptor field (MH) in the 2016 MeSH descriptor file, which comprises 27,883 descriptors (https://www.nlm.nih.gov/mesh/download_mesh.html), but 24,883 MeSH terms were observed in the collection.

3.2.2 QE models using LDA models and ANN classifiers

One of the key processes to QE is finding relevant words related to a query. Queries in this study consist of one or two sentences in most cases. A query can be considered as a document in IR. A query and retrieved documents by a search engine can be used to collect candidate QE words related to the query. LDA is a popular technique to predict a topic as a concept given a document. LDA topics are generated depending on a collection. In IR on the collection, words selected by LDA would be more appropriate for QE than words chosen from other terminologies that are created based on external sources. Of candidate words, more relevant words for QE can be identified by a classifier. An ANN classifier would be a good choice to identify relevant QE words because ANN classifiers haven shown better than other classifiers in many studies.

Three types of QE models were proposed according to:

- Whether the QE model uses only an LDA model (with thresholds for topic/word probability): RQ1
- Whether the QE model integrated an ANN classifier with an LDA model: RQ2
- Whether the QE model integrated multiple ANN classifiers with multiple LDA models (ensemble QE models): RQ3

To rank topic words for QE, a basic word score (S_w) was calculated using Topic Probability (TP), Word Probability (WP), and Document Rank (DR):

$$S_w = TP * WP / (DR)^2 \tag{3.1}$$

TP indicates how much a document is related to the topic and WP shows how much a word is related to the topic, therefore, the multiplication value of TP and WP can be used to rank words related to the document in the LDA model. DR means the rank of a document retrieved by a search engine. The first-ranked document would be more relevant to the query than the second-ranked document. LDA Topic words generated by the first-ranked document would be more relevant than topic words generated by the second-ranked document. The power value, 2, was applied to adjust a weight value by document rank (Section 4.1.1).

ANN classifiers were applied to two types of QE models for RQ2 and RQ3: 1) Word Score Weighting (WSW) and Positive Word Selection (PWS). In the WSW model, an ANN classifier was employed to give weight to the original word score (TP * WP / $(DR)^2$). The original word score was weighted by the probabilities for the three groups (positive/negative/neutral): original word score * (weight for positive/negative/neutral words). Weight values were given to increase the original word score of positive words and to decrease the original word score of negative and neutral words. The power value (*pw*), 2, which showed better performance than 1, 3, and 4, were applied.

- The weight for binary ANN classifier:
  - negative words: (1 – the probability to be classified into the negative word group) ^{pw}
  - positive words: (1 + the probability to be classified into the positive word group) ^{pw}
- The weight for 3-class ANN classifier (3 layers and 700 nodes per layer:
  - negative words: (1 – the probability to be classified into the negative word group) ^{pw}
  - positive words: (1 + the probability to be classified into the positive word group) ^{pw}
  - neutral words: (1 – the probability to be classified into the negative word group)

In the PWS model, an ANN classifier was used to identify positive words, which were used for QE.

For RQ2, WSW and PWS models, where an LDA model and an ANN classifier were integrated, were applied to QE.

For the ensemble QE models (RQ3), one ANN classifier or multiple ANN classifiers were used to select the top *k* relevant words for QE, of candidate words recommended by several WSW/PWS models. The best *k* for QE is different according to QE models.

The overall steps for QE models using LDA models and ANN classifiers were illustrated in Figure 3.

1. Search result generation by the search engine, Terrier.
2. LDA topic word generation by LDA models: topic words were generated with different thresholds for topic probability (TP), word probability (WP), and TP*WP.

- The default topic probability (TP) threshold was set up as 0.01. If the topic probability of the retrieved documents is higher than 0.01 or equal to 0.01, the topic was considered as a related topic to the document. Retrieved documents have a rank. The top1 ranked document or the top 2 ranked documents were used to generated LDA topic words. Topic words are scored by (TP*WP/(document rank)^2) and weighted by an ANN classifier. Otherwise, positive (relevant) topic words for QE are selected by an ANN classifier. The top 7 or top 10 words were used for QE.
- Topic words were filtered by specific thresholds for TP (e.g. 0.08 or 0.1), WP (e.g. 0.03), or TP*WP (e.g. 0.08). The threshold values were determined by the result in an LDA model with 1700 topics in terms of infAP, infNDCG, and the ratio of the number of positive words and negative words (Section 4.2). Threshold values generating high infAP and infNDCG scores and high ratio values were preferred. Topic words were sorted by word score. The top 10 words were added to the original query and search results by this new query were evaluated in terms of infAP and infNDCG.

3. QE by a Word Score Weighting (WSW, Figure 4) or Positive Word Selection (PWS, Figure 5) model. A WSW/PWS model consists of an LDA model and an ANN classifier. In the WSW model, an ANN classifier was used to weight topic word scores. If the word is classified into the positive word group, a weight value more than 1 is given, while a weight value smaller than 1 is given for

the negative and neutral words. For more sophisticated weighing, the probabilities of being a positive/negative/neutral word, were used rather than the same values by the classification. The top 7 or top 10 words with highest scores were used for QE. In the PWS model, all or the top 7 positive words selected by the probability of being a positive word were used for QE.

4. In the ensemble QE models, candidate words were recommended by multiple WSW/PWS models. Of the candidate words, the top *k* words for QE were selected by one classifier and multiple classifiers. The details were explained in the next section.

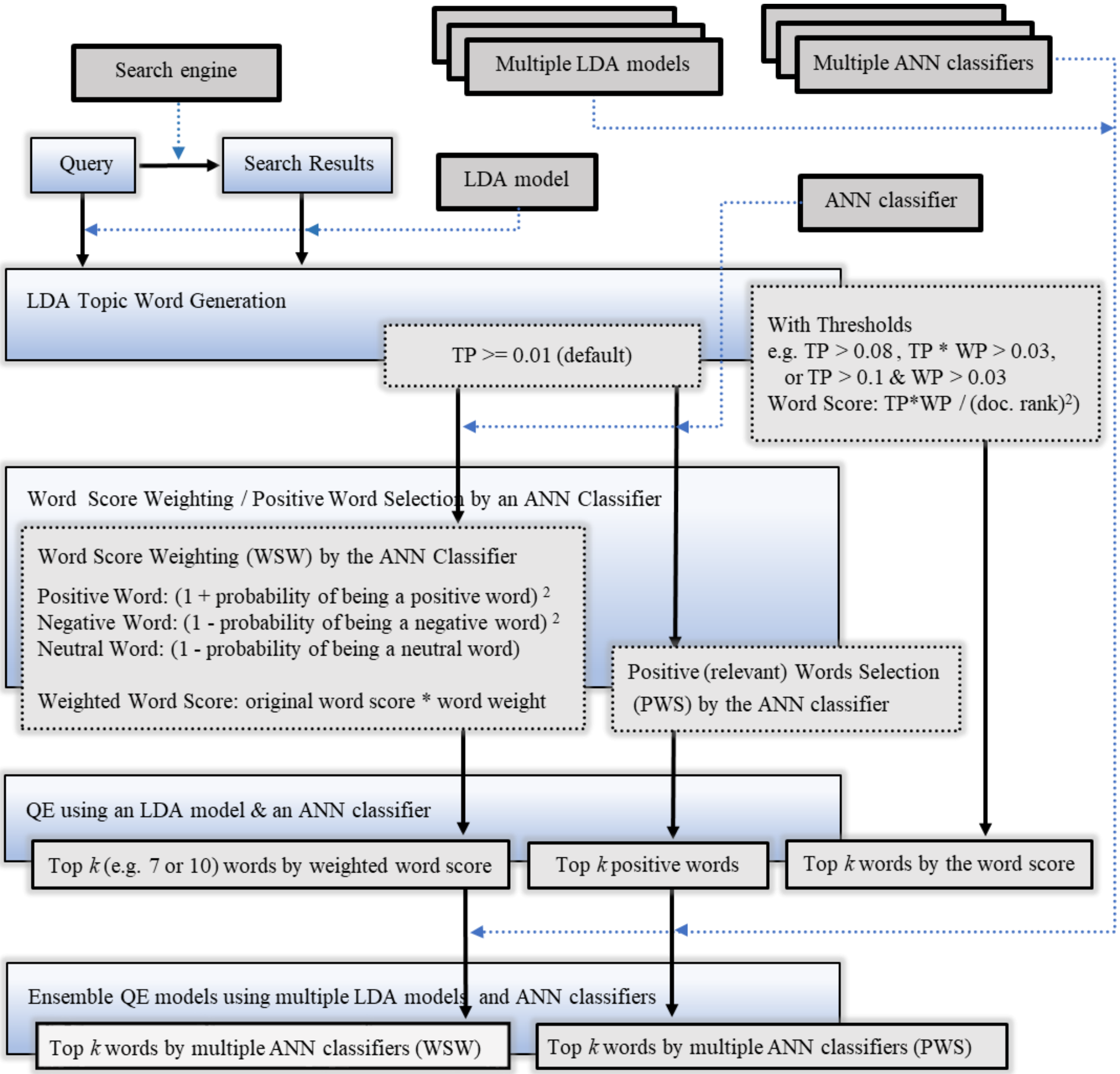
Search engine
Multiple LDA models
Multiple ANN classifiers
Query
Search Results
LDA model
ANN classifier
LDA Topic Word Generation
TP >= 0.01 (default)
With Thresholds
e.g. TP > 0.08 , TP * WP > 0.03,
or TP > 0.1 & WP > 0.03
Word Score: TP*WP / (doc. rank)²)
Word Score Weighting / Positive Word Selection by an ANN Classifier
Word Score Weighting (WSW) by the ANN Classifier
Positive Word: (1 + probability of being a positive word) ²
Negative Word: (1 - probability of being a negative word) ²
Neutral Word: (1 - probability of being a neutral word)
Weighted Word Score: original word score * word weight
Positive (relevant) Words Selection
(PWS) by the ANN classifier
QE using an LDA model & an ANN classifier
Top k (e.g. 7 or 10) words by weighted word score
Top k positive words
Top k words by the word score
Ensemble QE models using multiple LDA models and ANN classifiers
Top k words by multiple ANN classifiers (WSW)
Top k words by multiple ANN classifiers (PWS)

Figure 3. QE models using LDA models and ANN classifiers

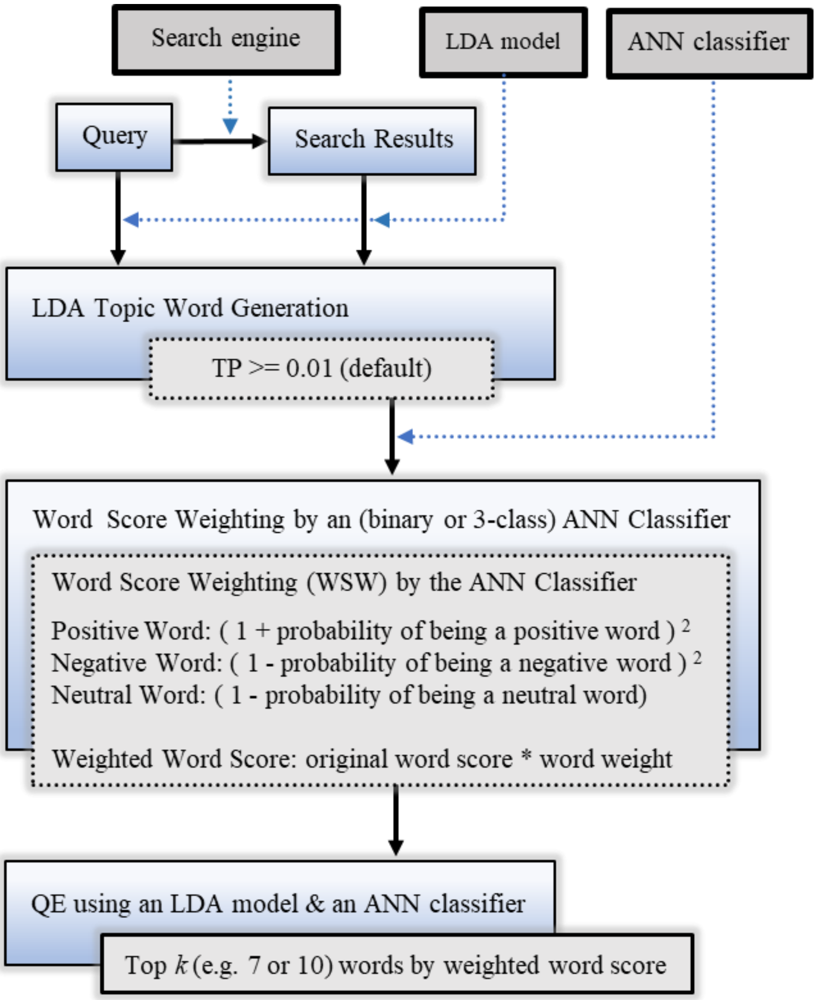
Search engine
LDA model
ANN classifier
Query
Search Results
LDA Topic Word Generation
TP >= 0.01 (default)
Word Score Weighting by an (binary or 3-class) ANN Classifier
Word Score Weighting (WSW) by the ANN Classifier
Positive Word: (1 + probability of being a positive word) 2
Negative Word: (1 - probability of being a negative word) 2
Neutral Word: (1 - probability of being a neutral word)
Weighted Word Score: original word score * word weight
QE using an LDA model & an ANN classifier
Top k (e.g. 7 or 10) words by weighted word score

Figure 4. The Word Score Weighting (WSW) model using an LDA models and an ANN classifier

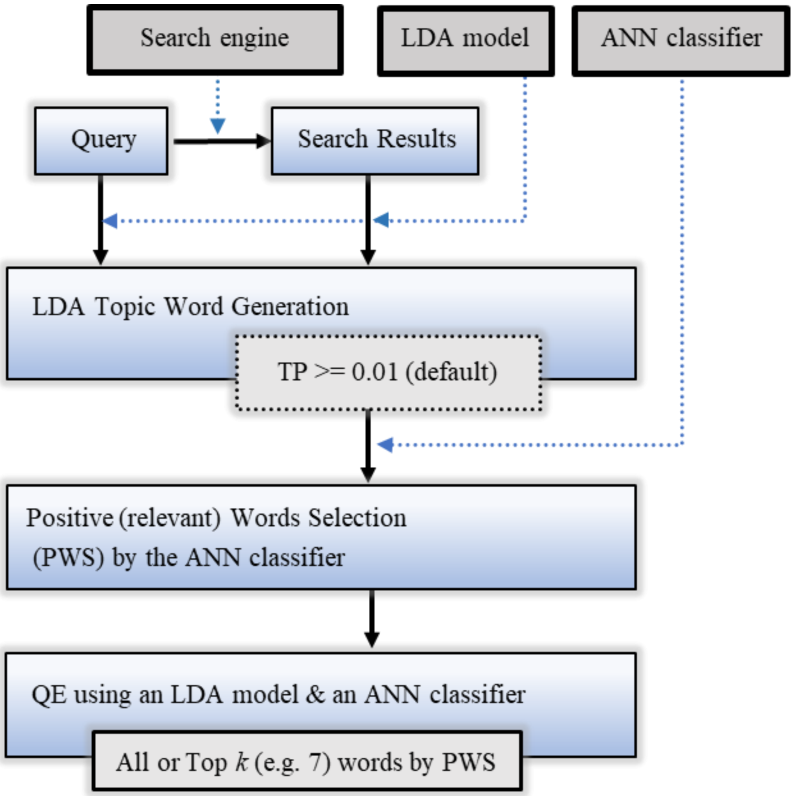
Search engine
LDA model
ANN classifier
Query
Search Results
LDA Topic Word Generation
TP >= 0.01 (default)
Positive (relevant) Words Selection
(PWS) by the ANN classifier
QE using an LDA model & an ANN classifier
All or Top *k* (e.g. 7) words by PWS

Figure 5. The Positive Word Selection (PWS) model using an LDA models and an ANN classifier

3.2.3 Ensemble QE models using multiple LDA models and ANN classifiers

Basically, ensemble QE models include multiple WSW or PWS (LDA model + one classifier) models. In addition, one classifier or multiple classifiers are used to rank candidate words that are recommended by multiple WSW or PWS models. In case that there were many duplicate candidate words (e.g. work, nature, review, etc.) with different feature values (explained in Section 3.4.1), the best feature values for ranking were considered.

3.2.3.1 Ensemble QE models based on the WSW model

Candidate words for QE are recommended by multiple WSW models. In each WSW model, topic words are sorted by the word score (TP*WP / (document rank)^2) and then weighted by an ANN classifier.

The top *k* (e.g. *k*=10) words per query from each WSW model are collected. The candidate words are ranked by one classifier or multiple classifiers as follows:

1. Topic words (e.g. the top 10 words per related topic) are generated by multiple (e.g. 20) LDA models with relatively good performance in terms of infAP or infNDCG.
2. Those words were scored by (TP * WP / (document rank)^2) in each LDA model, which are weighted using the probability estimate for positive/negative/neutral word group by an ANN classifier.
3. A maximum of the top *k* words per query (30 queries) is selected in each WSW model by the descending order of the weighted word score as candidate words. Candidate words are collected from multiple WSW models. Candidate words are ranked by one classifier or multiple classifiers.
   - When using one classifier, candidate words were ranked by the descending order of the probability for the positive word group.
   - When using multiple classifiers, the class score of a word is calculated according to the classification by each classifier: 0 for a negative word, 1 for a neutral word, and 2 for a positive word. For word ranking and filtering, 1) the sum of class scores of a word and 2) (the average of four class scores) * (the average of the probabilities for the positive word group), were calculated by multiple classifiers. Empirically, the performance of 3-class classifiers was better in the ensemble QE models than binary classifiers when using multiple classifiers. If the sum of class scores of a candidate word is less than *k* (e.g. 3), the word was removed. Remaining words were scored by (the average of the class scores) * (the average of the probabilities for the positive word group):
4. The top *k* (e.g. *k* = 1…30) words were added to the original query for QE.

3.2.3.2 Ensemble QE models based on the PWS model

Candidate words for QE are generated by multiple PWS models. In each PWS (LDA model + one ANN classifier) model, positive topic words are selected by an ANN classifier. The top *k* (e.g. *k*=15) positive words per query from multiple (e.g. 10) PWS (LDA + an ANN classifier) models, were ranked by multiple classifiers as follows.

1. Topic words were generated by multiple LDA models with relatively good performance in terms of infAP and infNDCG.
2. Those topic words were classified into two or three groups (the positive/negative/neutral word group) by an ANN classifier. Positive words are sorted by the probability estimated for the positive group.
3. A maximum of the top *k* (e.g. *k* = 15) positive words per query (30 queries) is selected in each PSW model by the descending order of the probability estimated for the positive group as candidate words for QE. Positive words are collected from multiple (e.g. 10) PWS models. Candidate words are ranked by one classifier or multiple classifiers.
   - When using one ANN classifier, positive words were ranked by the descending order of the probability for the positive word group without calculating class scores.
   - When using multiple ANN classifiers, the class score of a word was given according to the classification by each classifier: 0 for a negative word, 1 for a neutral word, and 2 for a positive word. For word ranking and filtering, 1) the sum of the class scores of a word and 2) (the average of the class scores) * (the average of the probabilities for the positive word group), are calculated by multiple classifiers. If the sum of class scores of a word was less than *k* (e.g. *k*=5), the word was ignored. Remaining positive words are scored by (the average of class scores) * (the average of the probabilities for the positive word group) values.
4. The top *k* (e.g. *k* = 1…40) words are added to the original query for QE.

The process for ensemble QE model based on the WSW/PWS model is illustrated in Figure 6.

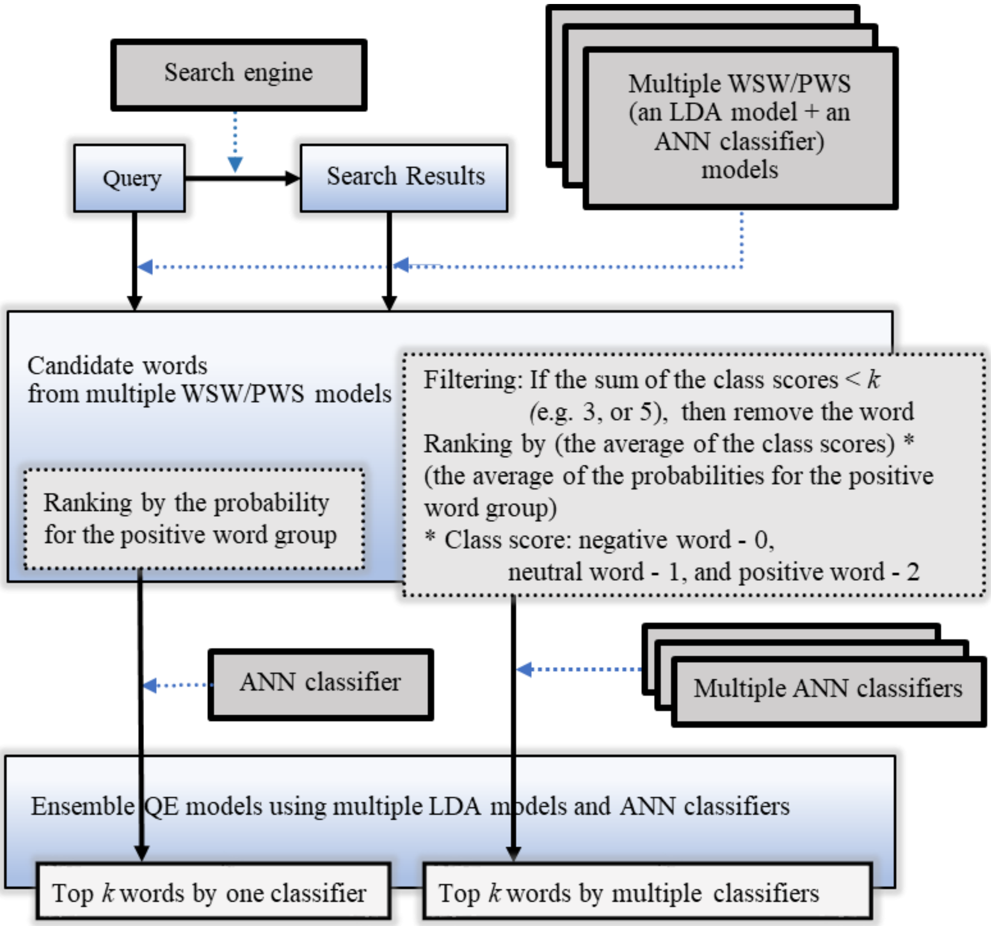
Search engine
Query
Search Results
Multiple WSW/PWS (an LDA model + an ANN classifier) models
Candidate words from multiple WSW/PWS models
Ranking by the probability for the positive word group
Filtering: If the sum of the class scores < k (e.g. 3, or 5), then remove the word
Ranking by (the average of the class scores) * (the average of the probabilities for the positive word group)
* Class score: negative word - 0, neutral word - 1, and positive word - 2
ANN classifier
Multiple ANN classifiers
Ensemble QE models using multiple LDA models and ANN classifiers
Top k words by one classifier
Top k words by multiple classifiers

Figure 6. Ensemble QE models using LDA models and ANN classifiers

3.2.4 Terminology

- A collection, **C** (or **D**), includes documents with the number of documents, N: **C** = **D** = {d_1, d_2... d_n}.
- **T** includes a group of topics with the number of topics, K: **T** = {t_1, t_2... t_k}. **TP** (d_i, **T**) is a vector including the topic probabilities for the *i*th document, d_i, for all topics, **T**: **TP** (d_i, **T**) = < tp_1, tp_2, ..., tp_k >. TP (d_i, t_j) is a scalar, which is the topic probability for the *i*th document, d_i, and the *j*th topic, t_j. The difference by the probability can be a delicate weighting factor. Compared with DF (Document Frequency), the maximum value of TP cannot exceed 1, so TP would be directly used for the calculation with normalized values.

- In this study, CTD (Collection Topic Density) and CTF were calculated for the topics, whose TP is higher than 0.01 or equal to 0.01. If the TP is lower than 0.01, CTD and CTF were set to 0. CTD (t_j) or CTD_{t_j} is a scalar representing the average topic probability for the *j*th topic, t_j, in a collection:

$$\mathrm{CTD}\ (t_j) = \mathrm{CTD}_{t_j} = \frac{\sum_{i=1,}^{N} \mathrm{TP}(d_i, t_j)}{\mathrm{N}} \tag{3.2}$$

  The term, *density*, is used to describe the probability for a collection, while *probability* is used for a document. **CTD** is a vector represented by the probability values for all topics in the collection:

$$\mathbf{CTD} = <\ \mathrm{CTD}_{t_1}, \mathrm{CTD}_{t_2}, ..., \mathrm{CTD}_{t_k}\ > \tag{3.3}$$

  When CTD is high for a topic, TP values (topic probabilities) for most documents might be relatively high regarding the topic. CTD would be very small if there are lots of topics. In that case, normalization and standardization would be useful in using the CTD values to compare other LDA models with different numbers of topics.

- **TO** (d_i) is a vector representing Topic Occurrence (0 and 1) in the *i*th document, d_i, for all topics. If the TP is lower than 0.01 (i.e. non-related topic), TO was set to 0 in this study. TO (d_i, t_j) is a scalar including the topic occurrence in the *i*th document, d_i, and the *j*th topic, t_j.

$$\mathbf{TO}\ (d_i) = <\ \mathrm{TO}\ (d_i, t_1), \mathrm{TO}\ (d_i, t_2), ..., \mathrm{TO}\ (d_i, t_k)\ > \tag{3.4}$$

- **CTF** is a vector representing the average Topic Frequency (the number of occurrences) for all topics in the collection. CTF (t_j) or CTF_{t_j} is a scalar representing the average topic probability for the *j*th topic, t_j, in the collection:

$$\mathrm{CTF}\ (t_j) = \mathrm{CTF}_{t_j} = \frac{\sum_{i=1,}^{N} \mathrm{TO}(d_i, t_j)}{\mathrm{N}} \tag{3.5}$$

  CTF was represented in vector forms as:

$$\mathbf{CTF}\ (\mathbf{T}) = < \mathrm{CTF}_{t_1}, \mathrm{CTF}_{t_2}, \ldots, \mathrm{CTF}_{t_k} > \tag{3.6}$$

Differently from CTD, CTF is based on frequency (not probability). Logarithmic normalization or standardization would be needed because CTF values would be large in case that there are a huge number of documents in a collection. CTD and CTF are similar concepts except that CTD is not a number but the average of probabilities between 0 and 1. CTF corresponds to CF (collection frequency). CTF would be used like a CF (collection frequency) if CTF is used in designing topic-weighting models.

3.3 Topic Modeling using LDA

Preprocessing was required before generating LDA models. LDA models with different numbers of topics were trained based on the 2016 PMC snapshot (Dec. 4). Before training, each document was represented by only MeSH terms that are included in the document. MeSH terms were extracted from the "MH" field (MeSH) in the 2016 MeSH descriptor file (https://www.nlm.nih.gov/mesh/download_mesh.html) so that LDA topics consist of MeSH terms. Only a complete MeSH term described in the MeSH was considered as a unit of analysis. MeSH terms are multi-gram based, which might be one word or more than one word. If a MeSH term consists of a word, the word is fine as a unit. Special characters (e.g. ",", "(", and ")") in the descriptors were ignored.

Document representation is based on the Bag of MeSH (n-gram) model. Each document was represented in the form of a pair of words and frequency. A dictionary (including MeSH terms) and a corpus (including document representations) were created for generating LDA topic models. The dictionary for LDA models includes all MeSH terms used in the collection. The average numbers of all and unique MeSH terms included in a document were 286.2 and 75.9, respectively.

Because the Python library (Rehurek & Sojka, 2010), *genism*, has been updated steadily and relatively stable and efficient to handle large size of datasets (documents), *genism* was used to create LDA topic models, which was implemented using the Variational Bayesian inference algorithm. LDA models were created with 50 iterations. 40 LDA models were generated using *genism*.

For evaluation for model fit, perplexity was measured to decide the best number of topics. With 80% of data for training and 20% of data for testing.

3.4 ANN Design

A topic word generated by LDA models for a retrieved document might be an effective word in increasing infAP and infNDCG scores or not. It is a critical process for QE to select a relevant word among the generated topic words. ANN showed good performance in supervised learning for a recent decade as the high-performance computing resources are available. ANN classifiers contribute to choosing appropriate words for QE.

3.4.1 Word features

Eight features for a word were chosen for training ANN classifiers assuming that those features would be helpful to identify relevant words for QE. Preferred features for a word would have dynamic values given a query, independently from the collection, however, static features depending on the collection might be helpful. Some features are dynamically generated by LDA models given a query and other values were calculated (saved) when the corpus was created. The features can be grouped into two types whether they are depending on an LDA model or a collection.

- LDA model - dependent features. If TP of a word is lower than 0.01 (i.e. non-related topic), the word was ignored.

  1. TP (topic probability) of a word: related topics, where TPs are higher than 0.01 or equal to 0.01, are generated by an LDA model given a text (e.g. a query or a document retrieved by the query). Also, TPs of the related topics are predicted. TP is at the document-level. Each topic has the probabilities of the words (MeSH terms) in the collection. When top *k* (e.g. *k* = 10 in this study)

topic words with highest word probabilities (WP) in a related topic were selected for QE, TPs of the words in a topic, which are generated by the same text, are the same. In case that a same word can be extracted from different topics (with different probabilities), a word can have multiple TP values, however, the same word from the different topic are considered as different words.

2. WP (word probability) in a topic related to the word, which is generated by an LDA model.
3. CTD calculated using topic probabilities for the collection.
4. CTF calculated using the numbers of topic occurrences in the collection. CTD and CTF were calculated for the topics, whose TP is higher than 0.01 or equal to 0.01. CTD and CTF are collection-level features. If words are in the same topic, CTD and CTF values are same.
5. TP * WP

- Corpus-dependent features:

6. Normalized IDF (Inverse Document Frequency):

$$\text{IDF} = \log_2 \frac{\text{N}}{DF} \tag{3.7}$$

where N is the number of documents, which was normalized by Min-Max scaling.

7. DF (Document Frequency)—the number of documents including the word in the collection
8. CF (Collection Frequency)—the frequency of the word in the collection

Five features (TP, WP, CTD, CTF, and TP*WP) are generated by LDA dynamically given a query, while the other three features including normalized IDF, DF, and CF have fixed values for a word because the features are related to the collection. TP, WP, CTD, CTF, and TP * WP depend on the LDA model, therefore, LDA models with different topic numbers generated different values for a word. Even if only one LDA model is used, the word can have multiple values for TP, WP, CTD, CTF, and TP * WP according to the topic that the word is included in. In case that a word can have different features, the word is recognized as a different sample (word). When a topic word has different feature values, only the most helpful feature values (by a ranking score for each QE model) for QE were chosen and the others were ignored. Meanwhile, a word has the same IDF, DF, and CF, depending on the corpus regardless of LDA models.

3.4.2 Data creation for an ANN classifier

To train an ANN classifier and evaluate it, training and validation datasets must be collected. The data consist of input data (features values) and output data (classification labels). Relu (Rectified linear units) and softmax functions were applied for the activation functions for the hidden layers and the output layer, respectively.

It is difficult for a researcher to decide how much data are needed for training a classifier. The more, the better, but it costs much time and effort to collect lots of data. At first, data (topic words) were generated from the top5 retrieved documents to train an ANN classifier, but the IR performance of the QE models using the ANN was not good, so, more data were generated from the top 10 retrieved documents.

For input data, the text in the title, keywords, and abstract of the top 10 ranked retrieved documents were matched to MeSH terms, and then were used in generating LDA topics. Features values were created by LDA models and a TF-IDF model, which generated by the python module, *gensim*.

For output data, the top 10 topic words, which are generated by an LDA model, were classified into three groups (positive/negative/neutral) according to whether to increase/decrease infAP and infNDCG. When the word was added in the query, if infAP or infNDCG score increases, the word was grouped into the *positive(relevant)* group, otherwise the word was classified into the *negative* group. If the word does not affect infAP and infNDCG, it was grouped into the *neutral* group. The total number of the words for three groups were counted in Table 2. The text of the top 1 retrieved document also included the query (one or two sentences in most cases) because the query can be regarded as the most important document including information need, therefore, the number of generated topics was more than normally retrieved documents (almost twice). The total number of words was 424, 288. Negative words were around three times more than positive words, while neutral words were most generated.

Table 2. The word count for the top 10 retrieved documents for three groups

	Positive	Negative	Neutral	Total
Top1 + Query	10,035	28,807	30,189	69,031
Top2	5,557	15,753	18,835	40,145
Top3	5,049	16,837	18,008	39,894
Top4	4,614	15,060	17,492	37,166
Top5	5,566	15,606	16,191	37,363
Top6	7,174	18,030	21,563	46,767
Top7	5,332	14,001	16,877	36,210
Top8	5,795	17,994	19,315	43,104
Top9	6,705	16,513	19,670	42,888
Top10	4,044	14,096	13,580	31,720
Total	59,871 (14.11%)	172,697 (40.7%)	191,720 (45.19%)	424,288

The means of the raw and standardized feature values in each group were calculated for 40 LDA models in Table 3. The mean values for TP, WP, CTD, CTF, DF, TF, TP*WP were higher in the positive group than the negative group. Meanwhile, the mean value for normalized IDF was higher in the negative group than the positive group. CTD, CTF, DF, and TF values were higher in the neutral group than the positive and negative groups. CTD, CTF, DF, and TF might be more influential features in 3-group classification than binary classification.

The details for each LDA model were described in Appendix A. To the relative comparison of feature values, feature values were standardized. Standardized value (z) for the feature value (x) was calculated using a python module, *sklearn*, as:

$$z = (x - u) / \mathbf{s} \quad (3.8)$$

where u is the mean of the feature values and s is the standard deviation.

Table 3. The means of the word feature values and p-values in the two-sample t-test for the top 10 retrieved documents in three groups (positive/negative/neutral)

	TP	WP	CTD	CTF	Norm_IDF	DF	TF	TP*WP
Raw								
Positive	**0.1713**	**0.1496**	0.00282	81768.4	0.3905	70097.8	149511.7	**0.0227**
Negative	0.1653	0.1012	0.00260	77393.9	**0.4398**	52445.1	103399.2	0.0143
Neutral	0.1702	0.1371	**0.00621**	**145368.5**	0.2738	**355446.2**	**1791520.5**	0.0202
Standardized								
Positive	**0.0288**	**0.0876**	-0.1924	-0.2123	0.1369	-0.35	-0.3444	**0.0865**
Negative	-0.0171	-0.0942	-0.1976	-0.2201	**0.3465**	-0.4039	-0.3681	-0.0815
Neutral	0.0064	0.0574	**0.2381**	**0.2646**	-0.3548	**0.4731**	**0.4392**	0.0464
p-value (t-test)								
	1.9E-31	0.0	*0.103*	0.025	0.0	9.4E-102	1.4E-239	0.0

A two-sample t-test (α = 0.05) was conducted for each feature to see the significant difference between two groups (positive words: 59,871 samples & negative words: 172,697). Except for one feature, CTD, there were significant mean differences of the standardized feature values between the positive and negative groups (Table 3). However, CTD (p-value = 0.103) was also included because CTD, actually, contributed to increasing average accuracy on the dataset including the top 5 retrieved documents. The accuracy was 0.7322 when including CTD for an ANN classifier with 2 layers with 500 nodes per layer and 1000 iterations, which was slightly higher than the average accuracy (0.7303) when not including CTD as a feature.

The p-values for WP, TP*WP, and DF were almost 0, which means that those features might be more influential in classification. It makes sense that words with high WP and TP*WP are likely to be relevant words for QE. It is interesting that the words with low normalized IDF are more in the positive word group. The words with low normalized IDF look common words in terms of the TF-IDF weighting scheme but were helpful in heath information IR when they are MeSH terms. According to the p-value, WP is more important than TP in the word classification.

3.4.3 ANN classifier evaluation

Based on 8 features for a word and three groups (positive/negative/neutral), binary and 3-class classifiers were generated. Classifiers showing good performance were applied to the QE models.

3.4.3.1 The binary classifier

A 30-cross validation test was conducted to evaluate ANN classifiers. 30 ANN classifiers were trained on 30 training datasets. The data were divided into 30 datasets for 30 queries for validation. Each training set includes 29 datasets for 29 queries. The remaining dataset was used as a validation dataset to evaluate the trained ANN classifier for the excluded query. Also, two sub-validation datasets, including only positive words and only negative words, were created, separately, for validation according to the word groups. ANN classifiers were designed with different numbers of layers and nodes. The number of nodes was selected empirically, considering the cost (calculation time) and efficiency (performance). As the number of layers increased, the accuracy for the overall validation dataset did not increase. Meanwhile, the accuracy for the training and validation datasets included in the positive word group, tended to increase. There were imbalanced classifications in most binary classifiers. Most words were classified into the negative group. One problem of imbalanced classification is that accuracy might be worse than when all samples are labeled into a group (e.g. the negative group) without classifiers. To overcome this irony, F1 and AUC (Area Under the ROC Curve) were measured. F1-score is the harmonic mean of precision and recall, while AUC is used to measure the degree of how well a model can separate samples into classes. Both metrics are referred to in imbalanced classification rather than accuracy. Considering the imbalanced number of validation data, weighted F1 scores were calculated.

Each classifier was trained with 1000 iterations and the batch size, 10000. The ANN classifier with one layer and 500 nodes per layer showed the best accuracy and AUC scores for validation data but showed low weighted F1 score. Overall classifiers with more layers and nodes showed better performance on the training set but did not guarantee better performance on the validation set (overfit). For example, the ANN classifier with 3 layers and 500 nodes per layer showed the best scores in terms of accuracy, weighted F1 and AUC for the training dataset but did show low accuracy and AUC scores on the validation dataset.

The best performance for individual LDA models was observed on the ANN classifier with 2 layers and 700 nodes, where overall accuracy, w_F1(weighted F1), and AUC scores for the validation set were good. Meanwhile, the ANN classifier with 3 layers and 500 nodes per layer would be more useful in detecting positive words. The performance of ANN classifiers would be improved on more relevant data. Table 4 listed the evaluation results for binary classifiers (best in bold and second best in italics).

Table 4. Average accuracy, F1, and AUC scores for binary ANN classifiers for 30 queries

	Acc	Acc	Acc	Acc	w_F1	w_F1	AUC	AUC
	(train)	(val_all)	(val_pos)	(val_neg)	(train)	(val_all)	(train)	(val_all)
1 layer								
500 nodes	0.7441	*0.7282*	0.0151	**0.9950**	0.6407	0.6310	0.6025	**0.5819**
1000 nodes	0.7441	**0.7287**	0.0167	*0.9948*	0.6411	0.6320	0.605	*0.5791*
2 layers								
300 nodes	0.7480	0.7261	0.0464	0.9833	0.6596	0.6410	0.6226	0.5779
400 nodes	0.7485	0.7264	0.0516	0.9828	0.6604	0.6414	0.6257	0.5775
500 nodes	0.7491	0.7246	0.0495	0.9807	0.6623	0.6405	0.6296	0.5712
700 nodes	0.7494	0.7233	0.0549	0.9779	0.6649	0.6414	0.6297	0.5772
3 layers								
200 nodes	0.7525	0.7244	0.0671	0.9744	0.6731	0.6472	0.644	0.5706
300 nodes	0.7569	0.7212	0.0761	0.9683	0.6838	0.6476	0.6603	0.5618
500 nodes	**0.8063**	0.6901	**0.1732**	0.8922	**0.78**	*0.6561*	**0.8162**	0.5342
700 nodes	0.747	0.7252	0.0393	0.9825	0.6564	0.6390	0.6175	0.5800
5 layers								
300 nodes	*0.7869*	0.705	*0.14*	0.9231	*0.7457*	**0.6573**	*0.7582*	0.5441
700 nodes	0.7820	0.7070	0.1374	0.9313	0.7335	0.6540	0.7347	0.5432

3.3.3.1 The 3-class classifier

The evaluation results for 3-class classifiers were described in Table 5. Overall accuracy, F1, and AUC scores of the 3-class classifiers were lower than the scores of the binary classifiers. Differently from the binary classifier, the neutral words were used in training, but still have an imbalanced classification problem, which barely detected positive words. The ANN classifier with 3 layers and 700 nodes per layer showed the highest weighted F1 score but the lowest AUC score for the validation set. Because the ANN classifier with 3 layers and 700 nodes per layer showed the best accuracy in detecting positive words of three classifiers, it was integrated with LDA models for QE.

Table 5. Average accuracy, F1, and AUC scores for 3-class ANN classifiers for 30 queries

	Acc (train)	Acc val_all	Acc val_pos	Acc val_neu	Acc val_neg	w_F1 (train)	w_F1 val_all	AUC (train)	AUC val_all
2 layers									
400 nodes	0.6252	0.6141	0.0335	0.5589	**0.9031**	0.5868	0.5815	**0.3223**	**0.3323**
700 nodes	**0.6301**	0.6141	0.0465	0.5644	0.8954	0.5942	0.5844	0.3147	0.3263
3 layers									
700 nodes	0.6710	0.6094	**0.0985**	**0.6119**	0.8127	**0.6514**	**0.5923**	0.2867	0.3106

3.5 IR Evaluation

TREC datasets and evaluation scheme of the TREC 2016 Clinical Decision Support (CDS) track was used. The TREC 2016 CDS track provides a snapshot of an open-access subset on March 28, 2016, for ad hoc retrieval tasks. Full-text articles were distributed in the NXML format (XML encoded using the NLM Journal Archiving and Interchange Tag Library). There are 30 queries (called topic) given in the CDS track. 30 queries were used for LDA topic generation along with retrieved documents. The text for original queries was integrated with the text for the top1 retrieved (ranked) document.

LDA models and ANN classifiers were used to expand the queries. LDA top *n* topic words for the top *k* retrieved documents were added to the original query for a baseline run. In addition to the baseline run for the original queries, several runs based on QE models using LDA models and ANN classifiers, were generated:

- Queries for the baseline run. 30 texts in the *summary* fields of original (query) topics (http://www.trec-cds.org/topics2016.xml) have been used as queries. The baseline run was created by the search engine using the original query without QE. 1000 search results per query were included in the baseline run. The search algorithm is based on the Language Model using Bayesian smoothing with Dirichlet Prior (Zhai & Lafferty, 2004). Porter stemmer was set up as the default for the retrieval in Terrier.

- Query Expansion (QE) using the LDA top 10 topic words. LDA topics words, which are related to the query and the top *k* documents that were retrieved by the query, were generated by an LDA model. The top 10 words were selected by the descending order of the word score based on the topic probability, word probability, and the rank of the retrieved document: TP * WP * (1 / (document rank) ^{2}) for the top *k* retrieved documents. Two types of LDA models with thresholds for TP, WP, and TP*WP were created:
  1) The basic QE model using the LDA model with a topic probability threshold, 0.01 (by default), because it is not effective to consider many topics with low topic probability values as related topics, topics with TP lower than 0.01 were ignored as unrelated topics.
  2) The QE model using the LDA model with specific LDA threshold values – e.g. the threshold, 0.08 or 0.1 for TP and 0 .03 for WP or 0.03 for TP*WP.
- QE using the words (MeSH terms) recommended by an (binary or 3-class) ANN classifier and an LDA model. The topic words generated by LDA models and then were classified into 2 or 3 groups for positive/negative/neutral words. The original word score (TP*WP) was weighted by the probability of being a positive word. Two types of QE models (WSW/PWS) were applied to select the top *k* words.
- QE using the words (MeSH terms) recommended by the ensemble QE models using (one classifier or multiple classifiers) and multiple WSW/PWS models. Words recommended by multiple WSW/PWS models were ranked by one classifier or multiple classifiers. The top *k* words by the ranking score were used for QE.

Evaluation for the IR tasks depended on the scheme of the TREC 2016 CDS track based on infAP (inferred Average Precision) and infNDCG (inferred Normalized Discounted Cumulative Gain) as IR evaluation measures.

3.6 Summary of Methodology

Three datasets were used for search engine indexing, LDA models, and ANN classifiers. Documents were represented by MeSH terms including 24,883 n-gram words based on 2016 Mesh descriptors.

- The PMC snapshot (the OA subset, Mar. 2016) for search engine indexing and IR evaluation including 1,495,289 full-text documents, which is provided by the 2016 TREC CDS track.
- The PMC 2016 (the OA subset, 12/04/2016) snapshot to generate LDA topic models including 1,451,661 documents.
- 424,288 words for training ANN classifiers.

Methods for each research question were listed in Table 6. Significant tests were conducted to compare mean values between two groups. Two-sample t-tests were conducted when comparing results with the baseline results including one infAP score and one infNDCG score, while paired t-tests were conducted when comparing paired results (e.g. 40 paired results for 40 LDA models with different topic numbers) between two groups.

Table 6. Methods for RQs

RQ1) How effective is the application of LDA topic words based on MeSH terms to QE in health IR?
• Topic word (MeSH term) generation by an LDA model with thresholds (e.g. TP: 0.01, 0.08 & 0.1, WP: 0.03, and TP*WP: 0.03) • Selection of the top 10 words by the word score (TP*WP / (doc. rank)^2) for QE • Comparison among QE models with different threshold values - Comparison of the average mean infAP and infNDCG scores - Comparison of the mean infAP and infNDCG scores of 40 LDA models and the baseline run (two-sample t-test)
RQ2) How effective is the application of LDA MeSH terms to QE in health IR when LDA topic words are weighted or selected by an ANN classifier?

- Selection of the top 7 or 10 words for QE using Word Score Weighting (WSW) by binary and 3-class ANN classifiers: word score * (weight for positive/negative/neutral words)
  - The weight for binary ANN classifier (2 layers and 700 nodes per layer):

    1) negative words: (1 – the probability of being a negative word) ^{2}

    2) positive words: (1 + the probability of being a positive word) ^{2}
  - The weight for 3-class ANN classifier (3 layers and 700 nodes per layer):

    1) negative words: (1 – the probability of being a negative word) ^{2}

    2) positive words: (1 + the probability of being a positive word) ^{2}

    3) neutral words: (1 – the probability of being a negative word)
- Selection of all or the top 7 positive words by the probability for positive word by ANN binary/3-class classifiers (Positive Word Selection, PWS)
- Comparison between QE models

  (WSW vs. PWS & binary classifier vs. 3-class classifier)
  - Comparison of the average mean infAP and infNDCG scores
  - Comparison of mean infAP and infNDCG scores of 40 WSW/PWS models and the baseline run (two-sample t-test)

RQ3) How effective are the ensembles of multiple LDA models and ANN classifiers in selecting MeSH terms for QE in health IR?

- Candidate words generated based on multiple WSW or PWS models (10 or 20 good-performed models in terms of infNDCG)
- Ensemble QE models based on the WSW model: the top *k* word selection for QE by ranking using one ANN classifier or multiple ANN classifiers
  - One classifier selects the top *k* words by the descending order of the probabilities for the positive word group.
  - Multiple classifiers are used to ignore the word if the sum of class scores of the word is less than a specific number (e.g. 3 or 5)—filtering. Candidate words are ranked by (the average class score) * (the average probability for the positive word group) estimated by multiple classifiers—ranking.

  * class score: 0 for negative words, 1 for neutral words (3-class classifiers), 2 for positive words
- Ensemble QE models based on the PWS model: the top *k* word for QE by ranking using one ANN classifier or multiple ANN classifiers in the same way as the ensemble QE models (WSW).
- Comparison among different ensemble QE models

(WSW vs. PWS & one classifier vs. multiple classifiers)

- Comparison of the mean infAP and infNDCG scores between the best runs of the ensemble QE models and the baseline run
- Comparison of the mean infAP and infNDCG scores between the best runs of the ensemble QE models and the baseline run for 30 queries (paired t-test)

For IR evaluation, infAP @1000 and infNDCG @1000 were measured on the 2016 TREC CDS evaluation scheme based on 30 IR Tasks (30 query topics).

Chapter 4 RESULTS & ANALYSIS

4.1 Parameter Setting

Parameters related to word selection for QE affect the baseline results. Two parameters, 1) the number of top-ranked retrieved documents to generate topic words and 2) a power value to weight word scores regarding the rank to score the words, were adjusted to generate better performance.

4.1.1 Topic word (MeSH term) scoring

When the topic probability of a query (the *summary* field, http://www.trec-cds.org/topics2016.xml) or a retrieved document is higher than or equal to 0.01, the topic is considered as a related topic to the query or the retrieved document. LDA topic words were identified by the topic of a query and the retrieved top-ranked documents. The query text was included in the text of the first ranked document. Using the rank of the retrieved document can be helpful to score the word for QE. For example, the topic words generated by the first-ranked documents have more weight than the topic words generated by the second-ranked document. To score words for QE, (TP * WP) values of the topic words and the rank of the retrieved document for the word were used: TP*WP / (document rank) ^{2}). A maximum of the top 10 words was selected as terms for QE by the descending order of the word score in 40 LDA models with different numbers of topics. If a word has more than two scores, the highest score was given to the word.

4.1.2 The number of the top-ranked retrieved documents

For 40 LDA models, infAP and infNDCG scores for 30 queries were calculated for 1000 results when terms from first-ranked document are selected for QE (Table 7). Even though there were five scores (in bold) shown more than the scores of the baseline run (infAP: 0.0209 & infNDCG: 0.1808), most LDA models showed lower infAP and infNDCG scores.

Table 7. Mean infAP and infNDCG scores of 40 LDA models with different numbers of topics for the top1 retrieved document (TP threshold: 0.01)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0163	0.0191	0.0203	0.0175	0.0152	0.0168	0.0166	0.0176	0.02	0.0175
infNDCG	0.1479	0.1596	**0.1817**	0.1645	0.1489	0.1678	0.1539	0.1729	0.1743	0.1731
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0179	0.0188	0.017	0.0156	0.0158	**0.0223**	**0.0247**	0.0188	0.0182	0.0167
infNDCG	0.1651	0.1754	0.1526	0.1565	0.1613	**0.1917**	**0.1845**	0.1837	0.1604	0.1594
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0179	0.0166	0.0205	0.0164	0.0165	0.0202	0.0188	0.0184	0.0208	0.0186
infNDCG	0.1783	0.1466	0.1747	0.1584	0.1644	0.175	0.1696	0.1627	0.1717	0.1744
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0186	0.0173	0.0188	0.0204	0.0181	0.0172	0.0169	0.0202	0.0192	0.0184
infNDCG	0.1744	0.1778	0.1674	0.1746	0.1761	0.1707	0.1689	0.1733	0.1712	0.1709

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

The top 2 retrieved documents include two documents when searching terms for query expansion: the first-ranked document and the second-ranked document. To compare the results for the top 2 retrieved documents with the results for the top1 document, infAP and infNDCG scores of 40 LDA models using the top 2 retrieved documents were listed in Table 8. Results were generated based on the ranking weight, the inverse value of the document rank to the power of two: 1 / (document rank) ^{2}.

Table 8. Mean infAP and infNDCG scores of 40 LDA models with different numbers of topics for the top 2 retrieved documents (the TP threshold, 0.01)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0161	0.0197	**0.023**	0.0202	0.0159	0.019	0.0202	0.0184	**0.0236**	0.0199
infNDCG	0.1513	0.1641	**0.1948**	0.1723	0.1567	0.1734	0.1738	0.1795	**0.1823**	0.**1841**
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0197	0.0193	0.0198	0.0201	0.0199	**0.0239**	**0.0255**	**0.0221**	**0.0211**	0.0204
infNDCG	0.1688	0.1746	0.1703	0.1768	0.1804	**0.1954**	**0.1935**	**0.1962**	0.1744	0.1786
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0203	0.0201	0.0208	**0.0213**	0.0181	**0.0223**	0.0197	**0.0213**	0.0206	**0.0212**
infNDCG	0.1868	0.1548	0.1777	0.1807	0.1682	0.1773	0.172	0.1718	0.1732	0.1778

no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0198	0.0198	**0.0216**	**0.0227**	0.0189	0.0199	**0.0232**	**0.022**	**0.0216**	0.0199
infNDCG	0.1662	**0.1822**	0.171	**0.184**	**0.1814**	0.1751	**0.1934**	0.1803	**0.1832**	0.1744

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

Being compared with the average mean infAP (0.0183) and infNDCG (0.1684) scores for the top1 document, the average mean infAP (0.0206) and infNDCG (0.1768) scores for the top 2 documents were higher, although the scores were lower than the scores for the baseline run, 0.0209 for infAP and 0.1808 for infNDCG. Several LDA models showed better infAP and infNDCG scores than the scores of the baseline run: 15 LDA models for infAP and 11 LDA models for infNDCG (Figure 7 & 8). LDA models for the top 2 retrieved documents have shown relatively better infAP and infNDCG scores than the LDA models for the top1 / top 2 / top3 (Appendix C) / top4 (Appendix D) / top5 (Appendix E) retrieved documents (Table 7 & 8). Therefore, the top 2 retrieved documents were used in generating topic words for proposed QE models.

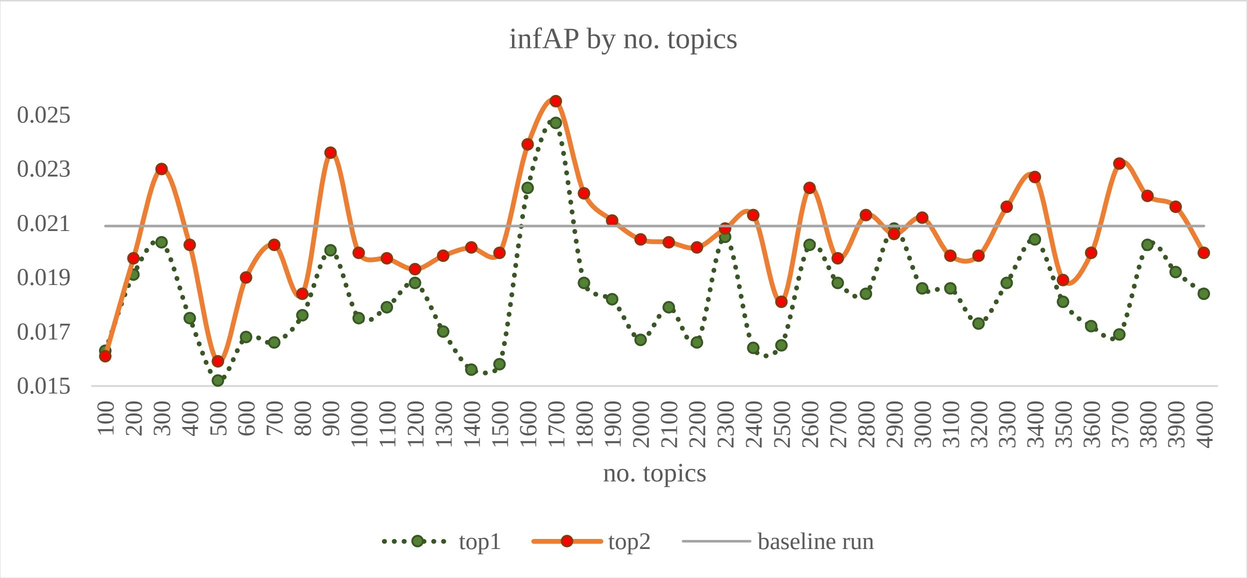

Figure 7. Mean infAP scores for the top1/top2 retrieved documents–weighing by the power of 2 for 40 LDA models

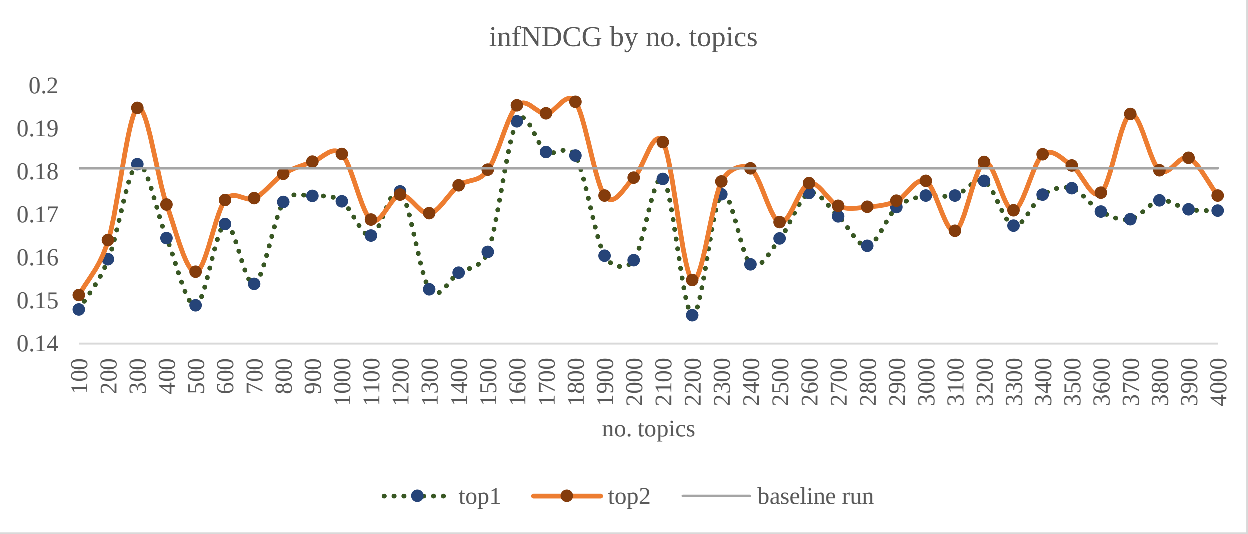

Figure 8. Mean infNDCG scores when top1/top2 retrieved documents are searched for query expansion terms–weighing by the power of 2 for 40 LDA models

4.1.1 Ranking weight

For the LDA model with 3700 topics and when top 2 retrieved documents are searched for expansion terms, which showed relatively high mean infAP (0.0232) and infNDCG (0.1934) scores, different ranking weights for scoring a word were given as the inverse value of the rank to the power of *k*: 1 / (document rank) ^{k}. Mean infAP and infNDCG scores were compared according to the power value, *k*, in Table 9.

Table 9. Mean infAP and infNDCG scores for the power values and the number of top retrieved documents (the LDA model with 3700 topics)

no. top docs	1	2	3	4	5	6	7	8	9	10
The power of 0.5										
infAP	0.0241	0.0251	0.0191	0.0203	0.0204	0.0196	0.02	0.0203	0.0191	0.0194
infNDCG	0.1955	0.1915	0.1811	0.1877	0.1833	0.1805	0.1754	0.1729	0.1752	0.1768
The power of 1										
infAP	0.0241	**0.0254**	0.0215	0.0225	0.024	0.0228	0.0214	0.0214	0.021	0.0211
infNDCG	0.1955	0.1942	0.1894	0.1951	0.1932	0.19	0.1864	0.1878	0.1826	0.1847
The power of 2										
infAP	0.0241	0.025	0.0251	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245

infNDCG	0.1955	**0.1988**	0.1971	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961
The power of 3										
infAP	0.0241	0.0242	0.0246	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242	0.0242
infNDCG	0.1955	0.1944	0.1968	0.1944	0.1944	0.1951	0.1944	0.1951	0.1944	0.1944

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

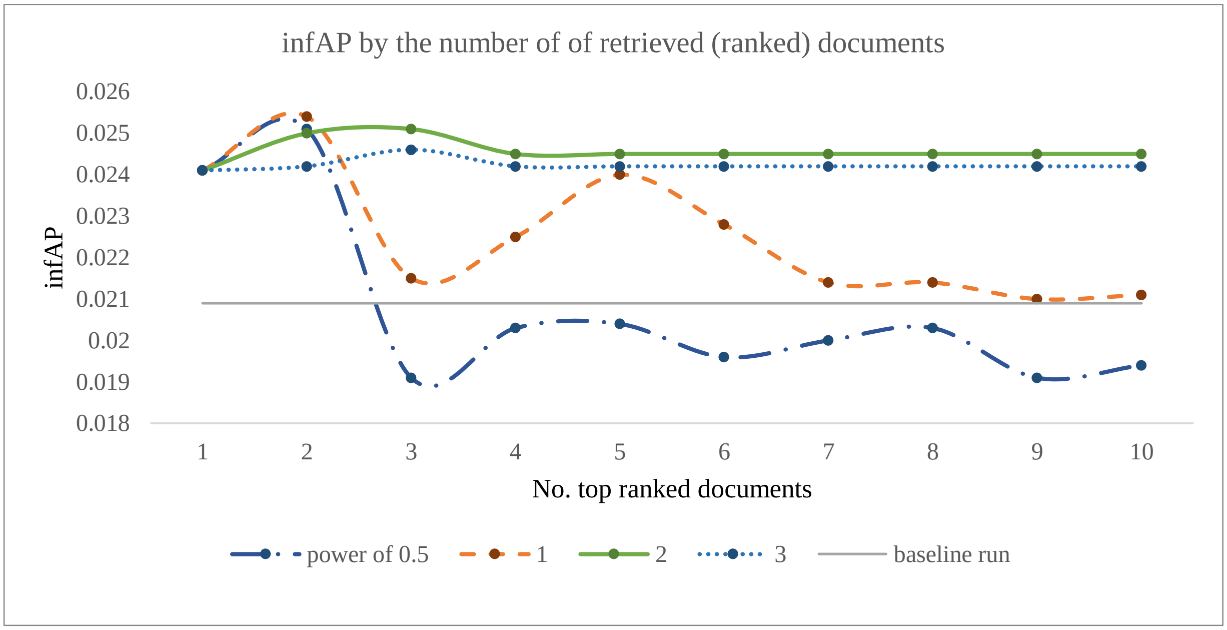

Figure 9. Mean infAP scores for ranking weight values by the number of top retrieved documents (the LDA model with 3700 topics)

QE using the power of 2 and 3 showed the stable infAP scores for the top 2 or 3 retrieved documents in Figure 9. Although the best mean infAP (0.0254) score for the top 1 document was observed when the power of 1 was applied, the mean infAP score for the power of 1 showed a high variance of infAP scores regarding 10 different numbers of retrieved documents.

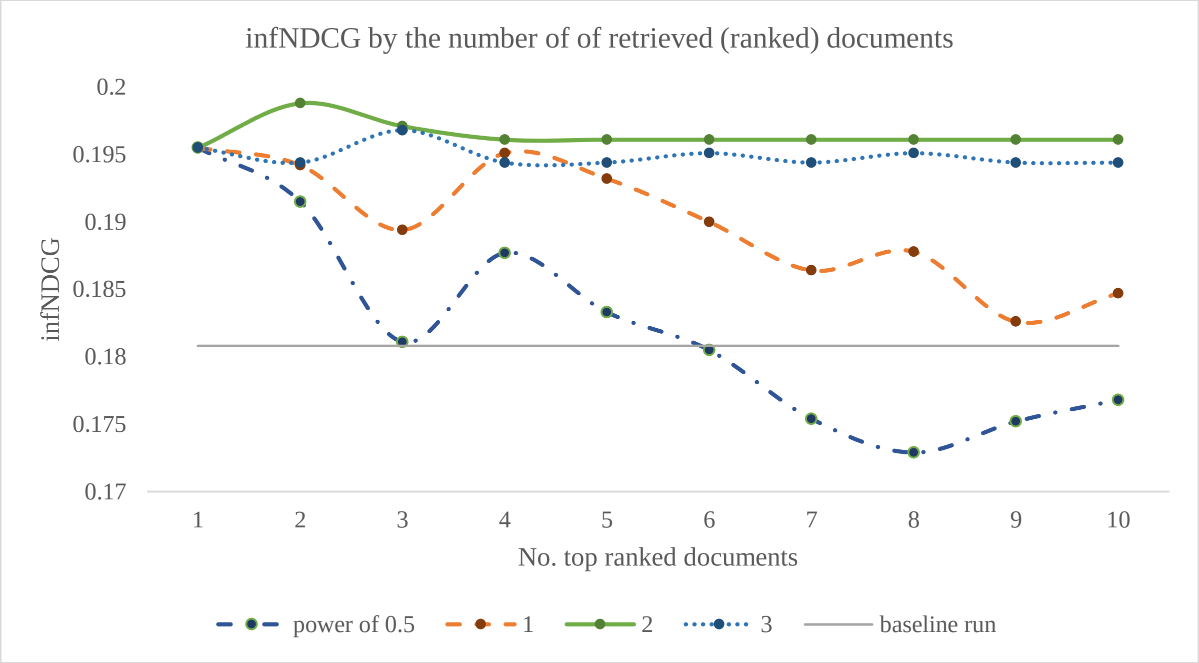

Figure 10. Mean infNDCG scores for ranking weight values by the number of top retrieved documents (the LDA model with 3700 topics)

Similarly, QE using top 2 or 3 retrieved documents showed stable and high infNDCG scores when the power of 2 and 3 were applied to the word score in Figure 10. The best infNDCG score (0.1988) for the top 2 retrieved documents was observed when the power of 2 was applied.

Mean InfAP and infNDCG scores of 40 LDA models for the top 2 retrieved documents were compared according to two different power values: the power of 1 (Table 10) and 2 (Table 8). The QE based on word scores weighted by the power of 2 showed slightly better average mean infAP (0.0206) and infNDCG (0.1768) scores of 40 LDA models than QE using the power of 1 (0.0204 for infAP and 0.1743 for infNDCG), although the QE using the power of 1 showed better mean scores in several LDA models: 11 LDA models with 100, 200, 1000, 1100, 1200, 1300, 2200, 2300, 2500, 2800, and 3000 topics (Figure 11) for infAP and 9 LDA models with 100, 200, 1100, 1300, 2200, 2500, 2800, 3100, and 3800 topics (Figure12) for infNDCG.

Table 10. Mean infAP and infNDCG scores for the LDA models with the weighting – the inverse value of the rank to the power of 1 for the top 2 retrieved documents (the best in bold)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0176	**0.0232**	0.0228	0.02	0.0145	0.018	0.0202	0.0178	0.0224	0.0206
infNDCG	0.1619	0.1749	**0.1897**	0.1686	0.1461	0.1694	0.1734	0.1698	0.1712	0.183
no. topics	**1100**	**1200**	**1300**	**1400**	**1500**	**1600**	**1700**	**1800**	**1900**	**2000**
infAP	0.0217	0.0197	0.0199	0.0193	0.0183	0.023	0.0246	0.0213	0.0207	0.0188
infNDCG	0.175	0.1658	0.1742	0.1759	0.1631	0.1867	0.1831	0.1939	0.1678	0.1774
no. topics	**2100**	**2200**	**2300**	**2400**	**2500**	**2600**	**2700**	**2800**	**2900**	**3000**
infAP	0.0194	0.0207	0.022	0.0208	0.0199	0.0215	0.0193	0.0215	0.0195	0.0226
infNDCG	0.184	0.1644	0.1777	0.1767	0.1728	0.1711	0.1683	0.1728	0.1713	0.17
no. topics	**3100**	**3200**	**3300**	**3400**	**3500**	**3600**	**3700**	**3800**	**3900**	**4000**
infAP	0.0194	0.0194	0.0215	0.0217	0.0187	0.019	0.0223	0.022	0.0211	0.019
infNDCG	0.1712	0.1782	0.1704	0.1778	0.1787	0.1724	0.1916	0.1829	0.179	0.1712

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

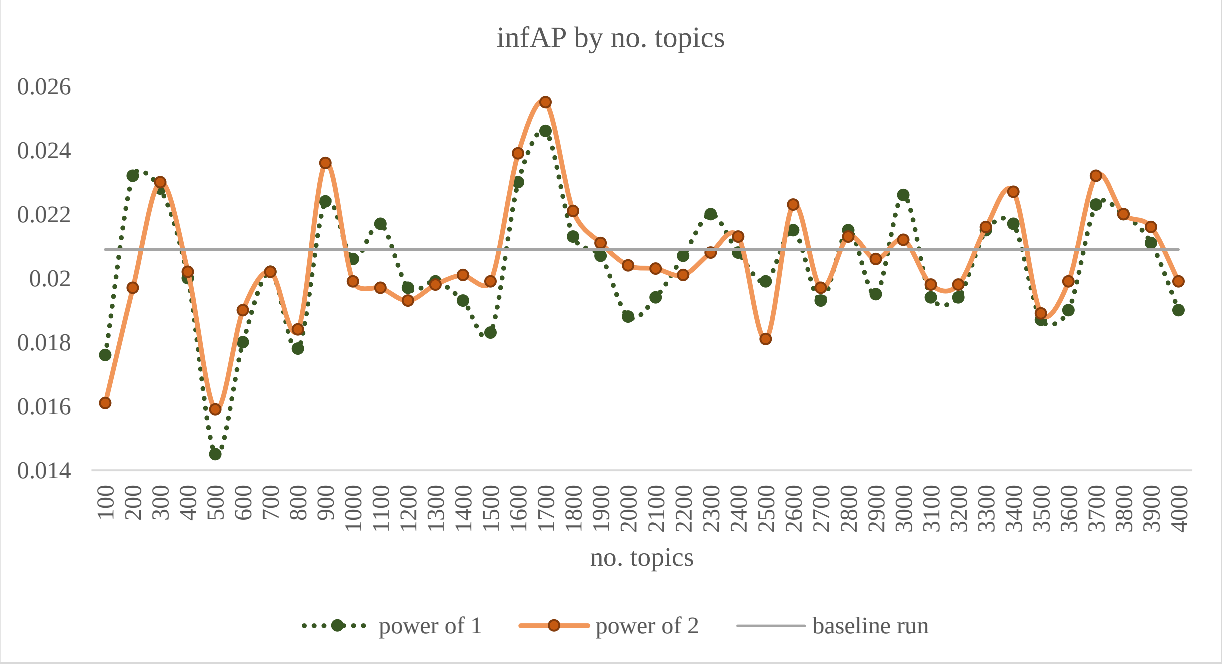

Figure 11. Mean infAP scores of 40 LDA models for the top 2 retrieved documents–weighted by the power of 1 and 2

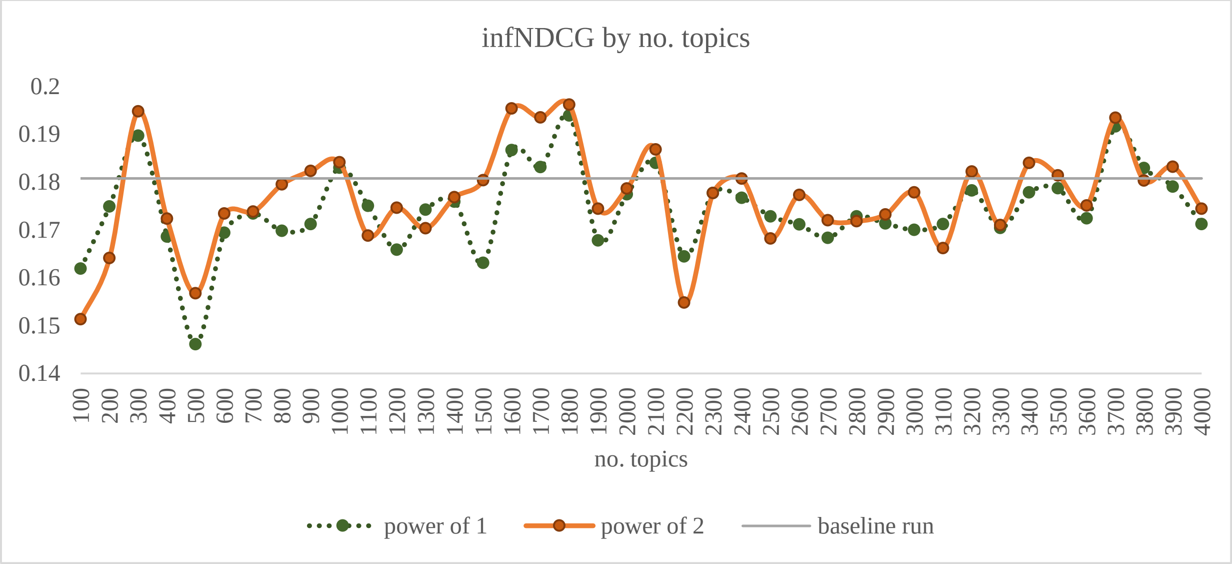

Figure 12. Mean infNDCG scores of 40 LDA models for the top 2 retrieved documents–weighted by the power of 1 and 2

4.2 QE using Thresholds for LDA TP, WP, and TP * WP (RQ1)

The thresholds for TP, WP, and TP*WP might affect IR performance. A high TP threshold would filter out minor topics from the top retrieved documents, while a high WP threshold would filter out less important words for a topic. For the model with 1700 topics, which showed a relatively high average infAP and infNDCG scores. infAP and infNDCG scores were measured by the thresholds for TP, WP, and TP * WP between 0 and 1.0 at 100 probability levels (level distance: 0.01).

4.2.1 QE using thresholds for TP and WP

According to different TP and WP threshold values, the mean infAP and infNDCG scores for the model with 1700 topics were calculated. The LDA model was generated based on the top1 retrieved document. The top 9 results by mean infAP and infNDCG scores were listed along with the number and ratio of positive and negative words in Table 11 and 12, respectively. For the case that there is no negative word, 1 is added for the divisor, preventing from being zero.

Table 11. Mean infAP and infNDCG scores for TP and WP thresholds sorted by infAP score (1700 topics based on the top1 retrieved document)

TP	WP	infAP	infNDCG	No. positive words	No. negative words	No. positive words / (No. negative words +1)
0.15	0.02	0.0277	0.1813	25	54	0.45
0.14	0.02	0.0276	0.1856	38	66	0.567
0.16	0.02	0.0274	0.18	21	49	0.42
0.09	0.02	0.0273	0.1876	64	110	**0.577**
0.08	0.02	0.0272	0.1869	64	110	**0.577**
0.15	0.03	0.0267	0.1892	23	51	0.442
0.11	0.02	0.0267	0.1872	49	92	0.527
0.19	0.02	0.0267	0.182	13	29	0.433
0.2	0.02	0.0267	0.182	13	29	0.433

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

TP values for the top 9 infAP scores were distributed between 0.08 and 0.2 for infAP, while there were only two WP values, 0.02 and 0.03. The highest infAP score was observed in the LDA models with the thresholds, 0.15 for TP and 0.02 for WP. The highest ratio of the number of positive words and negative words was shown in the LDA model with the thresholds, 0.09 / 0.08 for TP and 0.02 for WP.

Table 12. Mean infAP and infNCDG scores for TP and WP thresholds sorted by infNDCG score (1700 topics based on the top1 retrieved document)

TP	WP	infAP	infNDCG	No. positive words	No. negative words	No. positive words / (No. negative words + 1)
0.07	0.03	0.0258	0.1963	75	135	0.551
0.06	0.03	0.0253	0.1948	77	142	0.538
0.07	0.24	0.0237	0.1939	30	27	1.071
0.14	0.03	0.0266	0.1936	35	62	0.556
0.08	0.03	0.0263	0.1926	60	99	0.6
0.09	0.03	0.0263	0.1926	60	99	0.6
0.06	0.24	0.0233	0.1923	32	28	**1.103**
0.07	0.6	0.0219	0.1921	13	12	1.0

0.07	0.61	0.0219	0.1921	13	12	0.433

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

TP values were distributed between 0.06 and 0.14, while WP values were between 0.03 and 0.61 (0.03 for five cases) for the top 9 infNDCG scores. The highest inNDCG score was measured in the LDA model with the thresholds, 0.07 for TP and 0.03 for WP. The highest ratio (1.103) of positive words and negative words was shown in the LDA model with the thresholds, 0.06 for TP and 0.24 for WP where positive words were more than negative words.

The top 2 ranked TP (0.07 and 0.06) and WP (0.03) values were relatively low and positive and negative words were more than others. Based on the results, the thresholds for TP (0.1) and WP (0.03) were applied to 40 LDA models. The mean infAP and infNDCG scores were shown in Table 13. The average mean infAP and infNDCG scores of 40 LDA models with the thresholds for TP, 0.1, and WP, 0.03, were 0.0188 and 0.1633, respectively. The better scores than the scores of the baseline run were in bold. In most LDA models except for the LDA model with 1700 topics, infAP and infNDCG scores were lower than the scores of the baseline run.

Table 13. Mean infAP and infNDCG scores of 40 LDA models with the thresholds for TP – 0.1 and WP – 0.03 for the top1 retrieved document

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0135	0.019	0.019	0.0171	0.0156	0.0151	0.0191	0.0175	0.0203	0.0173
infNDCG	0.1265	0.1545	0.1598	0.1707	0.1534	0.1502	0.1648	0.1627	0.172	0.1692
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0171	0.019	0.017	0.0097	0.0166	**0.021**	**0.0254**	0.0188	0.0178	0.0192
infNDCG	0.152	0.1541	0.1531	0.1245	0.1615	0.1681	**0.1888**	0.1614	0.1551	0.1676
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0175	0.0185	0.0219	0.0168	0.0187	0.0194	0.018	0.0194	0.02	0.0175
infNDCG	0.1708	0.1573	0.1759	0.1541	0.1629	0.1561	0.1587	0.166	0.1599	0.17

no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0197	0.0202	0.019	**0.0227**	**0.0218**	0.0208	0.0205	0.0194	0.0203	**0.021**
infNDCG	0.1664	0.1667	0.1698	0.1731	**0.1842**	0.1683	0.1778	0.177	0.1745	0.1744

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

The ratio of positive and negative words (no. positive words / (no. negative words + 1)) can be another indicator to decide threshold values. The high ratios were shown in the LDA models with the threshold for WP, 0.24 (1.103 for TP, 0.06, and 1.071 for TP, 0.07) in Table 12. Because the infAP and infNDCG scores were low, another WP, 0.3 (more than 0.24, but roughly similar), was applied instead of 0.03 (Table 14). The ratio of positive (16) and negative words (16) generated by the LDA model with the thresholds (TP: 0.1 and WP: 0.3) were 0.9412 (16 / (1+16)).

Table 14. Mean infAP and infNDCG scores of 40 LDA models with the thresholds for TP (0.1) and WP (0.3) for the top1 retrieved document

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0204	0.0207	0.0205	0.0209	0.02	0.0197	0.0208	**0.0213**	**0.0214**	**0.023**
infNDCG	0.1789	0.1815	0.1782	0.1766	0.1739	0.1799	0.1807	**0.1828**	0.1781	**0.1842**
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	**0.0211**	0.0197	0.0209	0.0204	0.0189	**0.0224**	**0.0217**	0.0199	0.0197	**0.0218**
infNDCG	0.1795	0.1711	0.1766	0.1761	0.1666	0.1781	0.1793	0.1752	0.1732	**0.1881**
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	**0.0216**	0.0204	**0.0214**	**0.0214**	**0.022**	**0.0209**	**0.0213**	**0.0212**	0.0207	**0.0213**
infNDCG	**0.188**	0.1792	**0.1878**	0.1803	**0.1914**	**0.1804**	0.1731	**0.188**	0.1767	**0.1864**
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	**0.0231**	**0.0221**	**0.0233**	**0.0227**	**0.0223**	**0.0215**	**0.0228**	**0.0227**	**0.023**	**0.0225**
infNDCG	**0.1877**	**0.1917**	**0.1869**	**0.1861**	**0.1944**	**0.1893**	**0.1909**	**0.1871**	**0.192**	**0.1809**

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

Mean infAP and infNDCG scores were compared between two LDA models with different thresholds for TP and WP: 1) TP: 0.01 and 2) TP: 0.1 & WP: 0.3 (Figure 13 and 14). The average mean infAP and infNDCG scores of 40 LDA models with the thresholds for TP, 0.1, and WP, 0.3, were 0.0213 and 0.1819, respectively, which are higher than the scores of 40 LDA models with the thresholds for TP, 0.1, and WP, 0.03 (0.0188 for infAP and 0.1633 for infNDCG) as well as the scores for the baseline run. Better mean infAP and infNDCG scores were observed in 39 and 36 LDA models with the thresholds, 0.1 for TP and 0.3 for WP, respectively. Meanwhile, the average mean infAP and infNDCG scores of 40 LDA models with the threshold for TP, 0.01 were 0.0183 and 0.1684. There were statistically significant differences in the average mean infAP and infNDCG scores (paired t-test, alpha = 0.05, p-value = 2.3E-12 for infAP and 1.7E-09 for infNDCG).

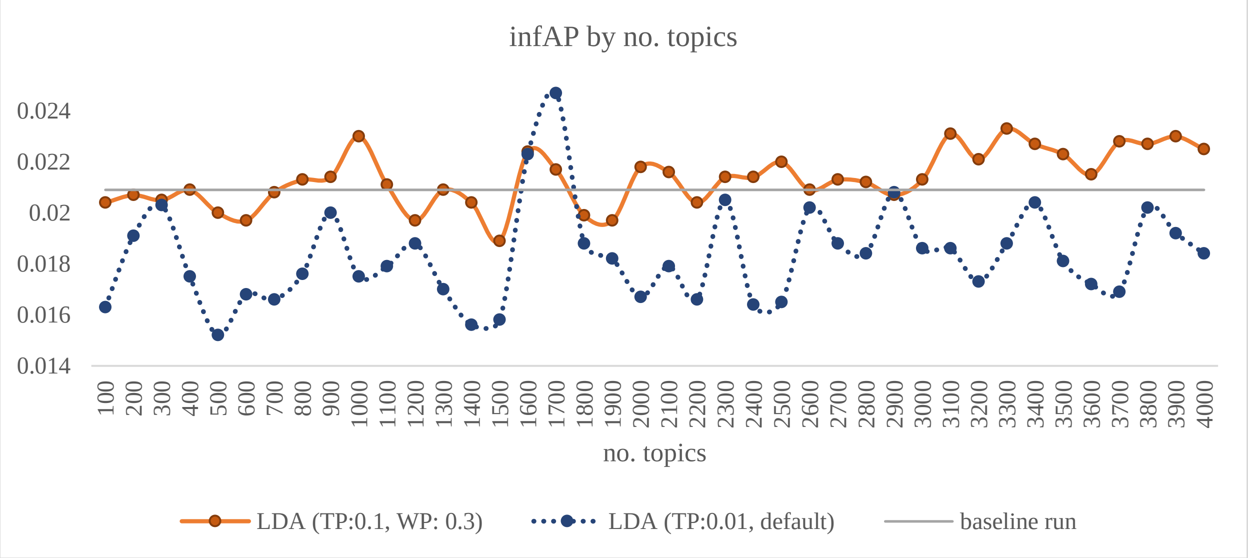

Figure 13. Mean infAP scores of 40 LDA models with different thresholds for TP and WP for the top1 retrieved documents

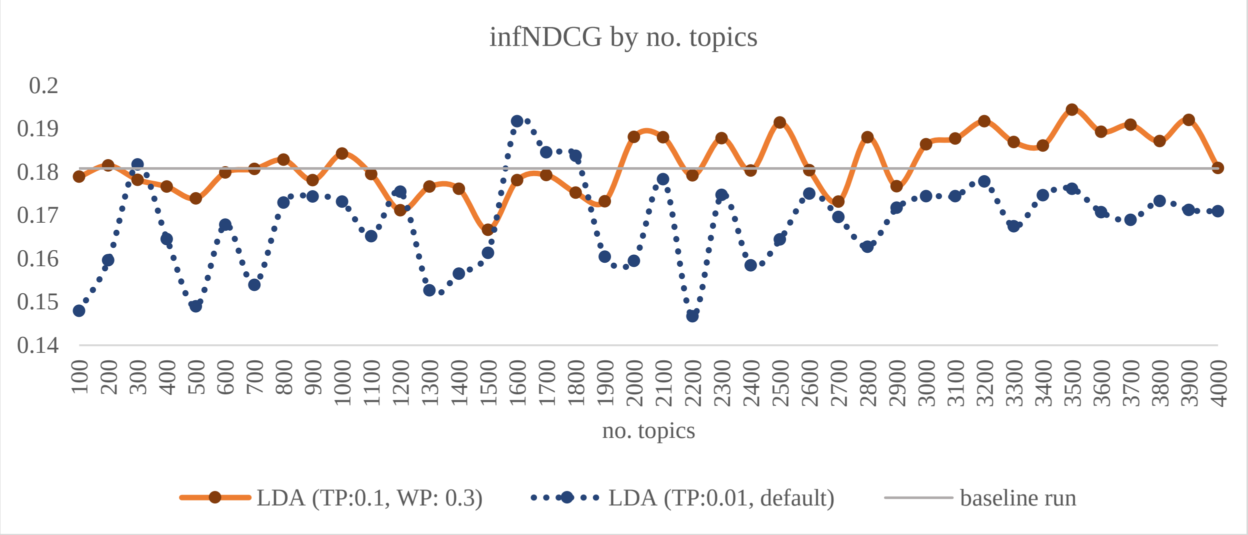

Figure 14. Mean infNDCG scores of 40 LDA models with different thresholds for TP and WP for top1 retrieved document

27 and 19 LDA models showed better mean infAP and infNDCG scores, respectively, than the baseline run. It implies that it might be effective to have specific thresholds for TP and WP, considering that just 2 and 3 LDA models with the threshold for TP, 0.01 (Table 7), showed better infAP and infNDCG scores, respectively, than the baseline run (Figure 13 & Figure 14). Compared with the scores of the baseline run, there was a statistically significant difference in the average mean infAP score, but not in the average mean infNDCG score (two-sample t-test, alpha = 0.05, p-value = 0.0135 for infAP and 0.2813 for infNDCG.

In a similar fashion, mean infAP and infNDCG scores of 40 LDA models with the thresholds for TP, 0.1, and WP, 0.3, were measured for the top 2 retrieved documents (Table 15). The average mean infAP and infNDCG scores were 0.0201 and 0.1696, respectively, which are lower than the scores of the baseline run as well as the scores of the LDA models for the top 1 retrieved document, 0.0213 for infAP and 0.1819 for infNDCG. Compared with the scores of the baseline run, there was a significant difference in the average mean score for infAP and infNDCG (two-sample t-test, alpha = 0.05, p-value = 0.0235 for infAP and 2.72E-11 for infNDCG).

Table 15. Mean infAP and infNDCG scores with the thresholds for TP – 0.1 and WP – 0.3 for the top 2 retrieved documents

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0178	**0.0231**	0.0203	0.0213	0.0176	0.0175	0.0209	0.0172	**0.024**	0.02
infNDCG	0.149	0.1764	0.1663	**0.1816**	0.161	0.1722	0.1683	0.1649	0.1796	**0.1833**
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0202	**0.0215**	**0.0212**	0.0139	0.0194	**0.0219**	**0.0232**	0.0202	0.0204	0.0183
infNDCG	0.1677	0.1769	0.1662	0.1452	0.1624	0.1775	0.1781	0.1756	0.1655	0.1691
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0166	0.0191	**0.0217**	0.0188	0.0189	**0.0213**	0.0193	0.0202	0.0187	**0.0236**
infNDCG	0.1671	0.1573	0.1803	0.1662	0.1567	0.1706	0.169	0.1704	0.1602	0.1802
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0189	0.0203	**0.0234**	0.0201	0.0197	**0.0214**	**0.0238**	0.0188	0.0207	0.0197
infNDCG	0.1533	0.1727	0.1706	0.1739	0.1725	0.1688	0.1862	0.1752	0.1777	0.168

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

4.2.2 QE using thresholds for TP and (TP * WP)

To find general threshold values for 40 LDA models, the ratio of positive words and negative words was referred rather than infAP and infNDCG scores. For the TP threshold, the ratio of positive words and negative words generated by the thresholds was calculated. TP values were sorted by the ratio for the top1 retrieved document in the LDA model with 1700 topics (Table 16). In the threshold, 0.08, negative words (315) were generated more than positive words (133) by more than twice.

Table 16. TP thresholds sorted by no. positive words / no. negative words (1700 topics) for the top1 retrieved document.

TP	No. positive words	No. negative words	No. positive words / (No. negative words +1)
0.08	133	315	0.421
0.09	131	311	0.420

0.07	168	403	0.416
0.06	175	426	0.410
0.14	74	192	0.383
0.13	79	207	0.380
0.05	**185**	495	0.373
0.11	99	265	0.372
0.12	83	223	0.371

In a similar way, the TP * WP values were sorted by the ratio of positive and negative words for the top1 retrieved document in the LDA model with 1700 topics (Table 17).

Table 17. TP * WP thresholds sorted by no. positive words / no. negative words (1700 topics) for the top1 retrieved document

TP * WP	No. positive words	No. negative words	No. positive words / (No. negative words +1)
0.13	5	5	0.833
0.05	22	26	0.815
0.08	11	13	0.786
0.04	28	35	0.778
0.06	16	20	0.762
0.07	12	17	0.667
0.09	8	12	0.615
0.02	50	81	0.610
0.14	3	4	0.600

TP * WP values showing a high ratio were between 0.02 and 0.14., TP * WP values less than 0.4 looked better because the LDA model with the threshold, 0.04, generated more positive words (28) than the LDA model with the threshold, 0.13 (5).

To improve infAP and infNDCG, two thresholds, 0.08 for TP and 0.03 for (TP * WP) were applied. A maximum of the top 10 words was chosen by the descending order of TP * WP / (document rank)^2. The infAP and infNDCG scores for the top 1 retrieved document were listed in Table 18. For more information, two LDA models with 50 topics and 4800 topics were generated but did not show interesting scores. The mean infAP and infNDCG for the LDA models with 50 topics and 4800 topics were 0.0196 & 0.1759 and 0.0224 & 0.1873, respectively.

Table 18. Mean infAP and infNDCG scores of 40 LDA models with different thresholds: TP (0.08) and TP * WP (0.03) for the top1 retrieved document

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0183	0.0226	0.022	0.0196	0.0175	0.0189	0.0205	0.0226	0.0218	0.0221
infNDCG	0.1688	0.1865	0.1917	0.1645	0.1606	0.1806	0.1742	0.1871	0.1837	0.1855
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0222	0.0204	0.0196	0.0193	0.0187	0.0221	0.0222	0.0225	0.0199	0.0205
infNDCG	0.1792	0.1731	0.1745	0.1786	0.1659	0.1875	0.1805	0.1965	0.1793	0.1873
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0203	0.0207	0.0211	0.0209	0.0228	0.0224	0.0216	0.022	0.0221	0.024
infNDCG	0.1859	0.1803	0.1847	0.1876	0.1904	0.1856	0.1811	0.1856	0.1795	0.1945
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0215	0.0239	0.0239	0.0234	0.0221	0.022	**0.0241**	0.0218	0.0218	0.0221
infNDCG	0.1827	0.1885	0.1885	0.1917	**0.1977**	0.1894	0.1955	0.1866	0.1838	0.1848

* baseline run – infAP: 0.0209 and infNDCG: 0.1808

When the mean infAP and infNDCG scores for the top1 retrieved document were compared with the mean scores of the baseline run, the average mean infAP and infNDCG scores of 40 LDA models were higher: 0.0214 and 0.183, respectively.

The LDA models with large numbers of topics showed better performance (Figure 15 & 16). Compared with the score of the baseline run, the LDA models with more topics than 2200 showed higher infAP scores. Meanwhile, the LDA model with smaller topics than 2300, 9 LDA models showed higher infAP scores than the score of the baseline run, but 13 models showed lower infAP scores. For LDA models with more topics than 2000, most LDA models showed higher infNDCG scores than the score of the baseline run, even though 2 LDA models with 2200 topics (0.1803) and 2900 topics (0.1795) showed lower infNDCG scores. Of the LDA models with 2000 or smaller numbers of topics than 2000, 8 LDA models showed higher infNDCG scores and 12 models showed lower infNDCG scores. There is statistically significant difference in the average mean infAP score in the two-sample t-test (alpha = 0.05, p-value = 0. 0335 for infAP), but not for infNDCG (p-value = 0.0712).

Also, the mean infAP and infNDCG scores of the LDA models with the thresholds (TP:0.08 & TP*WP: 0.03) were compared with the scores of 40 LDA models with only the default TP threshold value (0.01) in Figure 13 and 14. There were improvements in the LDA models the thresholds (TP:0.08 & TP*WP: 0.03): 38 LDA models for infAP and 36 LDA models for infNDCG (Figure 15 & 16). There were statistically significant differences of the average mean infAP and infNDCG scores between two groups in the paired t-test (alpha = 0.05, p-value = 3.3E-13 for infAP and 7.7E-13 for infNDCG).

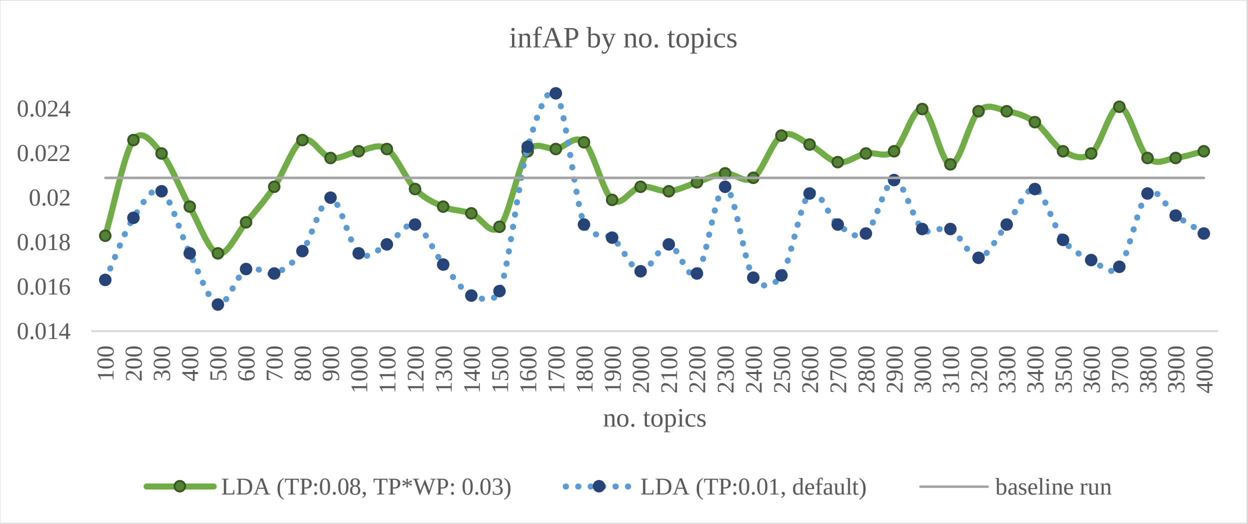

Figure 15. Mean infAP scores of 40 LDA models with different thresholds for TP and TP*WP for the top1 retrieved document

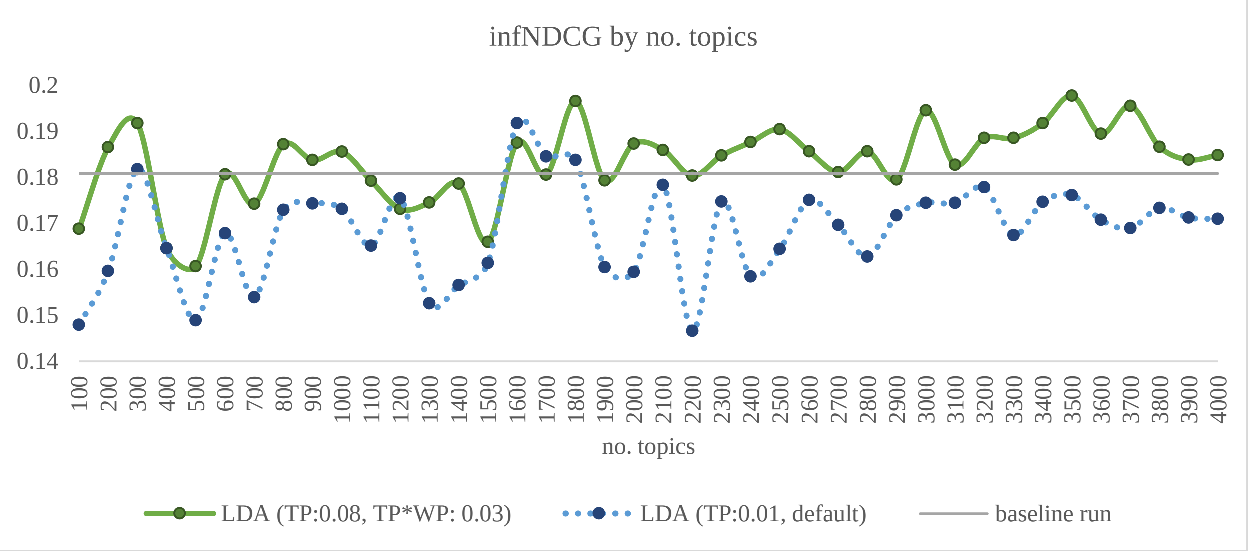

Figure 16. Mean infNDCG scores of 40 LDA models with thresholds for TP and TP*WP for the top1 retrieved document

One reason why infAP and infNDCG scores were not that high in 40 LDA models, might be that the threshold values for TP, WP, and TP * WP were optimized for a specific model (with 1700 topics). The ideal TP values would be different depending on individual LDA models, therefore, TP values would be standardized or normalized to be compared between models.

Two thresholds for TP (0.08) and TP*WP (0.03) were effective in increasing infAP for the top 2 retrieved documents (Table 19), while the average mean infNDCG score was lower than that the score of the baseline run. The average mean infAP and infNDCG scores were 0.0217 and 0.1804, respectively. The optimized threshold values would be found in a similar way to top1 retrieved document.

Figure 17 and 18 shows better mean infAP and infNDCG scores for the top 2 retrieved documents in 30 and 26 LDA model, respectively, compared with the scores of the LDA model with only one threshold for TP (0.01). There were statistically significant differences in mean infAP and infNDCG scores (paired t-test, alpha = 0.05, p-value = 0.00002 for infAP, 0.014 for infNDCG) between the LDA model with two thresholds (TP, 0.08 and TP * WP, 0.03) and the LDA model with the threshold (TP: 0.01, Table 8).

Compared with the baseline run, LDA models with two thresholds for TP (0.08) and TP*WP (0.03) showed a statistically significant difference in the average mean infAP score, but not in the average mean infNDCG score (two-sample t-test, alpha = 0.05, p-value = 0.0022 for infAP and 0.7341 for infNDCG).

Table 19. Mean infAP and infNDCG scores of 40 LDA models with the thresholds: TP (0.08) and TP * WP (0.03) for the top 2 retrieved documents

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0172	0.0236	0.023	0.0219	0.0187	0.0189	0.0209	0.0213	0.024	0.0226
infNDCG	0.1651	0.1919	0.1888	0.1739	0.1671	0.1737	0.1718	0.1745	0.1864	0.1887
no. topics	**1100**	**1200**	**1300**	**1400**	**1500**	**1600**	**1700**	**1800**	**1900**	**2000**
infAP	0.0234	0.0222	0.021	0.0213	0.0202	0.0227	0.0232	0.0225	0.0208	0.0197
infNDCG	0.1858	0.1775	0.1721	0.1753	0.1702	0.1839	0.1793	**0.1939**	0.1773	0.1845
no. topics	**2100**	**2200**	**2300**	**2400**	**2500**	**2600**	**2700**	**2800**	**2900**	**3000**
infAP	0.0214	0.0206	0.0225	0.0223	0.0214	0.0233	0.0211	0.0224	0.0201	0.0239
infNDCG	0.1888	0.1691	0.1876	0.1871	0.1823	0.19	0.178	0.1802	0.1697	0.1826
no. topics	**3100**	**3200**	**3300**	**3400**	**3500**	**3600**	**3700**	**3800**	**3900**	**4000**
infAP	0.0217	0.0208	**0.0238**	0.0222	0.0212	0.0213	0.0237	0.0205	0.0213	0.0216
infNDCG	0.1723	0.1743	0.1873	0.1837	0.1901	0.1881	0.1916	0.1757	0.178	0.177

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

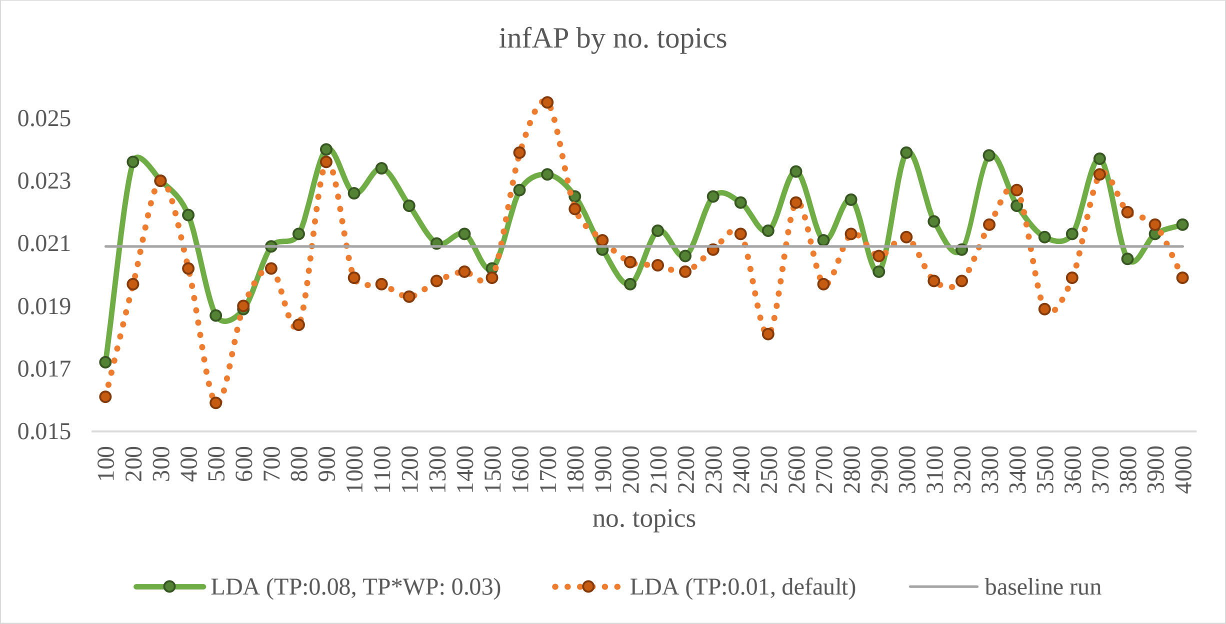

Figure 17. Mean infAP scores of 40 LDA models with different thresholds for TP and TP*WP for the top 2 retrieved documents

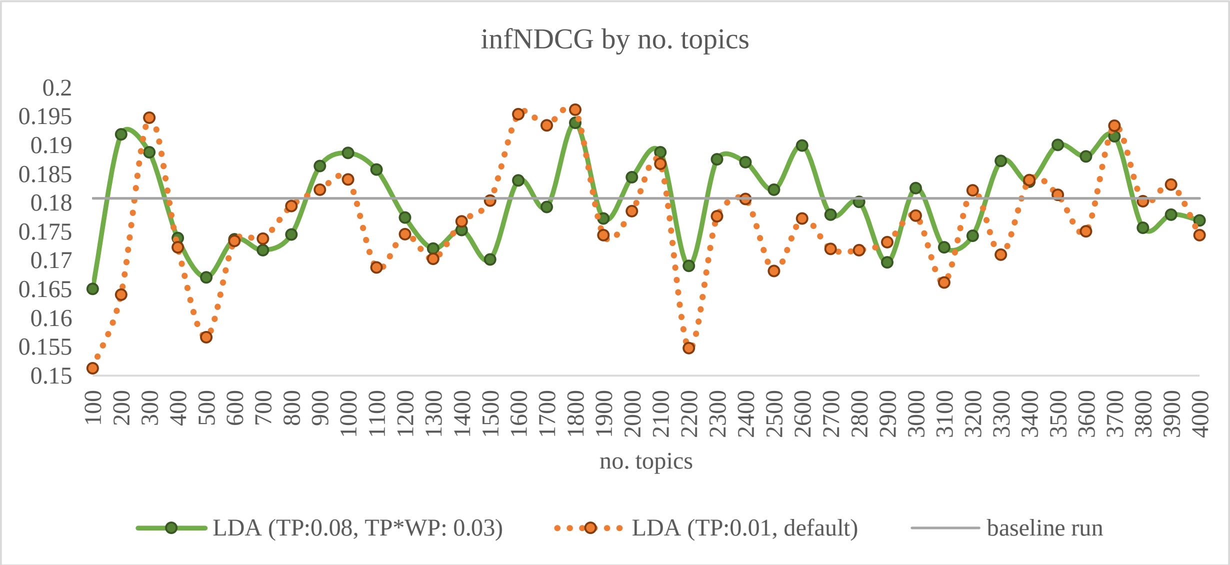

Figure 18. Mean infNDCG scores of 40 LDA models with different thresholds for TP and TP*WP for the top 2 retrieved documents

4.3 ANN Classifier Integration on LDA Models (RQ2)

Overall IR performance was shown better on the results for the top 2 retrieved (ranked) documents. Also, because QE using ANN models need enough candidate words generated by LDA models, the top 2 retrieved documents rather than the top 1 document were used in addition to considering the performance of the baseline run.

4.3.1 The Word Score Weighting (WSW) model

An ANN classifier assigns different weighting values for the positive/negative words. A classifier was used to give weight to topic words generated by LDA models. To evaluate the performance of the classifier and predict the classification for a topic word, 30 ANN classifiers were created based on the datasets regarding 30 queries. Each ANN binary classifier for a query was trained on a training data set, excluding the data related to the query. The trained classifier was used to predict topic words that were generated by an LDA model based on the excluded query. Weighting values can be given in various ways according to whether they are positive/negative/neutral words, which would be multiplied by the original word score. For sophisticated weighting, a probability estimated for each group was used. The power value, 2, was chosen to increase the IR performance.

- A binary ANN classifier with 2 layers and 700 nodes per layer:

  1) negative words: $(1 – \text{the probability to be classified into the negative word group})^2$

  2) positive words: $(1 + \text{the probability to be classified into the positive word group})^2$

- A 3-class ANN classifier with 3 layers and 700 nodes per layer:

  1) negative words: $(1 – \text{the probability to be classified into the negative word group})^2$

  2) positive words: $(1 + \text{the probability to be classified into the positive word group})^2$

  3) neutral words: (1 – the probability to be classified into the negative word group)

4.3.1.1 The binary ANN classifier

The QE models based on the WSW model (an LDA model + a binary ANN classifier) have shown relatively better average infAP and infNDCG scores comparing with the QE model depending on only an LDA model (Table 20), except for 10% of 40 LDA models: 4 LDA models with 300, 600, 900, and 1700 topics for infAP, and 4 LDA models with 300, 600, 900, and 1700 topics for infNDCG.

On the other hand, more LDA models showed better scores than the score of the baseline run when they were integrated with an ANN classifier: from 15 models to 38 models for infAP and from 11 models to 32 models for infNDCG. The highest scores, 0.0272 for infAP, 0.2056 for infNDCG were observed in the LDA model with 3000 and 2500 topics, respectively. There were statistically significant differences in the average mean infAP and infNDCG scores of 40 models in the paired t-test (alpha = 0.05, p-value = 1.19E-10 for infAP & 1.74E-08 for infNDCG).

Table 20. Mean infAP and infNDCG scores of 40 WSW (an LDA model + a binary ANN classifier) models for the top 2 retrieved documents (a maximum of the top 10 words for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0208	0.0219	0.0215	0.0213	0.0198	0.0185	0.0211	0.0215	0.0216	0.0228
infNDCG	0.1713	0.1925	0.1881	0.1789	0.1692	0.1723	0.1746	0.1835	0.1717	0.1914
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0244	0.0222	0.0226	0.0214	0.0207	0.025	0.024	0.0241	0.0234	0.0215
infNDCG	0.1864	0.1868	0.1809	0.1872	0.1719	0.1996	0.1883	0.2057	0.1888	0.1902
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0222	0.0242	0.0238	0.024	0.024	0.0239	0.0232	0.0241	0.0238	**0.0272**
infNDCG	0.1987	0.1876	0.1899	0.1972	**0.2056**	0.1808	0.1826	0.1916	0.1855	0.1949
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0222	0.0242	0.0255	0.0251	0.0222	0.0248	0.0257	0.0249	0.0246	0.0231
infNDCG	0.1867	0.1972	0.1876	0.1922	0.1998	0.1963	0.1937	0.1987	0.1997	0.1865

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

Mean infAP and infNDCG scores of 3 different types of QE models were compared in Figure 19 and 20.

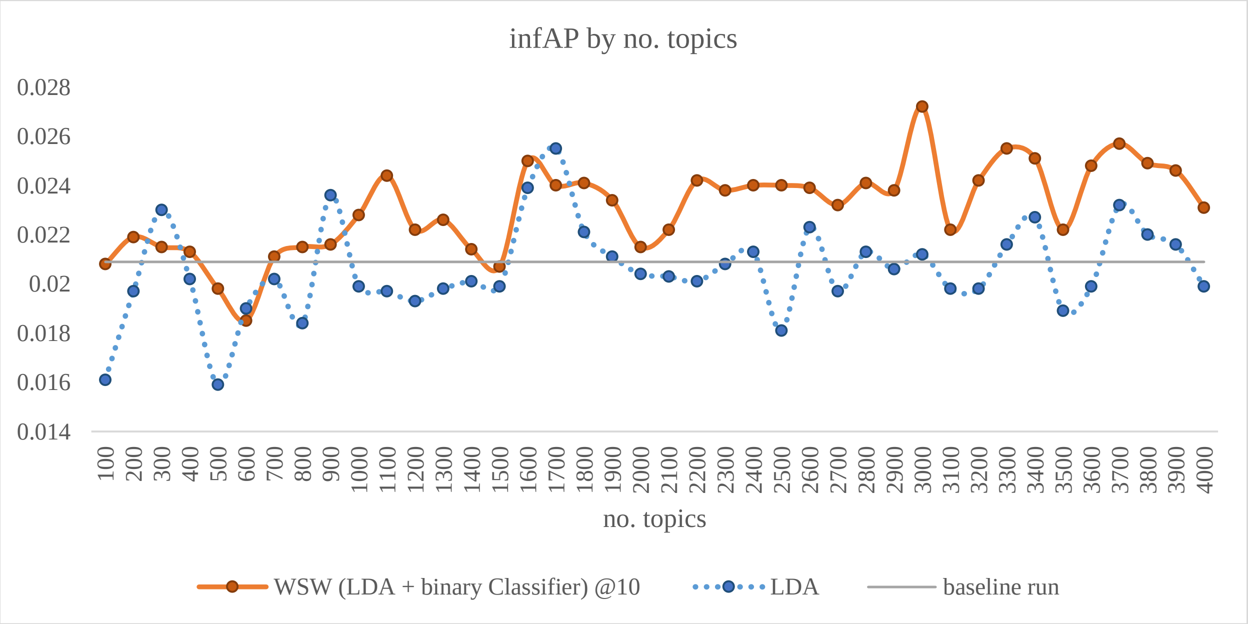

Figure 19. Mean infAP scores of 40 WSW (an LDA model + a binary ANN classifier) models for the top 2 retrieved documents

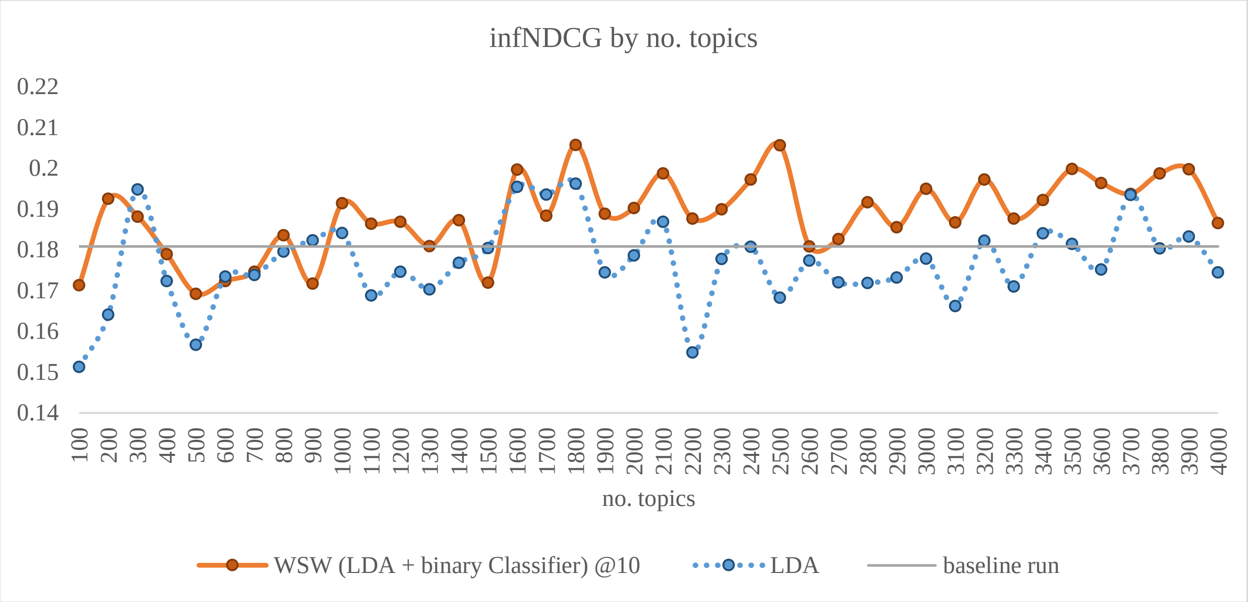

Figure 20. Mean infNDCG scores of 40 WSW (an LDA model + a binary ANN classifier) models for the top 2 retrieved documents

For more information, the IR performance of two different binary ANN classifiers were compared in Appendix F.

Adjusting the maximum number of words for QE was helpful slightly in increasing infAP and infNDCG. Table 21 shows the mean infAP and infNDCG scores when the maximum of the top 7 words was added to the original queries. Compared with the result for the maximum of the top 10 words, the mean infAP and infNDCG scores increased from 0.0231 to 0.0234 for infAP and from 0.1883 to 0.1891 for infNDCG, but there were no statistically significant differences in the average mean scores in the paired t-test (alpha = 0.05, p-values = 0.0979 for infAP and 0.3922 for infNDCG).

Table 21. Mean infAP and infNDCG scores of 40 WSW (an LDA model + a binary ANN classifier) models for the top 2 retrieved documents (a maximum of the top 7 words for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0212	0.0218	0.024	0.0231	0.0229	0.0196	0.0223	0.0238	0.0223	0.0246
infNDCG	0.1777	0.1854	0.1901	0.1857	0.1808	0.1758	0.1727	0.1843	0.1858	0.1858
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0239	0.0225	0.024	0.0227	0.0218	0.0247	0.0239	0.0239	0.0225	0.0234
infNDCG	0.1913	0.1827	0.1877	0.1902	0.1741	0.1924	0.193	0.1993	0.1851	0.191
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0237	0.0237	0.0231	0.0229	0.0246	0.0229	0.0235	0.0232	0.0228	0.0253
infNDCG	0.1975	0.1866	0.1906	0.1854	**0.1998**	0.1843	0.1821	0.2008	0.1929	0.1919
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0249	0.0241	0.0254	0.025	0.0227	0.0229	**0.0257**	0.0253	0.0234	0.0222
infNDCG	0.198	0.1979	0.1928	0.1918	0.1921	0.1947	0.1986	0.1965	0.1978	0.1812

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

4.3.1.2 The 3-class ANN classifier

The IR performance of the 3-class ANN classifier with 3 layers and 700 nodes per layer was not as good as the binary ANN classifier with 2 layers and 700 nodes per layer. Mean infAP and infNDCG scores were listed for the WSW (an LDA model + a 3-class classifier) model in Table 22 and compared with the mean scores of the baseline run in Figure 21 & 22. The average mean scores were 0.0217 for infAP and 0.1773 for infNDCG. Statistically significant differences were observed for infAP in a positive way (improvement) and infNDCG in a negative way (two-sample t-test, alpha = 0.05, p-value = 0.0096 for infAP and 0.0110 for infNDCG).

Table 22. Mean infAP and infNDCG scores of 40 WSW (an LDA model + a 3-class ANN classifier) models for the top 2 retrieved documents (a maximum of the top 10 words for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0175	0.0195	0.0248	0.0179	0.0198	0.0177	0.0208	0.0204	0.0211	0.0195

infNDCG	0.1686	0.1682	0.1896	0.1643	0.1679	0.1597	0.1825	0.1735	0.167	0.1709
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0219	0.0204	0.0209	0.0195	0.0199	0.025	0.023	0.0216	0.0209	0.0215
infNDCG	0.1785	0.1672	0.1806	0.1678	0.1703	0.1858	0.1784	0.1901	0.1701	0.1744
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0227	0.022	0.0246	0.0229	0.0228	0.0229	0.0227	0.0238	0.021	0.0242
infNDCG	0.1917	0.172	0.1882	0.1827	0.1824	0.1685	0.1765	0.1807	0.1756	0.1832
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0227	0.0224	0.023	0.0211	0.0212	0.024	0.0237	0.0225	0.022	0.0217
infNDCG	0.1818	0.1941	0.1745	0.1747	0.1823	0.1903	0.1862	0.1814	0.176	0.175

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

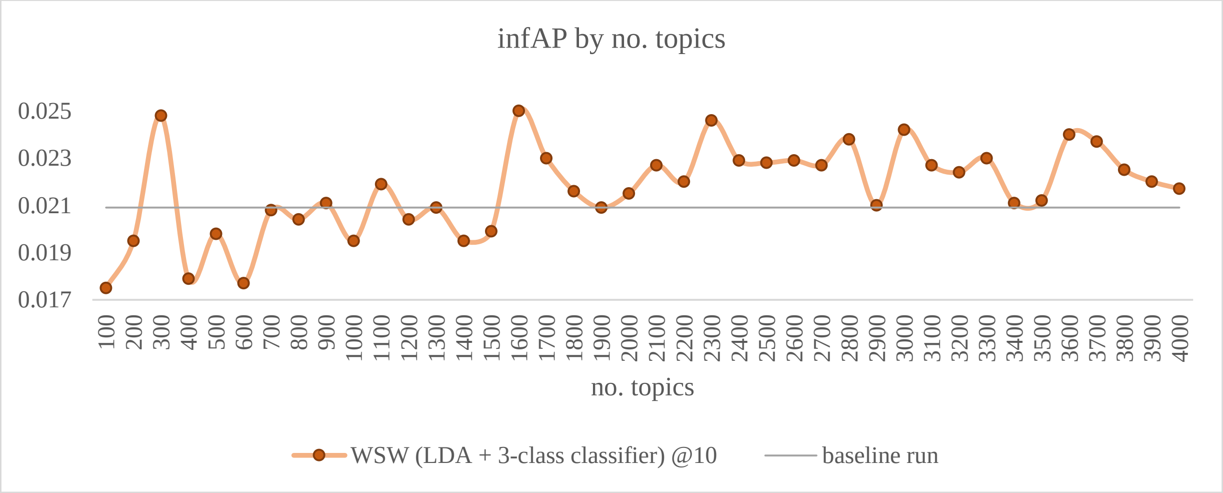

Figure 21. Mean infAP scores of 40 WSW (an LDA model + a 3-class ANN classifier) models for the top 2 retrieved documents based on word score weighting

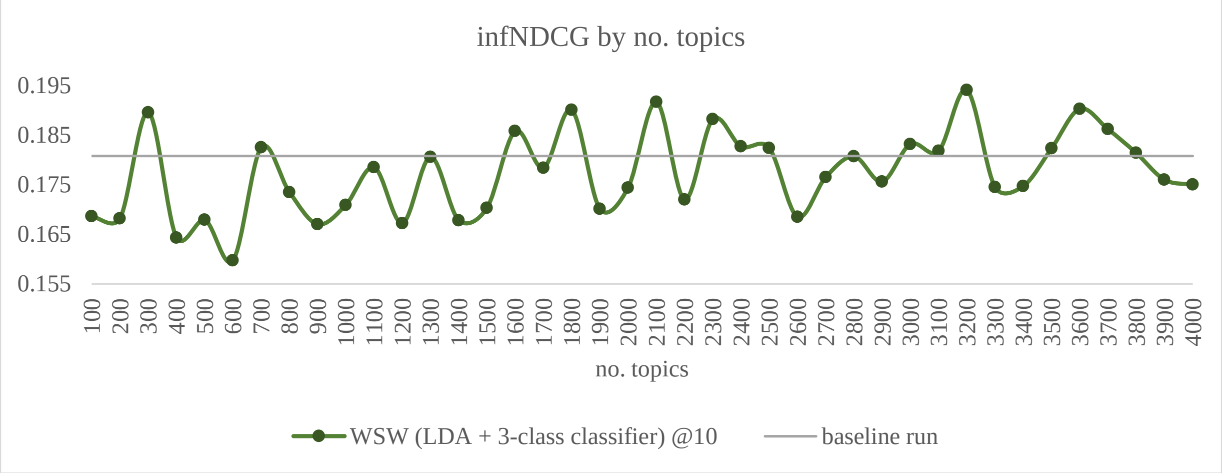

Figure 22. Mean infNDCG scores of 40 WSW (an LDA model + a 3-class ANN classifier) models for the top 2 retrieved documents based on word score weighting

4.3.2 The Positive Word Selection (PWS) model

Apart from QE based on word score weighting, the PWS model adds only positive words to an original query. To see how effective positive words are in IR, the queries were expanded by adding all positive words categorized by the ANN classifier. A binary ANN classifier with 2 layers and 700 nodes per layer and a 3-class ANN classifier with 3 layers and 700 nodes per layer were employed.

4.3.2.1 The binary ANN classifier

Mean infAP and infNDCG scores were described in Table 23. 33 models of 40 models showed better average infAP scores (82.5% of 40 models) than the scores of the baseline run, while 19 models showed better average infNDCG scores (47.5%).

Table 23. Mean infAP and infNDCG scores of 40 PWS (an LDA model + a binary ANN classifier) models for the top 2 retrieved documents (all positive words added for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0215	0.0188	0.0229	0.0222	0.0229	0.0195	0.0207	0.0228	0.0226	0.021
infNDCG	0.1787	0.1727	0.1829	0.1846	0.1842	0.1718	0.1753	0.1819	0.1863	0.1741
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000

infAP	0.0223	0.0209	0.0216	0.0226	0.0208	0.0217	0.0221	0.0213	0.0205	0.0221
infNDCG	0.1797	0.1773	0.183	0.1867	0.174	0.1821	0.1785	0.1835	0.176	0.1767
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0223	0.0224	0.0221	0.0231	0.0219	0.0223	**0.0236**	0.0217	0.0223	0.0223
infNDCG	0.1832	0.1817	0.1806	0.1841	0.1802	0.1794	**0.1943**	0.181	0.1797	0.1835
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.022	0.0223	0.0235	0.0204	0.0224	0.0218	0.0216	0.0214	0.0221	0.0213
infNDCG	0.1784	0.1839	0.1825	0.1761	0.1842	0.1823	0.1794	0.1766	0.1769	0.1768

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

When limiting the maximum number for QE up to 7, overall infAP and infNDCG have been improved (Table 24). Two PWS models were compared in Figure 23 (infAP) and 24 (infNDCG). Compared with the QE model using positive words without a maximum limit, there were better infAP scores in 30 models (75%) and infNDCG scores in 29 models (72.5%). Two QE models showed a statistically significant difference in mean infAP and infNDCG scores (alpha = 0.05, p-value = 0.0004 for infAP & 0.0006 for infNDCG). Adjusting the maximum number for QE from 10 to 7 was effective in increasing mean infAP and infNDCG scores.

Table 24. Mean infAP and infNDCG scores of 40 PWS (an LDA model + a binary ANN classifier) models for the top 2 retrieved documents (a maximum of the top 7 positive words added for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.021	0.0207	**0.0236**	0.0223	0.0231	0.0215	0.0217	0.0216	0.0217	0.0227
infNDCG	0.1822	0.1817	0.1875	0.1826	0.1857	0.1819	0.1799	0.1799	0.1829	0.1796
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0223	0.0222	0.0222	0.0226	0.0218	0.0218	0.0229	0.0214	0.0216	0.022
infNDCG	0.1802	0.1832	0.1822	0.184	0.179	0.1803	0.1841	0.1812	0.1809	0.178
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0219	0.0231	0.0233	0.0234	0.0229	0.0233	0.0233	0.0232	0.0231	**0.0236**
infNDCG	0.1812	0.1834	0.1868	0.1856	0.1849	0.1854	0.1863	0.1862	0.186	**0.1884**
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000

infAP	0.0224	0.0232	0.0216	0.021	0.0226	0.0231	0.0227	0.0231	0.0224	0.0208
infNDCG	0.1835	0.186	0.1819	0.182	0.1842	0.1855	0.1834	0.1847	0.1824	0.1796

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

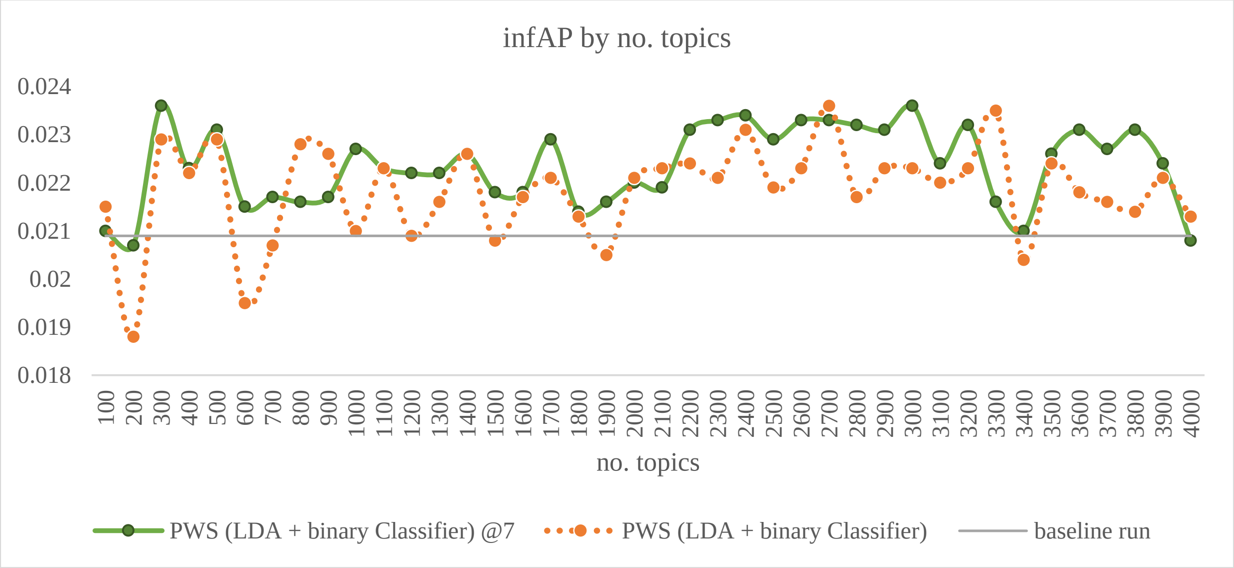

Figure 23. Mean infAP scores of 40 PWS (an LDA model + a binary ANN classifier) models for the top 2 retrieved documents (QE using positive words)

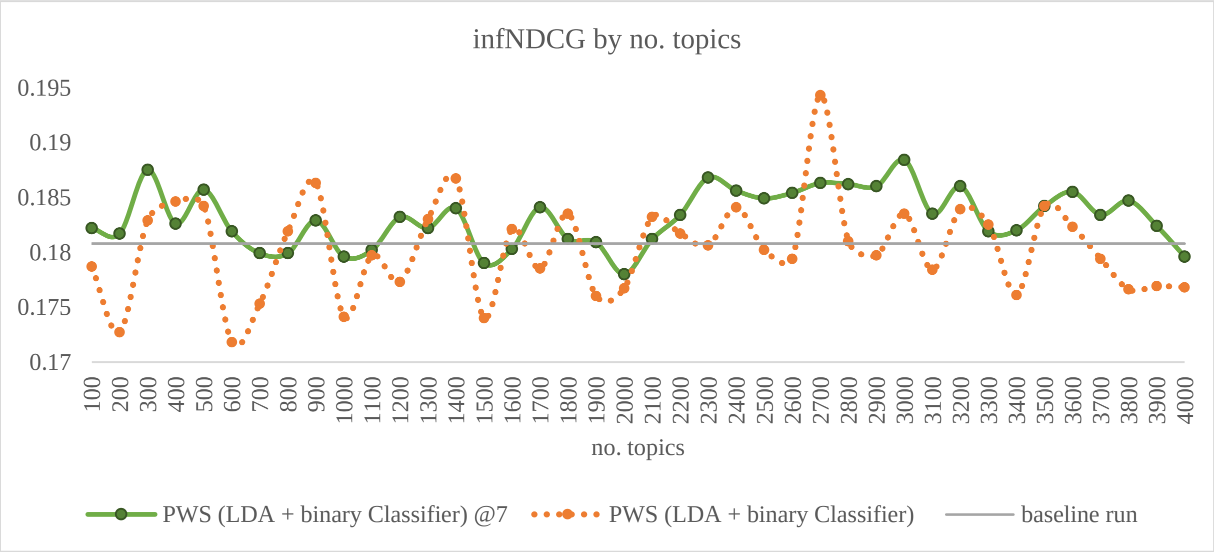

Figure 24. Mean infNDCG scores of 40 PWS (an LDA model + a binary ANN classifier) top 2 retrieved documents (QE using positive words)

4.3.2.2 The 3-class ANN classifier

Mean infAP and infNDCG scores of the PWS (an LDA model + a 3-class classifier with 3 layers and 700 nodes per layer) model were described in Table 25. The average mean scores (0.0191 for infAP and 0.1698 for infNDCG) were statistically significantly lower than the scores of the baseline run (two-sample t-test, alpha = 0.05, p-value = 3.47E-08 for infAP and 2.16E-13 for infNDCG). Differently from the binary ANN classifier, the 3-class ANN classifier was not effective in increasing infAP and infNDCG scores.

Table 25. Mean infAP and infNDCG scores of 40 PWS (an LDA model + a 3-class ANN classifier) models for the top 2 retrieved documents (only positive words added for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0194	0.0178	**0.0236**	0.0195	0.0181	0.0183	0.0202	0.0205	0.0172	0.0186
infNDCG	0.1698	0.1656	0.1795	0.1724	0.1707	0.1685	0.1726	0.1696	0.1642	0.1623
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0218	0.0212	0.02	0.0185	0.0187	0.0208	0.0202	0.0159	0.0167	0.0194
infNDCG	0.1812	0.1769	0.1812	0.1782	0.1677	0.1693	0.178	0.1565	0.1611	0.172
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0197	0.0184	0.0202	0.0172	0.0173	0.0177	0.0202	0.0211	0.0177	0.0194
infNDCG	0.1786	0.1731	0.1732	0.1598	0.1462	0.1559	0.1719	0.1765	0.1639	0.1717
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0203	0.016	0.0185	0.0194	0.0192	0.0153	0.0176	0.0222	0.0214	0.0207
infNDCG	0.1728	0.1656	0.1686	0.1695	0.1745	0.1581	0.1654	0.1728	**0.1841**	0.1737

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

4.4 Ensemble QE models (RQ3)

Proposed ensemble QE models were designed by integrating multiple LDA models and classifiers. Two types of ensemble QE models were introduced according to whether topic words were recommended by weighed word scores (word score * weight by an ANN classifier) or selecting positive words by an ANN classifier (the PWS model).

4.4.1 The ensemble of multiple LDA models and ANN classifiers based on Word Score Weighting (WSW)

Candidate words for QE were recommended by multiple WSW models. In each WSW model, topic words were sorted by the word score (TP*WP / (document rank)^2) and then weighted by an ANN classifier. The top 10 words per query from each WSW model were collected. 200 words (the top 10 words * 20 LDA models) per query, which were generated from 20 WSW (an LDA model + a binary ANN classifier) models, were ranked by one classifier or three classifiers as follows.

1. Topic words were generated by 20 LDA models of which mean infNDCG scores were relatively high.
2. Those words were scored by (TP * WP / (document rank)^2) in each LDA model, which were weighted by the probability estimate for the positive/negative/neutral word group by the binary ANN classifier with 2 layers and 700 nodes per layer.
3. A maximum of the top k ($k = 10$) words per query (30 queries) was selected from each WSW model by the descending order of the weighted word score as candidate words for QE: 300 words (the top 10 words * 30 queries) for 30 queries from each WSW model. Totally 6000 words (300 words per WSW model * 20 WSW models) were collected from 20 WSW models.

   – When using one binary classifier with 2 layers and 700 nodes per layer, 200 candidate words per query (top 10 words per query * 20 WSW models) were ranked by the descending order of the probability for the positive word group without calculating class scores.

   – When using three classifiers (one binary classifier with 2 layers and 700 nodes per layer and two 3-class classifiers with 2 layers & 3 layers with 700 nodes per layer), the class score of a word was calculated according to the classification by each classifier: 0 for a negative word, 1 for a neutral word, and 2 for a positive word. For word ranking and filtering, 1) the sum of class scores of a word and 2) (the average of four class scores) * (the average of four probabilities for the positive word group), were calculated by four classifiers (three classifiers plus one classifier included in the WSW model). If the sum of class scores of a candidate word is less than 3, the word was not considered as QE terms. Of 6000 words, 2981 words were ignored. The remaining 3019 words were scored by (the average of three class scores) * (the average of three probabilities for the positive word group.

4. The top *k* (*k* = 1…30) words were added to the original query for QE.

Mean infAP and infNDCG scores of two ensemble QE models based on WSW (one classifier vs. multiple classifiers) were compared by the number of the top words added for QE in Figure 25 and 26. Mean infAP and infNDCG scores based on one classifier and three classifiers were listed in Table 26 and 27, respectively. The expanded queries using more than 25 words were identical because no new words were added in the expanded queries using more than 25 words. Word filtering and ranking by multiple classifiers were helpful in increasing overall infAP and infNDCG scores. When the top 3 words in the ensemble QE model using multiple classifiers were added to the original query, the performance was most improved (infAP: 0.0271 and infNDCG: 0.2055), while the best infAP and infNDCG scores of the ensemble QE model using one classifier were 0.0247 and 0.1953 when adding the top 19 and 23 terms to the original queries. Ranking by the class score and the probability for the positive group was effective in selecting relevant words for QE, while word cut-off by the class score was effective in removing irrelevant words. All 30 expanded queries using multiple classifiers showed better mean infAP and infNDCG scores than the scores for the ensemble QE model (WSW) using one classifier.

Table 26. Mean infAP and infNDCG scores of the ensemble QE model using one classifier based on 20 WSW (an LDA model + one classifier) models for the top 2 retrieved documents

no. words	1	2	3	4	5	6	7	8	9	10
infAP	0.0209	0.0209	0.0209	0.0209	0.0207	0.0207	0.0207	0.0213	0.0226	0.0217
infNDCG	0.1808	0.1808	0.1808	0.1808	0.1808	0.1812	0.1816	0.1794	0.1835	0.1792
no. words	11	12	13	14	15	16	17	18	19	20
infAP	0.0223	0.0233	0.0233	0.0225	0.0224	0.0223	0.0241	0.0246	**0.0247**	0.0243
infNDCG	0.1773	0.1874	0.1858	0.1823	0.1819	0.188	0.1916	0.1947	0.193	0.1941
no. words	21	22	23	24	25	26	27	28	29	30

infAP	0.0237	0.0231	0.0225	0.0223	0.0214	0.0219	0.0215	0.0205	0.0197	0.0184
infNDCG	0.1911	0.1948	**0.1953**	0.1937	0.1868	0.1877	0.1841	0.176	0.1761	0.1716

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

Table 27. Mean infAP and infNDCG scores of the ensemble QE model using three classifiers based on 20 WSW (an LDA model + one classifier) models for the top 2 retrieved documents

no. words	1	2	3	4	5	6	7	8	9	10
infAP	0.0213	0.0235	**0.0271**	0.0242	0.025	0.0251	0.0245	0.0251	0.0251	0.0249
infNDCG	0.1816	0.1928	**0.2055**	0.195	0.1977	0.2011	0.1966	0.1991	0.2002	0.199
no. words	11	12	13	14	15	16	17	18	19	20
infAP	0.025	0.0249	0.0249	0.025	0.0252	0.0252	0.0252	0.0252	0.0254	0.0254
infNDCG	0.1986	0.1984	0.1982	0.2006	0.2025	0.2025	0.2025	0.2025	0.2033	0.2033
no. words	21	22	23	24	25	26	27	28	29	30
infAP	0.0254	0.0254	0.0254	0.0253	0.0253	0.0253	0.0253	0.0253	0.0253	0.0253
infNDCG	0.2033	0.2033	0.2031	0.2023	0.2023	0.2023	0.2023	0.2023	0.2023	0.2023

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

For instance, the top3 words added to the 10th original query ("*A 55-year-old woman with sarcoidosis, presenting today with confusion and worsening asterixis   In the waiting room, the pt became more combative and then unresponsive Ammonia level 280 on admission*") were "*prognosis*", "*France*", and "*urea*". The infAP (0.0168 → 0.0409) and infNDCG (0.1387 → 0.2055) scores increased in the expanded query. The top 3 terms used for QE were described in Appendix G.

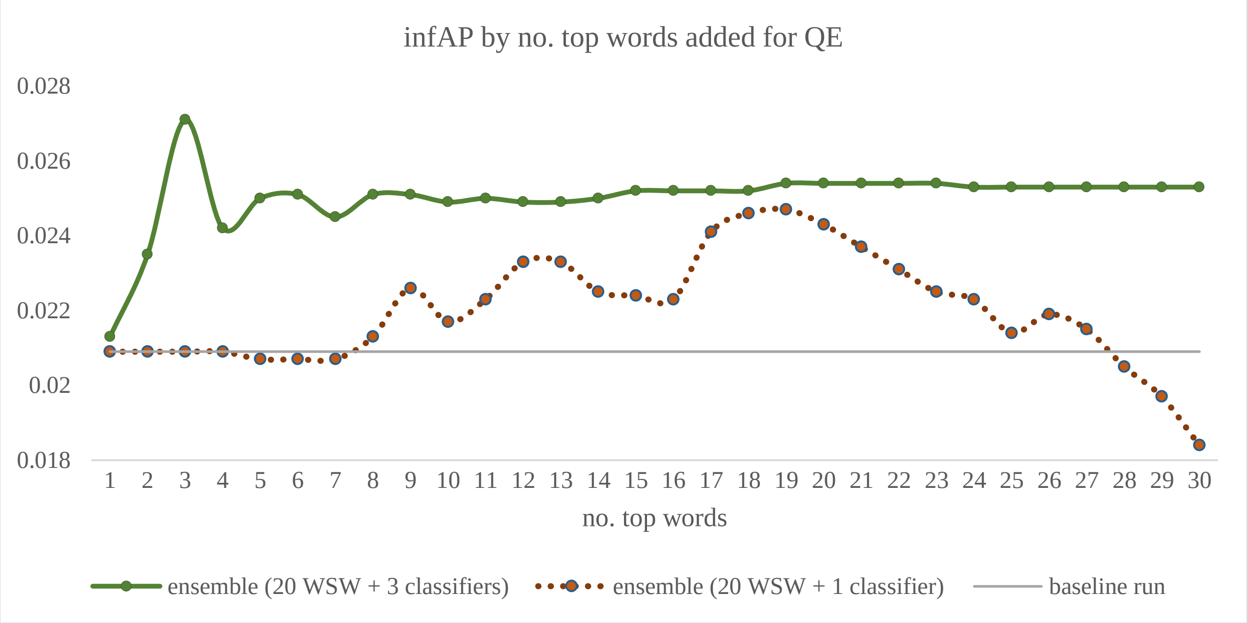

Figure 25. Mean infAP scores of the ensemble QE model (WSW) for the top 2 retrieved documents (multiple classifiers vs. one classifier)

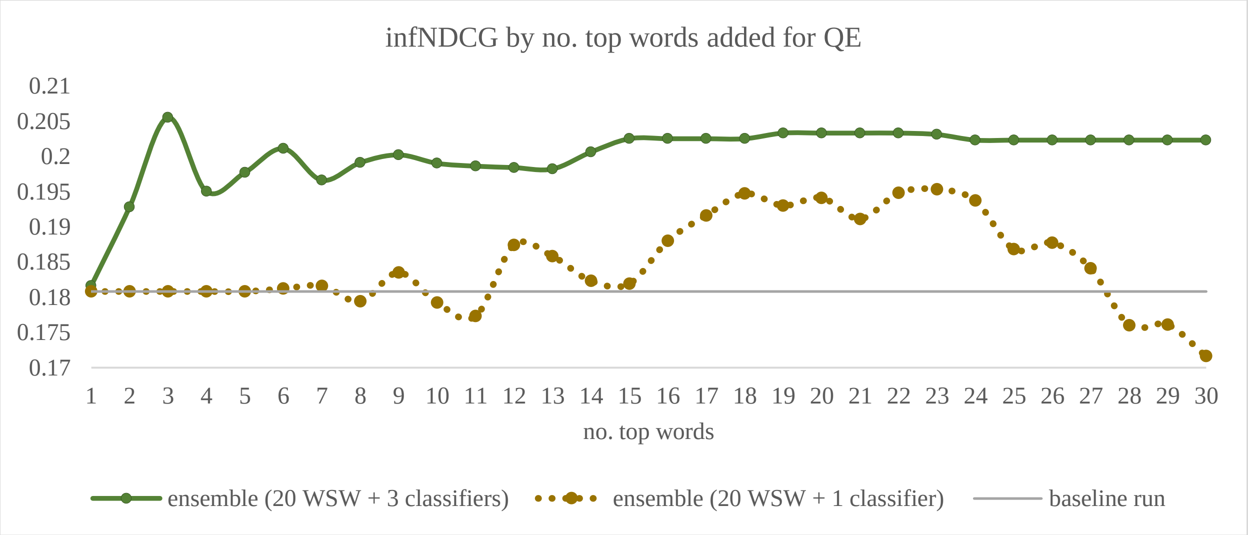

Figure 26. Mean infNDCG scores of the ensemble QE model (WSW) for the top 2 retrieved documents (three classifiers vs. one classifier)

For the best result of the ensemble QE model (three classifiers), which were expanded by the top 3 words, infAP and infNDCG scores for 30 queries were compared with the scores of the baseline run in Table 28 & 29 and Figure 27 & 28. There were improvements in terms of infAP and infNDCG in 22 queries and 21 queries of 30 queries, respectively. There were statistically significant differences in the mean infAP and infNDCG scores for 30 queries in the paired t-test (alpha = 0.05, p-value = 0.005 for infAP and 0.0029 for infNDCG). If the classifiers were trained on more data (including more queries) and better features, the IR performance would increase.

Table 28. Mean infAP and infNDCG scores of the baseline run for 30 queries

Query No	1	2	3	4	5	6	7	8	9	10
infAP	0.0186	0.0088	0.0005	0.0024	0.007	0.0339	0.0158	0.041	0.0229	0.0168
infNDCG	0.1388	0.0734	0.0198	0.0177	0.0828	0.2595	0.0985	0.6742	0.1955	0.1387

Query No	11	12	13	14	15	16	17	18	19	20
infAP	0.0153	0.0165	0.0232	0.006	0.0118	0.002	0.0409	0.0152	0.0033	0.0349
infNDCG	0.3224	0.1562	0.1688	0.0728	0.1331	0.049	0.2695	0.1095	0.0793	0.6214

Query No	21	22	23	24	25	26	27	28	29	30
infAP	0.0054	0.0351	0.0012	0.0169	0.0021	0.0203	0.0083	0.0031	0.1176	0.0806
infNDCG	0.0535	0.1373	0.0357	0.3979	0.073	0.162	0.1008	0.1229	0.4087	0.2515

Table 29. Mean infAP and infNDCG scores of the ensemble QE model (WSW) using the top 3 words for 30 queries

Query No	1	2	3	4	5	6	7	8	9	10
infAP	0.0284	0.0289	0.0006	0.0054	0.0065	0.0376	0.0458	0.0327	0.0221	0.0409
infNDCG	0.1343	0.1825	0.0251	0.0253	0.0852	0.2851	0.2079	0.6279	0.1936	0.2055

Query No	11	12	13	14	15	16	17	18	19	20
infAP	0.0211	0.0144	0.0254	0.006	0.0126	0.0109	0.047	0.0274	0.0023	0.0491
infNDCG	0.3302	0.1337	0.1892	0.0728	0.1356	0.1436	0.285	0.1428	0.0644	0.7416

Query No	21	22	23	24	25	26	27	28	29	30
infAP	0.008	0.0322	0.0016	0.0158	0.0017	0.0214	0.0096	0.0071	0.125	0.1271
infNDCG	0.0711	0.1313	0.0348	0.4504	0.0782	0.1712	0.1007	0.1475	0.4341	0.334

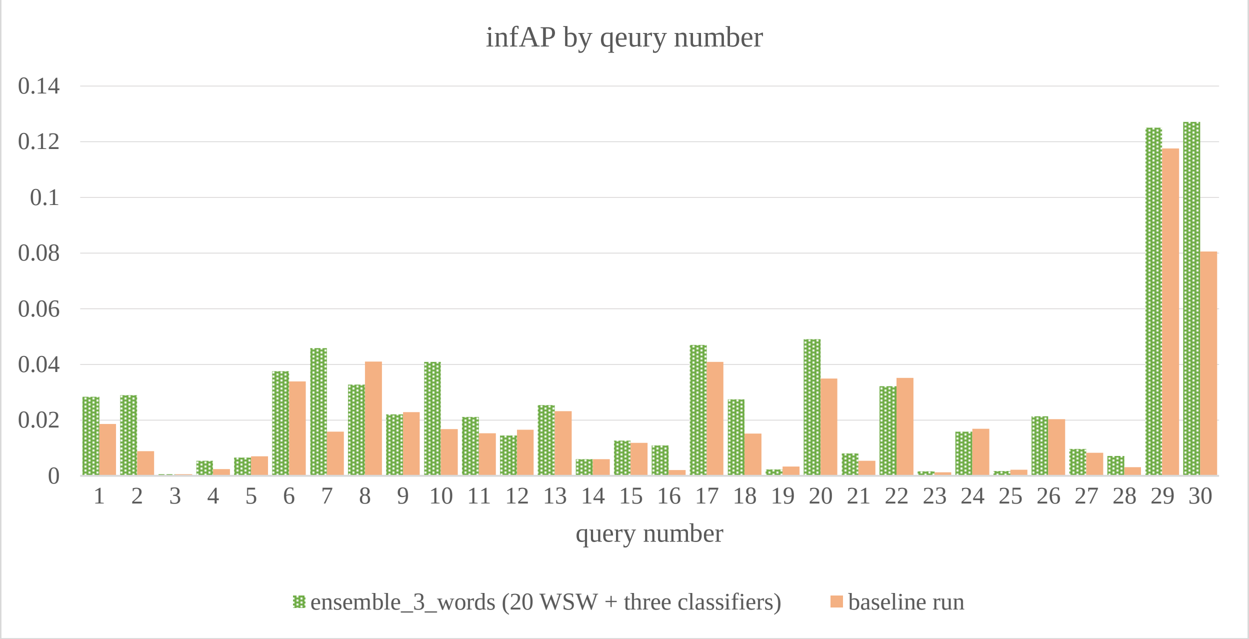

Figure 27. The infAP comparison by query number between the ensemble QE model (WSW) using top 3 words and the baseline run

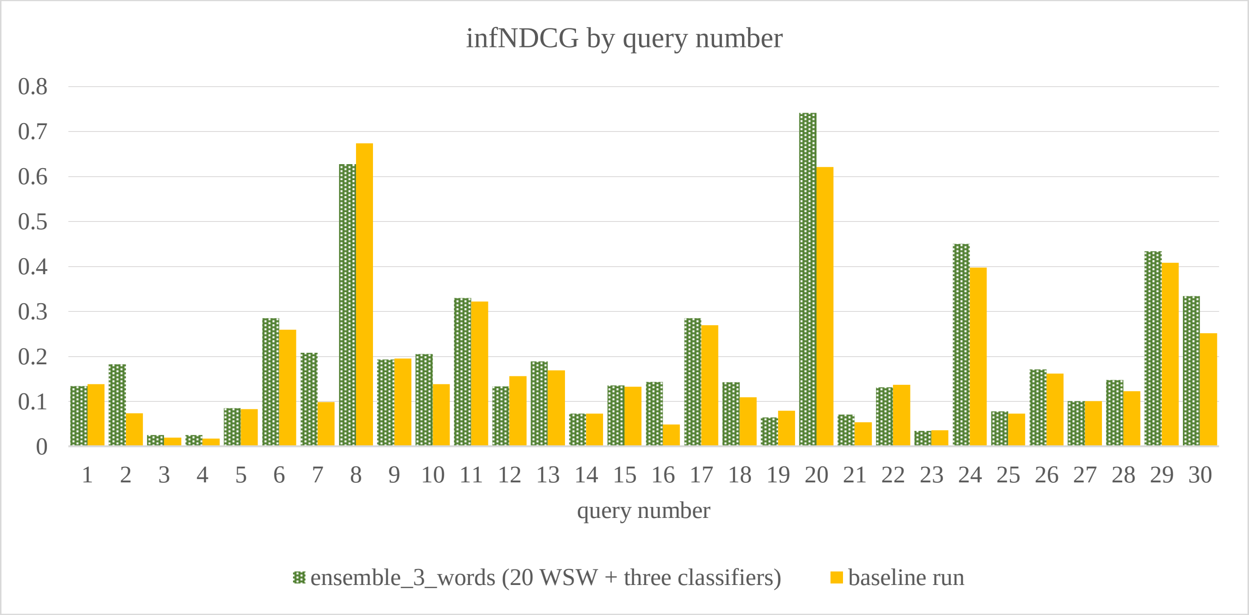

Figure 28. The infNDCG comparison by query number between the ensemble QE model (WSW) using top 3 words and the baseline run

4.4.2 The ensemble of multiple LDA models and ANN classifiers based on Positive Word Selection (PWS)

Candidate words for QE were generated by multiple PWS models. In each PWS (LDA model + one binary ANN classifier) model, positive topic words were selected by an ANN classifier. Top 15 positive words per query from 10 PWS (an LDA model + a binary ANN classifier with 2 layers and 700 nodes per layer) models were ranked by one classifier or four classifiers as follows.

1. Topic words were generated by 10 LDA models where the mean infNDCG scores were relatively good.
2. Those topic words were classified into two groups (the positive word group & the negative word group) by the binary ANN classifier with 2 layers (700 nodes per layer). The word in the positive word group were sorted by the probability estimated for the positive group.
3. A maximum of the top *k* (*k* = 15) positive words per query (30 queries) was selected in each PWS model by the descending order of the probability estimated for the positive group as candidate words for QE: a maximum of 450 (top 15 positive words * 30 queries) words from each PWS model. A maximum of 4500 words (450 words per PWS model * 10 PWS models), but, totally 4268 positive words were collected from 20 PWS models.

- When using one binary classifier with 2 layers and 700 nodes per layer, 4268 positive words were ranked by the descending of the probability for the positive word group without calculating class scores.
- When using three classifiers (one binary classifier with 2 layers and 700 nodes per layer and three 3-class classifiers with 2 layers (500 & 700 nodes per layer) & 3 layers (700 nodes per layer), the class score of a word was given according to the classification by each classifier: 0 for a negative word, 1 for a neutral word, and 2 for a positive word. For word ranking and filtering, 1) the sum of class scores of a word and 2) (the average of four class scores) * (the average of four probabilities for the positive word group), were calculated by four classifiers (three classifiers plus

one classifier included in the PWS model). If the sum of class scores of a word was less than 5, the word was ignored. Of 4268 words, 938 words were ignored. The remaining 3330 words were scored by (the average of four class scores) * (the average of four probabilities for the positive word group) values.

4. The top *k* (*k* = 1…40) words were added to the original query for QE.

Mean infAP and infNDCG scores of two ensemble QE models based on PWS (one classifier vs. multiple classifiers) were compared by the number of the top words added for QE in Figure 29 and 31. Mean infAP and infNDCG scores based on one ANN classifier and four ANN classifiers were listed in Table 30 and 31, respectively. When the top 4 words in the ensemble QE model using multiple classifiers were added to the original query, the performance was most improved (infAP: 0.0254 and infNDCG: 0.1939) while the best performance of the ensemble QE model using one classifier appeared in the query expanded by the top 17 words (infAP: 0.0247 and infNDCG: 0.1906). No new words were added after the expanded queries using 21 words. Ranking and filtering by the probability for the positive word group and class score were effective in generating new queries. All 30 expanded queries using multiple classifiers showed better mean infAP and infNDCG scores than the ensemble QE model using one classifier (Figure 29 and 31).

Table 30. Mean infAP and infNDCG scores of the ensemble QE model using one classifier based on 10 PWS models (an LDA model + one ANN classifier) models for the top 2 retrieved documents

no. words	1	2	3	4	5	6	7	8	9	10
infAP	0.0209	0.0209	0.0209	0.0209	0.0209	0.0209	0.0209	0.0211	0.0234	0.0236
infNDCG	0.1808	0.1808	0.1808	0.1808	0.1808	0.1808	0.1808	0.1809	0.1859	0.1881

no. words	11	12	13	14	15	16	17	18	19	20
infAP	0.0236	0.0236	0.0241	0.0247	0.0246	0.0243	**0.0247**	0.0246	0.0245	0.0245
infNDCG	0.1881	0.1862	0.1886	0.1895	0.1894	0.1903	**0.1906**	0.1904	0.1902	0.1902

no. words	21	22	23	24	25	26	27	28	29	30
infAP	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245

infNDCG	0.1902	0.1902	0.1902	0.1901	0.19	0.19	0.1901	0.1901	0.1902	0.1904

no. words	31	32	33	34	35	36	37	38	39	40
infAP	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245	0.0245
infNDCG	0.1904	0.1904	0.1904	0.1904	0.1904	0.1904	0.1904	0.1904	0.1904	0.1904

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

Table 31. Mean infAP and infNDCG scores of the ensemble QE model using four classifiers based on 10 PWS (an LDA model + four ANN classifiers) models for the top 2 retrieved documents

no. words	1	2	3	4	5	6	7	8	9	10
infAP	0.0229	0.0242	0.0247	**0.0254**	0.0254	0.0254	0.025	0.025	0.025	0.025
infNDCG	0.1884	0.1889	0.191	**0.1939**	0.1926	0.1926	0.1918	0.1918	0.1918	0.1918

no. words	11	12	13	14	15	16	17	18	19	20
infAP	0.025	0.025	0.025	0.025	0.025	0.025	0.025	0.0249	0.0249	0.0249
infNDCG	0.1918	0.1918	0.192	0.192	0.192	0.192	0.192	0.1918	0.1918	0.1918

no. words	21	22	23	24	25	26	27	28	29	30
infAP	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249
infNDCG	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918

no. words	31	32	33	34	35	36	37	38	39	40
infAP	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249	0.0249
infNDCG	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918	0.1918

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

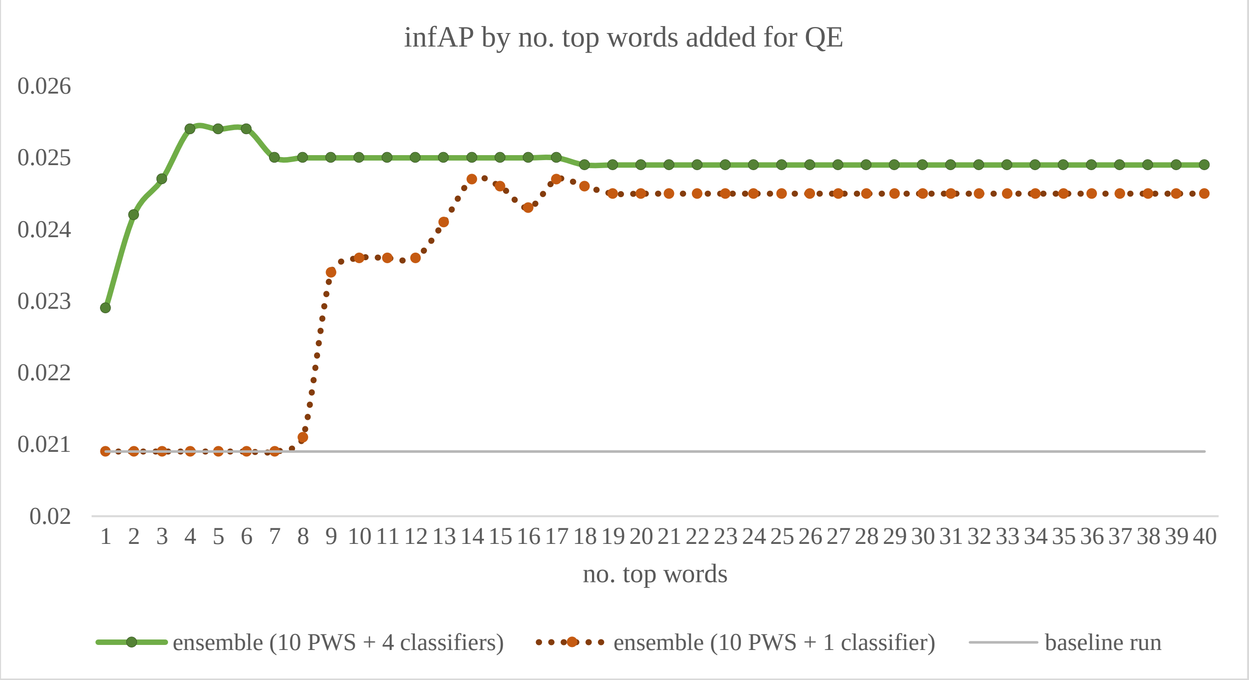

Figure 29. Mean infAP scores in the ensemble QE model (PWS) for the top 2 retrieved documents (four classifiers vs. one classifier)

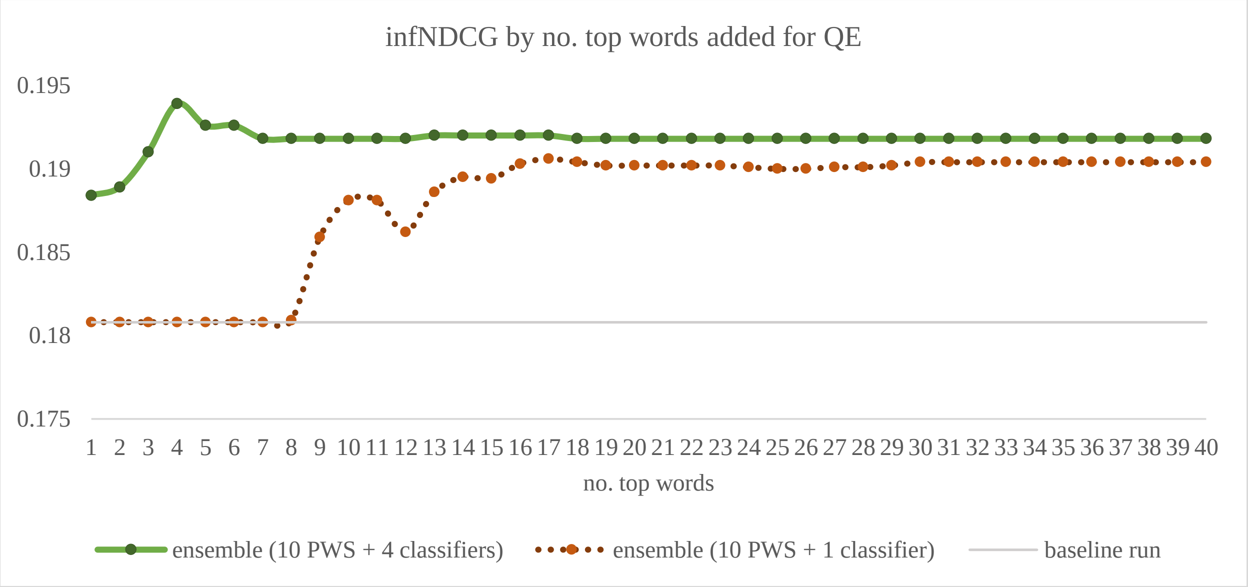

Figure 30. Mean infNDCG scores in the ensemble QE model (PWS) for the top 2 retrieved documents (four classifiers vs. one classifier)

For the best result the ensemble QE model (PWS) expanded by the top 4 words, infAP and infNDCG were compared with the scores of the baseline run for 30 queries in Table 28 & 32 and Figure 31 & 32. There were improvements of infAP and infNDCG in 15 queries (the same scores for 12 queries) and 15 queries (the same scores for 12 queries) of 30 queries, respectively. There were statistically significant differences in the mean infAP and infNDCG scores for 30 queries (paired t-test alpha = 0.05, p-value = 0.0304 for infAP and 0.0266 for infNDCG). The top 4 terms used for QE were described along with the queries in Appendix H.

Table 32. Mean infAP and infNDCG scores of the ensemble QE model (PWS) using top 4 words for 30 queries

Query No	1	2	3	4	5	6	7	8	9	10
infAP	0.0492	0.0128	0.0008	0.0027	0.007	0.0376	0.0158	0.041	0.0229	0.0168
infNDCG	0.2157	0.1153	0.033	0.0185	0.0828	0.2851	0.0985	0.6742	0.1955	0.1387
Query No	11	12	13	14	15	16	17	18	19	20
infAP	0.0122	0.014	0.0232	0.006	0.0202	0.0044	0.0493	0.0152	0.0056	0.0349
infNDCG	0.277	0.1392	0.1688	0.0728	0.1874	0.0791	0.2866	0.1095	0.0905	0.6214
Query No	21	22	23	24	25	26	27	28	29	30
infAP	0.0063	0.0375	0.0012	0.0161	0.0021	0.0238	0.0385	0.0031	0.1176	0.1248
infNDCG	0.066	0.1408	0.0357	0.4111	0.073	0.1765	0.1986	0.088	0.4087	0.3291

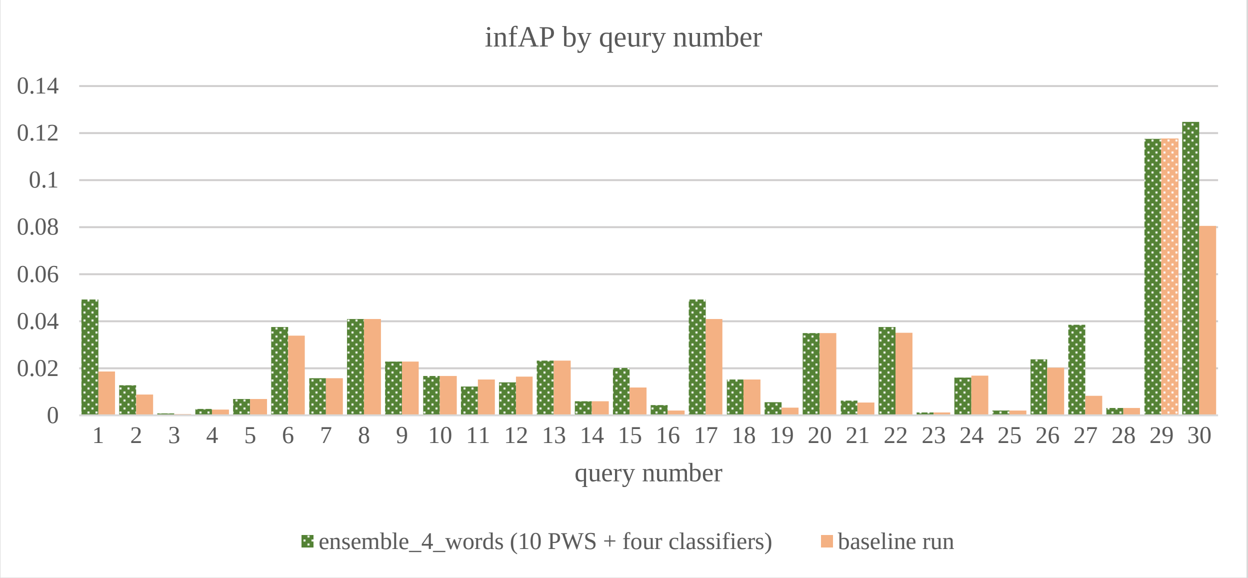

Figure 31. The infAP comparison by query number between the ensemble QE model (PWS) using top 4 words and the baseline run

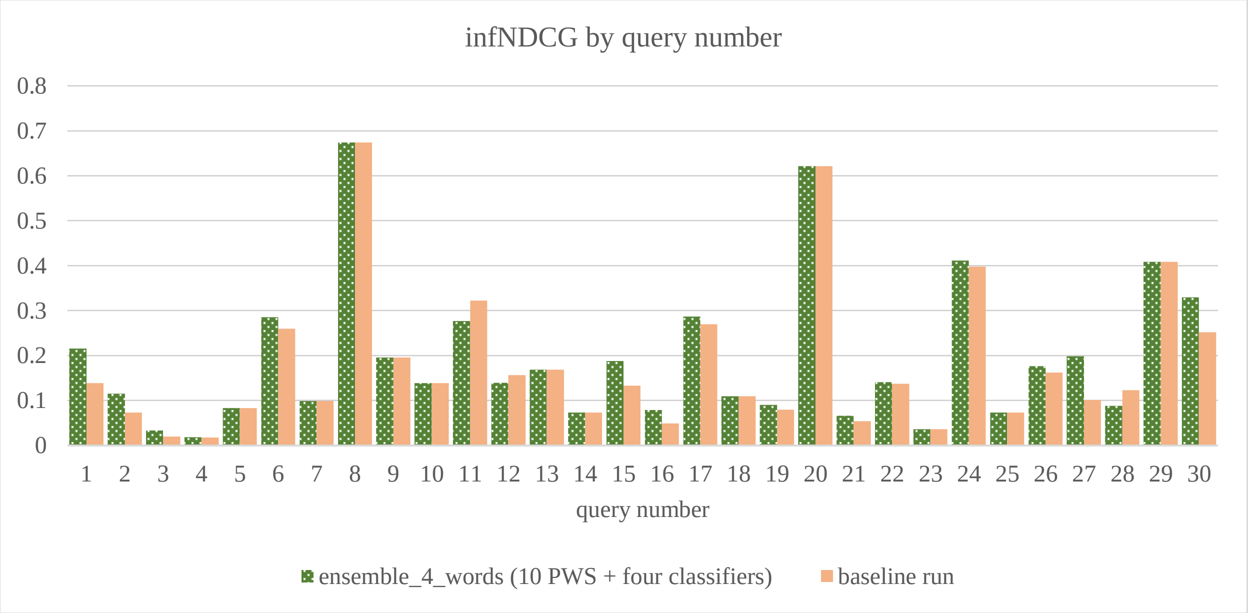

Figure 32. The infNDCG comparison by query number between the ensemble QE model (PWS) using top 4 words and the baseline run

4.5 Summary of Results

4.5.1 RQ1) How effective is the application of LDA topic words based on MeSH terms to QE in health IR?

The average mean infAP and infNDCG scores of the QE models using the LDA models with different threshold values were listed with p-values calculated in two-sample t-tests, comparing with the baseline run (Table 33). The improved results showing a significant difference (alpha = 0.05) are in bold.

Table 33. Average mean infAP and infNDCG scores of the LDA models with different thresholds for TP, WP, or TP * WP for the top1/top2 retrieved documents

Docs ranked	TP	WP	TP*WP	Ave (mean infAP)	Ave (mean infNDCG)	p-value (infAP)	p-value (infNDCG)
top1	0.01	-	-	0.0183	0.1684	6.98E-13	5.88E-11
top2	0.01	-	-	0.0206	0.1768	0.2766	0.0167
top1	0.1	0.03	-	0.0188	0.1633	2.20E-06	3.33E-13
top1	0.1	0.3	-	**0.0213**	0.1819	**0.0135**	0.2813
top2	0.1	0.3	-	0.0201	0.1696	0.0235	2.72E-11
top1	0.08	-	0.03	**0.0213**	0.1819	**0. 0335**	0.0712
top2	0.08	-	0.03	**0.0217**	0.1804	**0.0022**	0.7341

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

The thresholds for TP (0.1 and 0.08), WP (0.03 and 0.3), and TP * WP (0.03) were applied for the top1 retrieved document based on an LDA model with 1700 topics. High infAP and infNDCG scores were observed, such as 0.0277 (TP: 0.15 & WP: 0.02) for infAP and 0.1963 (TP: 0.07 & WP: 0.03) for infNDCG. However, because the threshold values were chosen on a specific condition including an LDA model with a specific number of topics (1700) and top1 retrieved document, they were not effective when applied to other LDA models with different numbers of topics and different numbers of top retrieved documents (e.g. top2).

Although LDA models with specific thresholds for TP, WP, and TP*WP showed overall better mean infAP and infNDCG scores than the scores of 40 LDA model with the default threshold for TP (0.01), the IR performance of each LDA model was not always better in comparison with the baseline run. There were two pairs of thresholds increasing infAP: 1) TP: 0.1 & WP: 0.3 for the top1 retrieved document, 2) TP: 0.08 & TP * WP: 0.03 for the top1/top2 retrieved documents). Three average mean infAP scores of 40 LDA models were statistically significantly better than the infAP score of the baseline run (in bold).

To find more general thresholds, the optimized thresholds from several LDA models based on different conditions (e.g. different numbers of topics and different numbers of retrieved documents) would be compared.

4.5.2 RQ2) How effective is the application of LDA MeSH terms to QE in health IR when LDA topic words are weighted or selected by an ANN classifier?

A binary (2 layers with 700 nodes per layer) and a 3-class (3 layers with 700 nodes per layer) ANN classifier were applied to choose relevant MeSH terms, which were generated by LDA models for 30 queries. An ANN classifier was used to weight original word scores (TP * WP * / (document rank for the word)^{2}) using a probability for the positive/negative/neutral word group (WSW) or select positive words (PWS) in an LDA model. The top *k* words with high weighted word scores or positive words were recommended for QE. Two-sample t-tests were conducted to compare the average mean infAP and infNDCG scores with the scores of the baseline run (Table 34).

Table 34. Average mean infAP and infNDCG scores of QE models based on the WSW/PWS model for the top 2 retrieved documents

Classifier (Weighting/Selection)	Ave (mean infAP)	Ave (mean infNDCG)	p-value (infAP)	p-value (infNDCG)
Binary				
Word Score Weighting @10	0.0231	0.1883	3.49E-11	3.16E-06
Word Score Weighting @7	**0.0234**	**0.1891**	1.10E-20	3.36E-10
Positive Word Selection	0.0218	0.1804	4.79E-08	0.5369
Positive Word Selection @7	**0.0224**	**0.1831**	3.41E-18	1.99E-07

3-Class				
Word Score Weighting @10	0.0217	0.1773	0.0096	0.0110
Positive Word Selection	0.0191	0.1698	3.47E-08	2.16E-13

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

WSW and PWS models based on a binary ANN classifier with 2 layers and 700 nodes per layer were most effective in increasing infAP and infNDCG, statistically, significantly, comparing with the baseline run (two-sample t-test) when the top 7 words were chosen for QE (in bold).

WSW models using the binary ANN classifier showed better performance in increasing average mean infAP and infNDCG scores statistically significantly (alpha = 0.05), p-value =3.49E-11 & 1.10E-20 for infAP and 3.16E-06 & 3.36E-10 for infNDCG) by weighting word scores (a maximum of the top 10 or 7 words). The average mean infAP and infNDCG scores were slightly better when using the top 7 words than the top 10 words. Meanwhile, the 3-class classifier was not as good as the binary classifier, even though the 3-class classifier is helpful to increase the average mean infAP score, statistically, significantly (alpha = 0.05, p-value = 0.0096).

Choosing positive words improved mean infAP scores when using the binary classifier statistically significantly (alpha = 0.05, p-value = 4.79E-08), but not for infNDCG (p-value = 0.5369). Because of poor classifier performance, some positive words might not be helpful to increase infAP and infNDCG. Instead of choosing all positive words, selecting the top 7 positive words by the descending order of the word scores was more effective, which showed statistically significant improvements in the mean scores for the binary classifier (alpha = 0.05, p-value = 3.41E-18 for infAP and 1.99E-07 for infNDCG).

4.5.3 RQ3) How effective are the ensembles of multiple LDA models and ANN classifiers in selecting MeSH terms for QE in health IR?

An ANN classifier was used to weight word scores in the WSW model or select positive words in the PWS model. Each WSW/PWS model recommends the top *k* words with high word scores or positive words for QE. The recommended words from multiple WSW/PWS models were ranked by one ANN classifier or multiple ANN classifiers. Paired t-tests were conducted to see differences in the mean infAP and infNDCG scores for 30 queries between the best results of the ensemble QE models using multiple classifiers and the scores of the baseline run. The best scores of ensemble QE models and p-values for 30 queries were listed in Table 35.

Table 35. Best mean infAP and infNDCG scores of the ensemble QE models based on the WSW/PWS model for the top 2 retrieved documents

Ensemble QE type	Best mean infAP	Best mean infNDCG	p-value @30Qs (infAP)	p-value @30Qs (infNDCG)
Word Score Weighting				
20 WSW models + One classifier	0.0247	0.1953	-	-
20 WSW models + Multiple (3) classifiers	**0.0271**	**0.2055**	0.0050	0.0029
Positive Word Selection				
10 PWS models + One classifier	0.0247	0.1906	-	-
10 PWS models + Multiple (4) classifiers	0.0254	0.1939	0.0304	0.0266

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

Multiple classifiers were more effective to remove irrelevant words and rank words than one classifier. The ensemble QE models using multiple classifiers showed better mean infAP and infNDCG scores for all new queries expanded using the top 30 (WSW) or 40 (PWS) terms than the ensemble QE models using only one classifier. The best results from the ensemble QE models using multiple classifiers also showed statistically significant mean differences in infAP and infNDCG scores for 30 queries (alpha = 0.05), comparing with the scores of the baseline run.

Although the ensemble QE models based on Word Score Weighting showed better performance, the ensemble QE models based on Positive Word Selection showed the potential to increase infAP and infNDCG. Word filtering and ranking by ensemble QE models were effective in identifying relevant words.

Chapter 5 DISCUSSION

5.1 The LDA Model Evaluation

LDA models have a various number of topics. How many topics are relevant? Although the number of topics would be dependent on the purpose of research, generally the topic number is decided by some metrics, such as perplexity, coherence, etc. The cost of generating an LDA model with lots of topics might be high if the data size is huge. It might take several days and need lots of memory (e.g. RAM). For instance, in this study, it took around 20 days to generate an LDA model with 4000 topics, so a cluster with lots of CPUs was used to 40 LDA models.

5.1.1 The Number of Topic on LDA for IR – Perplexity

The relationship between the model fit and IR performance is one concern in this study. The best K (the number of topics) decided by the model fit measure might be most effective in selecting words for QE, which would improve infAP and infNDCG. Perplexity was measured to evaluate the LDA model fit for the models with different numbers of topics. The validation dataset, randomly selected 20% of documents, was used to compare the perplexity of the models. The training dataset, 80% of data, was used to generate LDA models.

Wei and Croft (2006) compared the retrieval results on 242,918 Associated Press newswire documents (1988-90) for LDA models with different numbers of topics (K) in terms of AP (average precision). The LDA model with K=800 showed the best average precision. Meanwhile, in Liu and Croft's research (2004), the best number of K was 2000 in the cluster-based retrieval using hierarchical agglomerative clustering algorithms for both datasets (Associated Press newswire 1988−90: 242,918 documents & Federal Register 1988–89: 45,820 documents).

Even though perplexity is a measure to decide the best number (K) of topics for an LDA model, there is no clear conclusion about how related perplexity is to IR performance when LDA topic words are used for QE. To find out the relationship between perplexity and (infAP & infNDCG), perplexity was calculated for the LDA models with different numbers of topics (Figure 33). Randomly selected 80% and 20% of the dataset were used for a training set and a test set. The best *k* with the lowest perplexity (76.074) was 10. The mean infAP and infNDCG scores of the LDA model with 10 topics (the default TP threshold = 0.01) were 0.0199 and 0.1637 for the top1 retrieved document and 0.0209 and 0.1806 in the LDA model with thresholds for TP (0.08), TP*WP (0.03). Compared with the other LDA models (Table 7 and Table 17), mean infAP and infNDCG scores were not high. Overall, LDA models with a relatively large number of topics showed better infAP and infNDCG scores.

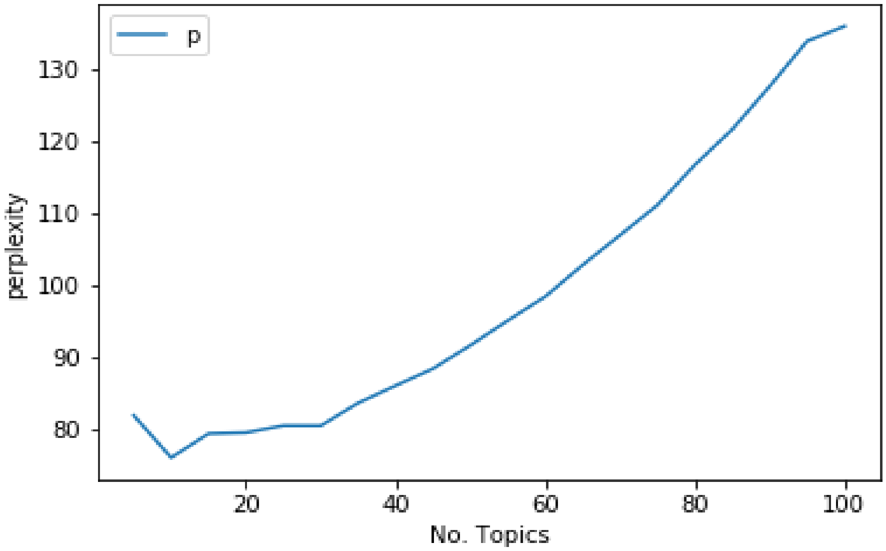

Figure 33. The perplexity for LDA models with different numbers of topics

5.2 Classifier Performance

A classifier played a critical role to identify relevant words for QE. Relevant features and appropriate parameters (the number of layers and nodes, iterations, batch size, etc.) as well as enough data, decide the performance of a classifier. Adjusting parameter values by testing the performance using validation sets is a repeated process to develop a decent classifier. Some issues for constructing classifiers were raised, which affected infAP and infNDCG.

5.2.1 Overfit

Generally, many layers and nodes are helpful to increase accuracy for a training set, however, which does not guarantee better scores on validation and test sets (overfit). The overall ANN classifiers with many layers and nodes showed high accuracy for training datasets but did not show high accuracy for the validation sets (Table 4 & 5), which implies overfitting. The relevant number of layers and nodes should be decided by testing the accuracy of the validation sets. Dropout (Hinton, Krizhevsky, Sutskever, & Srivastava, 2019) and early stopping (Yao, Rosasco, & Caponnetto, 2007) are applicable techniques to preventing overfitting in training classifiers. Dropout as a regularization technique limits the number of input data in training, which just accepts a part of input data to prevent overfitting. Early stopping rule can be applied to limit the iteration number of training. If the performance does not improve, the training process stops.

5.2.2 Imbalanced classification

Another problem is skewed classification in binary classification. The binary classifiers classified most words into the negative word group. Although there were more negative words about three times, most classifiers grouped 90% of the words in the validation sets into the negative word group, except one classifier with 3 layers including 700 nodes per layer.

F1 and AUC scores on the validation sets were calculated to overcome this weakness of accuracy measure. Classifiers trained on more than 3 layers showed relatively high F1 and AUC scores (Table 4 & 5). To overcome the weakness of imbalanced classification, the probability for a specific (positive/negative) class was used for weighing a word score instead of using the output class (label).

5.2.3 ANN vs. other classifiers

Even though ANN classifiers have shown good performance generally, other classifiers based on different algorithms, such as SVM, decision tree, naïve Bayes, logistic regression, or k-means, can outperform an ANN classifier. As an example, an SVM classifier was compared with an ANN classifier in Appendix I.

Instead of ANN classifiers, other classifiers would be more effective when they are incorporated with LDA models. Some classifiers would be more effective for filtering; others would be more effective for ranking. The combination of different types of classifiers would lead to the best ensemble QE model.

5.3 A Cost-effective IR System

Normally, a more cost/investment results in better performance, however, a reasonable amount of input cost must be considered in practice because more input units are needed to improve the same amount of performance when IR performance is beyond a specific threshold in many cases. A compact but well-performing, and efficient IR system should be designed with reasonable cost and effort unless an IR system with very high performance is not necessary.

5.3.1 The number of vocabulary words

Document representation gives huge impacts on not only IR performance but also costs in implementing an IR system. In this study, MeSH terms including 24,883 n-gram words were considered to represent a document. Some MeSH terms barely or frequently appear. Those words might be ignored for pre-processing efficiency if the collection size is too huge. MeSH terms barely appeared might not that influential in IR. MeSH terms frequently occurred would be likely to be general terms, which may not critical in IR.

MeSH descriptors include a list of Check Tags that are very general (e.g. "Humans"). Check Tags are mostly used for filtering search results. Although Check Tags were not removed in this study, they would be removed for both effectiveness and efficiency.

5.3.2 The number of topic models and classifiers

In designing ensemble QE models, the number of models is important as much as the quality of models, which affect IR performance. Even if topic models or classifiers are homogeneous, QE using more topic models and classifiers would derive better performance. However, when resources are limited, the reasonable numbers of LDA topic models and classifiers would be decided according to how much IR performance is improved by one inputted cost unit. Also, the complexity of an IR system affects IR speed and maintenance. The more complicated the IR system is, the more resources would be required and the slower IR speed would be. The reasonable numbers of topic models and classifiers would be different according to domain areas.

Chapter 6 CONCLUSION

The PMC 2016 snapshot including 1,451,661 documents was used to generate LDA models. Full-text documents in the health domain were represented by MeSH terms assuming that the professional terminology would be more helpful for QE to increase the performance in health IR.

LDA topic models generated topic words (MeSH terms) using a query or retrieved documents by the query. Because generated topic words include many irrelevant words for QE, selecting relevant words is the key point to increase the IR performance. Setting up thresholds for topic probability (TP), word probability (WP), or (TP * WP) can filter out negative words for QE. Although thresholds values for filtering words were effective to increase infAP and infNDCG scores on several individual LDA models, one problem is that optimized thresholds for an individual LDA model did not function well in other LDA models with different numbers of topics.

An ANN classifier solves this problem by predicting the relevance of a word for QE. Multiple (binary and 3-class) ANN classifiers were designed to judge whether topic words (MeSH terms) were positive/negative/neutral for QE. Positive words increase infAP and infNDCG scores when they are added to the original query, while negative words decrease the scores. Neutral words give no impact on the scores. 424,288 MeSH terms, which were generated by 40 LDA models for the top 10 retrieved documents, were used for training ANN classifiers. The evaluation set provided by the 2016 TREC CDS track was employed in evaluating the terms.

ANN classifiers were trained on LDA/collection-related features. Most features showed differences in the mean values of the features statistically significantly (alpha = 0.05).

In the proposed QE models based on Word Score Weighting (WSW) and Positive Word Selection (PWS), an ANN classifier was integrated with an individual LDA model to 1) give weight to the word score using the probability estimated for the positive word group (WSW) or 2) to identify positive words (PWS).

40 WSW/PWS models showed improved the average mean infAP and infNDCG scores. The top *k* (e.g. 7) MeSH terms selected by a binary classifier based on both approaches were helpful in increasing mean infAP and infNDCG scores statistically significantly (alpha = 0.05) comparing with the mean scores of the baseline run.

Ensemble models using multiple types of data/models/algorithms/techniques have shown better performance in IR than individual models. The weakness of an individual model can be complemented by other models, general ensemble IR models based on multiple models show stable performance.

Ensemble QE models using multiple LDA models and ANN classifiers showed high IR performance in terms of infAP and infNDCG in health IR. Candidate topic words (MeSH terms) were recommended by multiple WSW/PWS models. And then candidate terms were ranked by one classifier or multiple classifiers. Multiple classifiers were employed to 1) remove negative words and 2) rank the words using the classification and the probability of being a positive word, while one classifier only ranks candidate words. The ensemble QE models using multiple classifiers showed better infAP and infNDCG scores. The best results from the ensemble QE models showed statistically significant improvements in the mean infAP and infNDCG scores for 30 queries (alpha = 0.05) comparing with the baseline result.

The proposed ensemble QE models showed how the integration of multiple LDA models and ANN classifiers can enhance IR performance. Ensemble QE models using multiple LDA models and ANN classifiers based on MeSH terms, showed the potential to improve health IR performance in terms of infAP and infNDCG. If the ANN classifiers can be designed based on more data and effective features, the ensemble QE models would play a key role to improve IR systems. The application of ensemble QE models based on various types of models would guarantee stable search results in the health IR.

6.1 Limitations

Limitations of this study can be discussed methodologically, theoretically, and practically.

6.1.1 Methodological Limitations

The main limitations in the methodological perspective are the absence of data and method triangulation regarding data collection, terminology, qualitative LDA model evaluation, and so on.

6.1.1.1 Data triangulation (collection/terminology scope)

In this study, only academic publications were used through MeSH. Journal articles are usually focused on research rather than real-life needs (i.e. consumer's interest). The document representation using MeSH would reflect the expert point of views rather than consumers. Although search results were generated based on full-text articles, LDA models were generated based on short text including only MeSH terms in documents due to pre-processing efficiency.

Other kinds of collections, social media data, such as YahooAnswers Health-related data might be used to compare different types of terminology: user-generated terms vs. expert terms (MeSH) or journal papers vs. social Q&A. YahooAnswers data can be crawled using general scraping APIs (e.g. Python QA-scrapers, https://github.com/collab-uniba/qa-scrapers). PubMed abstracts, or practical text like clinical trial descriptions, which is provided by ClinicalTrials.gov, might be selected as additional data.

PMC data consists of articles in open access journals. Some traditional journals requiring subscriptions are showing higher impact factors in health information (Björk & Solomon, 2012). The health topics based on open access journals might not cover overall topics of health information in the academic field.

6.1.1.2 Method triangulation (LDA model evaluation)

LDA models have been used widely over a decade, the reliability of generated topics has been discussed in terms of qualitatively as well as quantitatively. In this study, the reliability and validity were discussed quantitatively using perplexity and topic consistency although it was not that related to IR performance. Qualitative approaches based on human interpretations might give another insight if conducted.

LDA models can be implemented in two ways (Rosen-Zvi, Griffiths, Steyvers, & Smyth, 2004): variational EM (Blei et al., 2003) and Gibbs sampling (Griffiths & Steyvers, 2004). The performance might be different between the two kinds of algorithms. The relationship between IR performance and different topic models might be studied further.

6.1.1.3 Miscellany

Word interaction. Interesting interactions between words for QE were observed in IR in a few cases. QE using a negative word would show low infAP and infNDCG scores. However, when the negative word is added to the original query along with other terms, the negative term can help increase infAP or infNDCG scores. For example, *fosfomycin* is a negative term for the first query, "*A 78 year old male presents with frequent stools and melena*". When *fosfomycin* is added to the query text, infAP and infNDCG scores decreased: from 0.0186 to 0.0119 (infAP) and from 0.1388 to 0.1148 (infNDCG). Another term, *double-balloon enteroscopy* increased the scores: from 0.0186 to 0.0221 (infAP) and from 0.1388 to 0.1540 (infNDCG). When two terms were used for QE together, interesting scores were generated for infNDCG. Although the infAP score decreased slightly from 0.0221 (*double-balloon enteroscopy*) to 0.0218 (*double-balloon enteroscopy fosfomycin*), the infNDCG score increased from 0.1540 to 0.1633. The word interaction for QE would be explored in a further study.

LDA model stability in topic word distribution. In the very rare cases, the LDA model based on the python module, *gensim*, generated different top 10 words which did not affect that much the measurement of infAP and infNDCG even though measurement reliability would decrease. Maybe some words might have the same word probability value.

6.1.2 Theoretical/practical contributions

Some proposed concepts, such as CTD (Collection Topic Density) and CTF (Collection Topic Frequency), would be incorporated into the LDA model as important features as TP and WP. Similar concepts to IDF, inverse CTD or CTF would be studied more for IR improvement, which might generate an LDA variation like topic weighting LDA models by CTD or CTF.

Also, the implementation of the proposed concepts related to topic weighting into existing LDA-related modules, such as *gensim*, might be another future project.

6.2 Further Studies

The ensemble of multiple LDA models and classifiers (binary & 3-class classifiers) showed the potential to improve IR performance in the health domain. The performance of the classifiers is critical to select effective words. More effective features would be integrated into the existing features and more data including more queries and training data would enhance the performance of the classifiers.

Using journal topics is helpful to improve IR performance. A collection can be divided according to journals assuming that there are journal articles enough to generate topics. Query topics and journal topics would be compared to decide the search scope (extension or shrinking). How to apply the journal topics to IR in health information might be different according to a specific area. This is a kind of combination of query-based IR and browsing. In addition, the relationship between topics can be identified using variation models of LDA. Approaches based on different types of units (character vs. sentence and structure vs. semantic) from bag-of-words may give another insight. Those approaches would not be limited to LDA. Other machine learning methods like deep learning might show more effective classification and clustering results.

Scholars should find relevant journals to publish their articles, which might be hard for novice scholars to read. Designing a prediction system for a given document is useful for scholars to find more appropriate journals related to the document topic, which might be used to decide which journal looks proper to publish the paper. If the system can give a numerical degree/score of how a manuscript is acceptable to a journal in terms of topic match, scholars might use the system in reviewing the content of the paper by comparing topics between the manuscript and the journal.

REFERENCES

Alghamdi, R., & Alfalqi, K. (2015). A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, *6*(1).

Asuncion, A., Welling, M., Smyth, P., & Teh, Y. W. (2009, June). On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (pp. 27–34). AUAI Press.

Azad, H. K., & Deepak, A. (2019). Query expansion techniques for information retrieval: A survey. *Information Processing & Management*, *56*(5), 1698–1735.

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. *ICWSM*, *8*, 361–362.

Bates, M. J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online review*, *13*(5), 407–424.

Beall, J. (2008). The weaknesses of full-text searching. *The Journal of Academic Librarianship*, *34*(5), 438–444.

Bedrick, S., Edinger, T., Cohen, A., & Hersh, W. (2012). *Identifying patients for clinical studies from electronic health records: TREC 2012 medical records track at OHSU*.

Belkin, N. J., & Croft, W. B. (1987). Retrieval techniques. *Annual review of information science and technology*, *22*, 109–145.

Bergamaschi, S., Po, L., & Sorrentino, S. (2014, April). Comparing Topic Models for a Movie Recommendation System. In *WEBIST (2)* (pp. 172–183).

Björk, B. C., & Solomon, D. (2012). Open access versus subscription journals: a comparison of scientific impact. *BMC medicine*, *10*(1), 73.

Blei, D. M., & Lafferty, J. D. (2006, June). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning* (pp. 113–120). ACM.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *the Journal of machine Learning research*, *3*, 993–1022.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, *2008*(10), P10008.

Brisebois, R., Abran, A., Nadembega, A., & N'techobo, P. (2017). A semantic metadata enrichment software ecosystembased on machine learning to analyze topic, sentiment and emotions. *International Journal of Scientific Research in Science Engineering and Technology (IJSRSET)*, *8*(4), 16698–16714. doi: http://dx.doi.org/10.24327/ijrsr.2017.0804.0200

Bompada, T., Chang, C. C., Chen, J., Kumar, R., & Shenoy, R. (2007, July). On the robustness of relevance measures with incomplete judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 359–366).

Buckley, C., & Voorhees, E. M. (2004, July). Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 25–32).

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, *44*(1), 1–50.

Chang, Y., Ounis, I., & Kim, M. (2006). Query reformulation using automatically generated query concepts from a document space. *Information processing & management*, *42*(2), 453–468.

Chen, Y., Zhang, P., Song, D., & Wang, B. (2015, October). A Real-Time Eye Tracking Based Query Expansion Approach via Latent Topic Modeling. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (pp. 1719–1722). ACM.

Chrislb, (2005). Artificial Neuron Model, In *Wikipedia*, *The Free Encyclopedia*. Retrieved from https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png.

Choudhury, M., Lin, Y. R., Sundaram, H., Candan, K. S., Xie, L., &Kelliher, A. (2010). How does the data sampling strategy impact the discovery of information diffusion in social media?.*ICWSM*, *10*, 34–41.

Christopher, C. (2010). *Encyclopaedia Britannica: definition of data mining*. Retrieved from https://www.britannica.com/technology/data-mining.

Cooper, W. S. (1973). On selecting a measure of retrieval effectiveness. *Journal of the American Society for Information Science*, *24*(2), 87–100.

Cox, K. (1992, November). Information retrieval by browsing. In *Proceedings of The 5th International Conference on New Information Technology, Hongkong*.

Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), 391.

Diaz, F., Mitra, B., & Craswell, N. (2016). Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*.

Díaz-Galiano, M. C., García-Cumbreras, M. Á., Martín-Valdivia, M. T., Montejo-Ráez, A., & Ureña-López, L. A. (2007, September). Integrating mesh ontology to improve medical information retrieval. In *Workshop of the Cross-Language Evaluation Forum for European Languages* (pp. 601–606). Springer, Berlin, Heidelberg.

Efthimiadis, E. N. (1996). Query Expansion. *Annual review of information science and technology (ARIST)*, *31*, 121–87.

Falagas, M. E., Kouranos, V. D., Arencibia-Jorge, R., & Karageor Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37.

Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press.

Ferilli, S. (2011). Information Management. In *Automatic Digital Document Processing and Management: Problems, Algorithms and Techniques*. Springer Science & Business Media.

Figueiredo, F., Rocha, L., Couto, T., Salles, T., Gonçalves, M. A., & Meira Jr, W. (2011). Word co-occurrence features for text classification. *Information Systems*, *36*(5), 843–858.

Gibbs, J. W. (1902). *Elementary principles in statistical mechanics: developed with especial reference to the rational foundation of thermodynamics*. C. Scribner's sons.

Good, I. J. (1956). Some terminology and notation in information theory. *Proceedings of the IEE-Part C: Monographs*, *103*(3), 200–204.

Goodwin, T., & Harabagiu, S. M. (2014). *UTD at TREC 2014: Query expansion for clinical decision support*. Texas Univ. at Dallas Richardson.

Gopalan, P. K., Charlin, L., & Blei, D. (2014). Content-based recommendations with poisson factorization. In *Advances in Neural Information Processing Systems* (pp. 3176–3184).

Griffiths, T. L., Jordan, M. I., Tenenbaum, J. B., & Blei, D. M. (2004). Hierarchical topic models and the nested chinese restaurant process. In *Advances in neural information processing systems* (pp. 17–24).

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, *101*(suppl 1), 5228–5235.

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political analysis*, *21*(3), 267–297.

Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. (1996). *Neural network design* (Vol. 20). Boston: PWS publishing company.

Han, J., Pei, J., &Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Harris, Z. S. (1954). Distributional structure. *Word,10*(2/3): 146–62.

Hawking, D. (2000, November). Overview of the TREC-9 Web Track. In *TREC*.

Hellinger, E. (1909). Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, *136*, 210–271.

Heo, G. E., Kang, K. Y., Song, M., & Lee, J. H. (2017). Analyzing the field of bioinformatics with the multi-faceted topic modeling technique. *BMC bioinformatics*, *18*(7), 251.

Hiemstra, D. (2009). Information retrieval models. *Information Retrieval: searching in the 21st Century*, 2–19.

Hinton, G. E., Krizhevsky, A., Sutskever, I., & Srivastava, N. (2019). *U.S. Patent No. 10,366,329*. Washington, DC: U.S. Patent and Trademark Office.

Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences of the United States of America*, *102*(46), 16569.

Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent Dirichlet allocation. In *advances in neural information processing systems* (pp. 856–864).

Hofmann, T. (1999, August). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 50–57). ACM.

Hornik, K., & Grün, B. (2011). topic models: An R package for fitting topic models. *Journal of Statistical Software*, *40*(13), 1–30.

Hruschka, H., & Natter, M. (1999). Comparing performance of feedforward neural nets and K-means for cluster-based market segmentation. *European Journal of Operational Research*, *114*(2), 346–353.

Huang, P. S., He, X., Gao, J., Deng, L., Acero, A., & Heck, L. (2013, October). Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management* (pp. 2333–2338). ACM.

Hughes, M., Li, I., Kotoulas, S., & Suzumura, T. (2017). Medical text classification using convolutional neural networks. *Stud Health Technol Inform*, *235*, 246–250.

Ibrahim, Z., & Rusli, D. (2007, September). Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression. In *21st Annual SAS Malaysia Forum, 5th September*.

Jaccard, P. (1901). *Etude comparative de la distribution floraledansune portion des Alpeset du Jura*. Impr. Corbaz.

Jacobi, C., van Atteveldt, W., & Welbers, K. (2016). Quantitative analysis of large amounts of journalistic texts using topic modelling. *Digital Journalism*, *4*(1), 89–106.

Joo, S., Choi, I., & Choi, N. (2018). Topic Analysis of the Research Domain in Knowledge Organization: A Latent Dirichlet Allocation Approach. *Knowledge Organization*, *45*(2).

Kagolovsky, Y., & Mohr, J. R. (2001). A new approach to the concept of" relevance" in information retrieval (IR). *Studies in health technology and informatics*, (1), 348–352.

Karami, A. (2015). *Fuzzy Topic Modeling for Medical Corpora* (Doctoral dissertation), University of Maryland, Baltimore County.

Karimzadehgan, M., & Zhai, C. (2010, July). Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (pp. 323–330). ACM.

Krestel, R., Fankhauser, P., &Nejdl, W. (2009, October). Latent Dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems* (pp. 61–68). ACM.

Kuzi, S., Shtok, A., & Kurland, O. (2016, October). Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 1929–1932).

Lafferty, J. D., & Blei, D. M. (2006). Correlated topic models. In *Advances in neural information processing systems* (pp. 147–154).

Lambiotte, R., Delvenne, J. C., & Barahona, M. (2008). Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*.

Lancaster, F.W., Fayen, E.G. (1973), *Information Retrieval On-Line,* Melville Publishing Co., Los Angeles, California

Lavrenko, V., & Croft, W. B. (2001, September). Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 120–127). ACM.

Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188–1196).

Leaman, R., Khare, R., & Lu, Z. (2013). NCBI at 2013 ShARe/CLEF eHealth Shared Task: disorder normalization in clinical notes with DNorm. In *Proceedings of the CLEF 2013 Evaluation Labs and Workshop*, Valencia, Spain.

Leskovec, J., Rajaraman, A., & Ullman, J. (2011). *Mining of Massive Datasets*. Cambridge University Press. Retrieved from http://infolab.stanford.edu/~ullman/mmds/book.pdf

Li, W., & McCallum, A. (2006, June). Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning* (pp. 577–584). ACM.

Liu, H., & Singh, P. (2004). ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal*, *22*(4), 211–226.

Liu, X., & Croft, W. B. (2004, July). Cluster-based retrieval using language models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 186–193). ACM.

Liu, X., Gao, J., He, X., Deng, L., Duh, K., & Wang, Y. Y. (2015). Representation learning using multi-task deep neural networks for semantic classification and information retrieval.

Lopes, C. T. (2008). *Health Information Retrieval: State of the art report.* Faculdade de Engenharia da Universidade do Porto. Retrieved from http://www.carlalopes.com/pubs/Lopes_SOA_2008.pdf

Lu, K., & Wolfram, D. (2012). Measuring author research relatedness: A comparison of word-based, topic-based, and author cocitation approaches. *Journal of the Association for Information Science and Technology*, *63*(10), 1973–1986.

Lu, Z., Kim, W., & Wilbur, W. J. (2009). Evaluation of query expansion using MeSH in PubMed. *Information retrieval*, *12*(1), 69–80.

Lu, Z., & Li, H. (2013). A deep architecture for matching short texts. In *Advances in neural information processing systems* (pp. 1367–1375).

Lupu, M., Zhao, J., Huang, J., Gurulingappa, H., Fluck, J., Zimmermann, M., ... & Tait, J. (2011, November). Overview of the TREC 2011 Chemical IR Track. In *TREC*.

MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 14, pp. 281–297).

Mann, G. S., Mimno, D., & McCallum, A. (2006, June). Bibliometric impact measures leveraging topic analysis. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries* (pp. 65–74). ACM.

Manning, C.D., Raghavan, P., & Schütze, H. (2008). *An Introduction to Information Retrieval*. Cambridge University Press. Retrieved from http://www-nlp.stanford.edu/IR-book/

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. Retrieved from http://mallet.cs.umass.edu

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115–133.

Merabti, T., Letord, C., Abdoune, H., Lecroq, T., Joubert, M., & Darmoni, S. J. (2009). Projection and inheritance of SNOMED CT relations between MeSH terms. In *MIE* (pp. 233–237).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Mingers, J., & Leydesdorff, L. (2015). A review of theory and practice in scientometrics. *European Journal of Operational Research*, *246*(1), 1–19.

Moghadasi, S. I., Ravana, S. D., & Raman, S. N. (2013). Low-cost evaluation techniques for information retrieval systems: A review. *Journal of Informetrics*,*7*(2), 301–312.

Mu, X., Lu, K., & Ryu, H. (2014). Explicitly integrating MeSH thesaurus help into health information retrieval systems: An empirical user study. *Information Processing & Management*, *50*(1), 24–40.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807–814).

Natsev, A., Haubold, A., Tešić, J., Xie, L., & Yan, R. (2007, September). Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *Proceedings of the 15th ACM international conference on Multimedia* (pp. 991-1000).

Newman, M. E. (2008). The mathematics of networks. *The new palgrave encyclopedia of economics*, *2*(2008), 1–12.

Nguyen, D. Q. (2015). jLDADMM: A Java package for the LDA and DMM topic models.

NIH. (2015). *PMC FAQs*. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/about/faq/

NIH. (2018). *Health Information*. Retrieved from https://www.nlm.nih.gov/hinfo.html#GH

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311–318). Association for Computational Linguistics.

Paul, M. J., & Dredze, M. (2014). Discovering health topics in social media using topic models. PloS one, 9(8), e103408.

Phan, X. H., Nguyen, L. M., & Horiguchi, S. (2008, April). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web* (pp. 91–100). ACM.

Proffitt, E. (2016). TopicModelsVB.jl [Computer software]. Retrieved from https://github.com/esproff/TopicModelsVB.jl

Ponte, J. M., & Croft, W. B. (1998, August). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 275–281). ACM.

Quinn, K. M., Monroe, B. L., Colaresi, M., Crespin, M. H., & Radev, D. R. (2010). How to analyze political attention with minimal assumptions and costs. *American Journal of Political Science*, *54*(1), 209–228.

Ramage, D., & Rosen, E. (2009). Stanford Topic Modeling Toolbox [Computer software]. Retrieved from https://nlp.stanford.edu/software/tmt/tmt-0.4

Rao, Y., & Li, Q. (2012, December). Term weighting schemes for emerging event detection. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology* (Vol. 1, pp. 105–112). IEEE.

Rehurek, R. (2013). Asymmetric LDA Priors, Christmas Edition [blog]. Retrieved from https://rare-technologies.com/python-lda-in-gensim-christmas-edition/#

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Roberts, K., Simpson, M. S., Voorhees, E. M., & Hersh, W. R. (2015, November). Overview of the TREC 2015 Clinical Decision Support Track. In *TREC*.

Roberts, K., Demner-Fushman, D., Voorhees, E. M., & Hersh, W. R. (2016, November). Overview of the TREC 2016 Clinical Decision Support Track. In *TREC*.

Roberts, K., Demner-Fushman, D., Voorhees, E. M., Hersh, W. R., Bedrick, S., Lazar, A. J., & Pant, S. (2017, November). Overview of the TREC 2017 Precision Medicine Track. In *TREC*.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. *Nist Special Publication*, 109–126.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004, July). The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence* (pp. 487–494). AUAI Press.

Salton, G., & Yang, C. S. (1973). On the specification of term values in automatic indexing. *Journal of documentation*, *29*(4), 351–372.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

Salton, G., Fox, E. A., & Voorhees, E. (1985). Advanced feedback methods in information retrieval. *Journal of the American Society for Information Science*, *36*(3), 200–210.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), 210–229.

Saracevic, T. (1976). Relevance: A review of the literature and a framework for thinking on the notion in information science. In *Eds.), Advances in Librarianship 6*.

Saracevic, T. (2007). Relevance: A review of the literature and a framework for thinking on the notion in information science. Part III: Behavior and effects of relevance. *Journal of the American Society for Information Science and Technology*, *58*(13), 2126–2144.

Song, Y., Yan, R., Li, X., Zhao, D., & Zhang, M. (2016). Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149*.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, *28*(1), 11–21.

Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, *427*(7), 424–440.

Strauss, A. L., & Glaser, B. G. (1965). Awareness of dying. *Chicago, Adline*.

Talja, S., Keso, H., & Pietiläinen, T. (1999). The production of ‘context’in information seeking research: a metatheoretical view. *Information Processing & Management*, *35*(6), 751–763.

Tuarob, S., Tucker, C. S., Salathe, M., & Ram, N. (2014). An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages. *Journal of biomedical informatics*, *49*, 255–268.

Voorhees, E. M. (2014, July). The effect of sampling strategy on inferred measures. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (pp. 1119–1122).

Vorontsov, K. V. (2014, May). Additive regularization for topic models of text collections. In *Doklady Mathematics* (Vol. 89, No. 3, pp. 301–304). Pleiades Publishing.

Vorontsov, K. V. (2015). BigARTM [Computer software]. Retrieved from http://docs.bigartm.org.

Vorontsov, K., & Potapenko, A. (2014, April). Tutorial on probabilistic topic modeling: Additive regularization for stochastic matrix factorization. In *International Conference on Analysis of Images, Social Networks and Texts_x000D_* (pp. 29–46). Springer, Cham.

Wallach, H. M., Mimno, D. M., & McCallum, A. (2009). Rethinking LDA: Why priors matter. In *Advances in neural information processing systems* (pp. 1973–1981).

Wang, Y., Rastegar-Mojarad, M., Elayavilli, R. K., Liu, S., & Liu, H. (2016). An Ensemble Model of Clinical Information Extraction and Information Retrieval for Clinical Decision Support. In *TREC*.

Wei, X., & Croft, W. B. (2006, August). LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 178–185). ACM.

World Health Organization. (2008). Framework and standards for country health information systems.

Xu, J., & Croft, W. B. (2017, August). Quary expansion using local and global document analysis. In *Acm sigir forum* (Vol. 51, No. 2, pp. 168–175). New York, NY, USA: ACM.

Yan, R., Song, Y., & Wu, H. (2016, July). Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 55–64). ACM.

Yanagawa, A., Chang, S. F., Kennedy, L., & Hsu, W. (2007). Columbia university's baseline detectors for 374 lscom semantic visual concepts. *Columbia University ADVENT technical report*, 222–2006.

Yao, Y., Rosasco, L., & Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, *26*(2), 289–315.

Yilmaz, E., & Aslam, J. A. (2006, November). Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM international conference on Information and knowledge management* (pp. 102–111). ACM.

Yilmaz, E., Kanoulas, E., & Aslam, J. A. (2008, July). A simple and efficient sampling method for estimating AP and NDCG. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 603–610). ACM.

Yu, C. T., & Salton, G. (1976). Precision weighting—an effective automatic indexing method. *Journal of the ACM (JACM)*, *23*(1), 76–88.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on systems, Man, and Cybernetics*, (1), 28–44.

Zeng, Q. T., Redd, D., Rindflesch, T., & Nebeker, J. (2012). Synonym, topic model and predicate-based query expansion for retrieving clinical documents. In *AMIA Annual Symposium Proceedings* (Vol. 2012, p. 1050). American Medical Informatics Association.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, *22*(2), 179–214.

Zhang, J., & Nguyen, T. N. (2005). A new term significance weighting approach. *Journal of Intelligent Information Systems*, *24*(1), 61–85.

Zhang, X. P., Zhou, X. Z., Huang, H. K., Feng, Q., Chen, S. B., & Liu, B. Y. (2011). Topic model for chinese medicine diagnosis and prescription regularities analysis: case on diabetes. *Chinese journal of integrative medicine*, *17*(4), 307–313.

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E. P., Yan, H., & Li, X. (2011). Comparing Twitter and traditional media using topic models. In *Advances in Information Retrieval* (pp. 338–349). Springer Berlin Heidelberg.

Zhou, Y., & Croft, W. B. (2005, October). Document quality models for web ad hoc retrieval. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (pp. 331–332). ACM.

Zhu, X., &Gauch, S. (2000, July). Incorporating quality metrics in centralized/distributed information retrieval on the World Wide Web. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 288–295). ACM.

Zuccon, G., Koopman, B., Bruza, P., & Azzopardi, L. (2015, December). Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian document computing symposium* (p. 12). ACM.

APPENDICES

Appendix A. Mean values of word features for 40 models with different numbers of topics (three groups – positive/negative/neutral)

	TP	WP	CTD	CTF	Norm_IDF	DF	TF	TP*WP
100 topics								
Positive	0.21	0.05	0.012676	244465.1	0.2205	93562.6	301125.1	0.0103
Negative	0.1938	0.0495	0.010787	214885.4	0.2476	73372.2	238057.3	0.0091
Neutral	0.1902	0.0852	0.019141	319171.4	0.1631	298501.3	1519051.1	0.0156
All	0.1948	0.0653	0.014752	265346.7	0.2062	175333.3	809944.9	0.0122
200 topics								
Positive	0.1804	0.0676	0.007188	178492.6	0.248	74857.8	222568.8	0.0117
Negative	0.166	0.0584	0.006285	160905.1	0.2793	57076.3	170398.8	0.0098
Neutral	0.1636	0.0991	0.012747	248549.8	0.191	268758.7	1354697.0	0.016
All	0.1675	0.0771	0.009161	200837.1	0.2367	149223.3	677698.6	0.0127
300 topics								
Positive	0.1712	0.0787	0.004927	140764.6	0.2773	63887.5	172812.3	0.0138
Negative	0.152	0.0628	0.004435	134517.3	0.3131	47284.3	119334.7	0.0093
Neutral	0.1547	0.1073	0.010088	215173.6	0.2308	235912.6	1177304.1	0.0162
All	0.1563	0.0841	0.006883	169316.5	0.2727	128998.1	571082.7	0.0129
400 topics								
Positive	0.162	0.0899	0.004342	128122.2	0.2784	68085.2	179477.9	0.0141
Negative	0.1484	0.072	0.003638	114893.2	0.3206	49803.1	122960.6	0.0105
Neutral	0.1493	0.103	0.008738	197057.9	0.2315	244582.9	1190385.4	0.0151
All	0.1508	0.0888	0.006067	154304.0	0.2739	141342.0	618302.2	0.0131
500 topics								
Positive	0.1582	0.0875	0.005704	192260.5	0.3063	62154.6	154434.2	0.0134
Negative	0.149	0.0685	0.004592	159290.8	0.3537	43253.6	100346.1	0.0099
Neutral	0.1469	0.114	0.008947	295608.1	0.2597	251448.6	1207865.0	0.0163
All	0.1495	0.091	0.006640	223106.5	0.3059	135912.9	586161.3	0.0132
600 topics								
Positive	0.1549	0.0971	0.002570	93740.6	0.3214	72546.4	139469.7	0.0146
Negative	0.1493	0.0725	0.002641	95040.5	0.3622	55902.6	103370.8	0.0105
Neutral	0.1458	0.1147	0.007291	169144.3	0.2711	247659.4	1139495.0	0.0161
All	0.1485	0.0947	0.004705	127898.8	0.3159	143728.0	570389.8	0.0136
700 topics								
Positive	0.1639	0.1049	0.002862	92452.0	0.3417	72030.4	137673.6	0.0164

Negative	0.1488	0.0761	0.002590	87478.9	0.3858	53162.8	97229.8	0.0107
Neutral	0.1504	0.1214	0.007152	165270.8	0.2846	260302.1	1264801.3	0.0171
All	0.1516	0.0999	0.004612	122003.2	0.3356	145886.5	610573.7	0.0143
800 topics								
Positive	0.1592	0.1103	0.002801	89485.6	0.3547	63165.6	130457.9	0.0175
Negative	0.1538	0.0722	0.002573	85759.9	0.3974	47315.7	92123.6	0.0103
Neutral	0.1512	0.1258	0.007281	165224.6	0.3044	253572.0	1236971.2	0.0177
All	0.1534	0.1005	0.004612	120156.0	0.3517	137474.0	585482.7	0.0145
900 topics								
Positive	0.1554	0.1106	0.002952	88249.1	0.3768	60616.8	125274.9	0.0157
Negative	0.1536	0.0811	0.002472	79470.2	0.4217	46677.3	89553.9	0.0116
Neutral	0.1528	0.1347	0.007368	160506.3	0.3041	278753.8	1330403.3	0.0189
All	0.1535	0.1079	0.004587	114590.4	0.3659	145593.1	612666.1	0.0153
1000 topics								
Positive	0.167	0.1172	0.001865	70016.1	0.3877	55868.2	113461.6	0.0181
Negative	0.1529	0.0816	0.001884	72110.5	0.434	41572.5	78475.4	0.0111
Neutral	0.1492	0.1358	0.002353	89781.5	0.3118	273116.7	1335654.8	0.019
	0.1534	0.1098	0.002081	79322.8	0.3754	142057.1	617951.6	0.0155
1100 topics								
Positive	0.1625	0.1315	0.003346	92861.5	0.385	60898.0	118656.6	0.0198
Negative	0.1578	0.085	0.002557	77949.8	0.434	45496.9	81958.3	0.0122
Neutral	0.1595	0.1401	0.006542	149916.8	0.3181	276910.3	1360200.1	0.0201
All	0.1592	0.1149	0.004351	110440.3	0.3781	145311.3	626370.1	0.0166
1200 topics								
Positive	0.1659	0.119	0.002411	75650.9	0.3881	59982.0	116449.8	0.0176
Negative	0.1553	0.087	0.002254	75742.9	0.4414	46836.9	84803.3	0.0121
Neutral	0.1592	0.1471	0.007342	159854.8	0.3055	306368.8	1485983.4	0.0209
All	0.1586	0.1171	0.004414	111042.4	0.3763	157788.3	677863.6	0.0166
1300 topics								
Positive	0.1662	0.1374	0.002457	75125.8	0.4017	58847.8	115134.2	0.0212
Negative	0.1588	0.0865	0.002502	80093.1	0.4592	44674.4	78011.9	0.0121
Neutral	0.159	0.1517	0.006152	145142.7	0.3228	292227.2	1451920.0	0.0211
All	0.16	0.1214	0.004026	106650.9	0.3935	150633.1	659936.7	0.0172
1400 topics								
Positive	0.166	0.1512	0.001681	59842.2	0.4201	57589.5	112311.6	0.0226

Negative	0.1632	0.0946	0.001921	66224.2	0.4621	44261.4	79967.9	0.013
Neutral	0.1566	0.1546	0.006980	158684.4	0.3272	290483.5	1442713.5	0.0215
	0.1607	0.1286	0.004073	105260.5	0.3978	152552.7	673394.2	0.0181
1500 topics								
Positive	0.1755	0.1376	0.002702	69110.6	0.4221	59992.6	110581.9	0.02
Negative	0.164	0.1003	0.002131	63458.6	0.4617	46879.9	81456.3	0.0139
Neutral	0.1585	0.1609	0.007141	153720.8	0.3255	309760.9	1520846.8	0.0227
All	0.1634	0.1315	0.004336	102502.0	0.3981	160090.1	694955.9	0.0185
1600 topics								
Positive	0.1641	0.1509	0.002128	64918.1	0.4305	54556.7	108923.9	0.0218
Negative	0.1616	0.0979	0.002062	66531.6	0.4695	44041.7	80164.7	0.0134
Neutral	0.1598	0.153	0.007289	150257.4	0.3225	305903.4	1519664.0	0.0214
All	0.1612	0.1292	0.004339	102634.1	0.4003	159125.9	708753.0	0.018
1700 topics								
Positive	0.1686	0.1545	0.002313	67701.0	0.4407	55625.5	110070.0	0.0235
Negative	0.1671	0.1026	0.002755	73803.7	0.4759	44192.5	80021.0	0.0138
Neutral	0.1622	0.1705	0.007539	161404.1	0.3259	330220.7	1656370.9	0.0236
All	0.1653	0.1386	0.004678	109302.1	0.4084	164761.7	739598.5	0.0193
1800 topics								
Positive	0.1646	0.1623	0.001525	54925.2	0.4413	57203.4	110040.1	0.0233
Negative	0.1676	0.1036	0.001887	59632.9	0.4834	45242.8	79741.9	0.0144
Neutral	0.1669	0.1672	0.005932	141969.3	0.3058	369231.0	1790670.3	0.0236
All	0.1668	0.1389	0.003529	93433.8	0.4028	182724.3	800932.2	0.0196
1900 topics								
Positive	0.1691	0.1597	0.002218	63789.5	0.4363	60253.6	116507.7	0.0239
Negative	0.1638	0.1049	0.003127	79307.4	0.4799	44326.7	78729.0	0.0144
Neutral	0.1642	0.1562	0.007135	155436.3	0.3229	324043.5	1612841.9	0.0215
All	0.1647	0.135	0.004768	110718.6	0.4048	169709.3	759617.1	0.0188
2000 topics								
Positive	0.1693	0.187	0.001420	34266.4	0.4468	59381.5	116120.3	0.0276
Negative	0.1675	0.1125	0.002458	50831.6	0.492	46665.6	81827.7	0.0157
Neutral	0.1692	0.1665	0.005349	94230.3	0.3099	375330.6	1874388.6	0.0231
All	0.1685	0.1462	0.003580	67542.0	0.4063	191978.6	869613.3	0.0206
2100 topics								

Positive	0.1677	0.1644	0.001264	48354.6	0.4487	56594.7	109522.6	0.0237
Negative	0.1647	0.116	0.001625	53542.6	0.4979	45898.5	80957.4	0.0156
Neutral	0.1667	0.1664	0.006839	150265.1	0.32	359832.8	1804436.1	0.0231
All	0.166	0.1453	0.003872	95433.1	0.4122	185904.0	845153.1	0.0201
2200 topics								
Positive	0.169	0.1839	0.002266	56650.7	0.4542	61162.0	115322.3	0.0267
Negative	0.1717	0.1166	0.001978	56299.4	0.5046	48235.8	83194.9	0.0159
Neutral	0.1698	0.162	0.005963	130673.1	0.3114	372779.3	1876470.5	0.0226
All	0.1704	0.1467	0.003830	90150.6	0.4097	197548.2	902718.8	0.0205
2300 topics								
Positive	0.1721	0.1863	0.001415	44016.5	0.4476	67367.4	128785.7	0.0264
Negative	0.1697	0.1142	0.001295	42313.3	0.4918	48329.9	86019.0	0.0153
Neutral	0.1696	0.1738	0.007688	157613.2	0.3058	385453.8	1860154.8	0.0245
All	0.17	0.1502	0.004084	92543.1	0.405	197170.9	861238.0	0.0209
2400 topics								
Positive	0.1703	0.187	0.002263	58842.9	0.446	69256.8	129096.6	0.0262
Negative	0.1705	0.1213	0.002103	57064.8	0.4967	50298.3	87318.0	0.0166
Neutral	0.1811	0.1604	0.004513	111992.9	0.2922	400323.1	1971873.6	0.0236
All	0.1754	0.1484	0.003233	82573.6	0.3956	213910.8	959840.7	0.0212
2500 topics								
Positive	0.1715	0.1879	0.001454	50603.7	0.4438	70245.9	136083.1	0.0281
Negative	0.1731	0.1166	0.001622	54387.5	0.4945	48412.8	84515.5	0.0158
Neutral	0.1802	0.1548	0.003545	109801.0	0.283	420170.2	2329574.4	0.0227
All	0.176	0.1428	0.002453	78462.2	0.3942	216054.0	1086486.9	0.0205
2600 topics								
Positive	0.1792	0.2101	0.001225	41826.8	0.4524	69184.3	133981.8	0.0307
Negative	0.1733	0.1317	0.001455	45199.1	0.5029	53146.3	92493.9	0.0176
Neutral	0.1768	0.1676	0.004877	121382.4	0.2997	403707.7	1975077.1	0.024
All	0.1757	0.1592	0.003023	80340.9	0.4011	219146.9	977875.5	0.0224
2700 topics								
Positive	0.1776	0.1983	0.001156	44553.7	0.4495	69189.5	135230.3	0.0298
Negative	0.1756	0.1209	0.001381	48801.0	0.4937	51085.7	90610.0	0.0175
Neutral	0.1809	0.1392	0.003512	104964.5	0.2727	430112.3	2337572.9	0.0211
All	0.1783	0.1398	0.002347	74488.9	0.3845	230699.4	1147016.8	0.0208

2800 topics								
Positive	0.1704	0.2081	0.001237	46282.5	0.4319	83449.3	160675.6	0.0305
Negative	0.1727	0.1281	0.001195	43698.8	0.4903	62892.4	111507.3	0.0181
Neutral	0.1808	0.142	0.003650	111526.1	0.2572	447964.0	2424301.6	0.0207
All	0.1763	0.1452	0.002387	76799.3	0.3701	251574.8	1235136.2	0.021
2900 topics								
Positive	0.1775	0.2055	0.001341	45903.0	0.4611	67135.8	128997.2	0.0303
Negative	0.1673	0.1302	0.001644	53550.7	0.4929	51708.6	93444.8	0.0177
Neutral	0.1826	0.1323	0.003867	117173.7	0.2573	449401.7	2338682.6	0.02
All	0.176	0.1407	0.002689	83594.2	0.3741	247467.7	1192140.9	0.0204
3000 topics								
Positive	0.1725	0.2133	0.000734	28080.8	0.4481	77380.8	146703.5	0.0314
Negative	0.1723	0.1336	0.000988	25517.7	0.4974	57423.4	98786.1	0.0187
Neutral	0.1843	0.1412	0.002827	61623.2	0.2636	450859.9	2116048.8	0.0215
All	0.1783	0.1476	0.001867	43747.2	0.3751	255045.8	1105077.4	0.0217
3100 topics								
Positive	0.1735	0.2261	0.001041	39415.3	0.4521	76703.1	151304.1	0.0341
Negative	0.1766	0.1335	0.001333	48727.1	0.5032	52313.5	95959.2	0.0187
Neutral	0.1882	0.1414	0.002628	86259.5	0.2609	449523.2	2413834.6	0.0211
All	0.1818	0.1488	0.001920	65633.1	0.3803	246453.1	1218071.5	0.0218
3200 topics								
Positive	0.1822	0.2096	0.001555	47934.9	0.4334	91727.5	181221.1	0.0323
Negative	0.1793	0.1351	0.002154	55087.3	0.4836	68249.7	123503.3	0.0191
Neutral	0.1904	0.1408	0.005309	119511.8	0.2453	464916.6	2317281.6	0.0216
All	0.1852	0.1475	0.003637	86040.4	0.3592	267554.8	1216446.6	0.022
3300 topics								
Positive	0.177	0.2104	0.001374	43885.1	0.442	84083.0	174751.1	0.0326
Negative	0.1753	0.1336	0.001913	54308.9	0.4962	61217.5	116293.0	0.0181
Neutral	0.188	0.1441	0.003864	95887.8	0.2481	453030.4	2105413.3	0.0223
All	0.1817	0.1488	0.002798	73309.7	0.3676	256112.0	1098176.7	0.022
3400 topics								
Positive	0.1835	0.2265	0.001375	46399.7	0.4589	73151.5	150347.1	0.0349
Negative	0.1855	0.1394	0.001460	48996.0	0.5073	58734.8	104120.5	0.0194

Neutral	0.1958	0.1337	0.002623	83300.4	0.2589	459395.0	2427505.7	0.0218
All	0.1904	0.1475	0.002032	65862.6	0.3767	261359.8	1274416.8	0.0226
3500 topics								
Positive	0.1794	0.203	0.001216	43347.2	0.4389	93825.4	186319.4	0.0306
Negative	0.1761	0.1401	0.001400	46254.9	0.4933	70160.1	127159.6	0.0195
Neutral	0.1925	0.1248	0.003098	96981.9	0.2409	465627.3	2457650.1	0.0201
All	0.1849	0.1402	0.002249	71955.3	0.3567	276393.8	1332316.6	0.0212
3600 topics								
Positive	0.1803	0.2126	0.002866	68022.4	0.4355	87931.5	176934.3	0.0328
Negative	0.1775	0.1385	0.002214	56084.4	0.4953	64172.1	118148.8	0.02
Neutral	0.1887	0.1351	0.003937	103990.8	0.2298	481187.5	2389731.2	0.0207
All	0.1835	0.1461	0.003170	81887.6	0.3531	278733.2	1278057.6	0.0219
3700 topics								
Positive	0.1853	0.228	0.001134	40731.4	0.4386	90423.0	177652.2	0.0351
Negative	0.1875	0.1472	0.001066	38438.0	0.4855	72158.3	134451.6	0.0223
Neutral	0.1987	0.1214	0.002583	86028.8	0.2127	500347.9	2641402.2	0.02
All	0.1932	0.1432	0.001882	64040.4	0.3346	302224.9	1473742.8	0.0226
3800 topics								
Positive	0.1948	0.2029	0.001104	37343.9	0.4221	99510.6	211988.3	0.0326
Negative	0.1827	0.148	0.001077	36406.1	0.4853	76599.8	150403.8	0.0212
Neutral	0.1983	0.1216	0.003075	89371.0	0.2205	493699.3	2624308.2	0.0202
All	0.1923	0.1419	0.002104	63666.2	0.3411	293380.6	1426091.5	0.0222
3900 topics								
Positive	0.1786	0.2167	0.001389	42186.4	0.4192	113113.4	229986.4	0.0332
Negative	0.1875	0.1375	0.001761	48583.5	0.4883	79132.2	147450.5	0.02
Neutral	0.1932	0.129	0.003721	89300.7	0.2277	483521.8	2585532.1	0.0206
All	0.1893	0.1434	0.002705	68369.4	0.3475	288151.2	1391937.4	0.022
4000 topics								
Positive	0.1882	0.2141	0.001272	44660.1	0.4295	103537.4	224528.0	0.0332
Negative	0.1762	0.1448	0.001226	43910.4	0.4929	68199.9	141934.2	0.0213
Neutral	0.2053	0.1257	0.002340	77944.3	0.2364	471007.3	2501796.0	0.0209
All	0.1928	0.1438	0.001807	61585.2	0.3523	280791.8	1371412.0	0.0226
All topics								
Positive	0.1713	0.1496	0.002820	81768.4	0.3905	70097.8	149511.7	0.0227

Negative	0.1653	0.1012	0.002599	77393.9	0.4398	52445.1	103399.2	0.0143
Neutral	0.1702	0.1371	0.006212	145368.5	0.2738	355446.2	1791520.5	0.0202
All	0.1683	0.1242	0.004263	108726.4	0.3578	191851.0	872705.5	0.0181

Appendix B. The mean of standardized word feature values for 40 models with different umbers of topics

(three groups – positive/negative/neutral)

	TP	WP	CTD	CTF	Norm_IDF	DF	TF	TP*WP
100 topics								
Positive	0.1016	-0.1479	-0.153	-0.1268	0.1341	-0.3221	-0.3	-0.0763
Negative	-0.0066	-0.1526	-0.2922	-0.3064	0.3871	-0.4016	-0.3372	-0.128
Neutral	-0.0309	0.1936	0.3233	0.3268	-0.4034	0.4851	0.4181	0.145
200 topics								
Positive	0.0967	-0.075	-0.1738	-0.1539	0.0939	-0.3032	-0.2798	-0.0352
Negative	-0.0113	-0.1485	-0.2534	-0.2751	0.3544	-0.3757	-0.3119	-0.1062
Neutral	-0.0292	0.1741	0.316	0.3287	-0.3802	0.4874	0.4163	0.1169
300 topics								
Positive	0.1168	-0.0344	-0.192	-0.1941	0.0341	-0.2584	-0.2541	0.0283
Negative	-0.034	-0.1352	-0.2403	-0.2366	0.3008	-0.3243	-0.2883	-0.116
Neutral	-0.0124	0.1478	0.3146	0.3118	-0.3122	0.4243	0.3868	0.104
400 topics								
Positive	0.0911	0.0069	-0.175	-0.1837	0.0311	-0.286	-0.2763	0.0297
Negative	-0.0193	-0.0994	-0.2464	-0.2766	0.318	-0.3574	-0.3119	-0.0789
Neutral	-0.012	0.0848	0.2709	0.3	-0.288	0.403	0.3603	0.0596
500 topics								
Positive	0.0706	-0.0187	-0.0398	-0.0385	0.0022	-0.2722	-0.2694	0.0052
Negative	-0.0043	-0.1202	-0.0871	-0.0796	0.2823	-0.3419	-0.3031	-0.0934
Neutral	-0.0209	0.1226	0.0981	0.0904	-0.2731	0.4263	0.3879	0.0883
600 topics								
Positive	0.0523	0.0117	-0.2418	-0.242	0.0313	-0.2478	-0.2719	0.026
Negative	0.0063	-0.1099	-0.2337	-0.2328	0.2657	-0.3058	-0.2947	-0.0798
Neutral	-0.0223	0.0984	0.293	0.2922	-0.2567	0.3619	0.3591	0.066
700 topics								
Positive	0.1002	0.0232	-0.19	-0.2077	0.0313	-0.2499	-0.2839	0.0541
Negative	-0.0232	-0.1098	-0.2195	-0.2426	0.2588	-0.3137	-0.3081	-0.091
Neutral	-0.01	0.0996	0.2758	0.3041	-0.2626	0.3871	0.3927	0.0711
800 topics								
Positive	0.047	0.0446	-0.1933	-0.2126	0.0153	-0.2533	-0.2759	0.0738
Negative	0.0029	-0.1282	-0.2177	-0.2384	0.2314	-0.3074	-0.2992	-0.1027
Neutral	-0.0187	0.1148	0.2849	0.3124	-0.2393	0.3958	0.3951	0.0793
900 topics								

Positive	0.0149	0.0119	-0.1677	-0.1761	0.0513	-0.2791	-0.2889	0.0102
Negative	0.0006	-0.1152	-0.2169	-0.2348	0.2626	-0.3249	-0.3101	-0.0853
Neutral	-0.006	0.1156	0.2853	0.3069	-0.2915	0.4373	0.4254	0.0852
1000 topics								
Positive	0.1111	0.031	-0.0218	-0.0233	0.0584	-0.2815	-0.2948	0.06
Negative	-0.0036	-0.1165	-0.0199	-0.018	0.2792	-0.3282	-0.3152	-0.0999
Neutral	-0.0338	0.1078	0.0275	0.0261	-0.3031	0.428	0.4194	0.0811
1100 topics								
Positive	0.026	0.0688	-0.1075	-0.1157	0.0322	-0.2696	-0.2907	0.0693
Negative	-0.0107	-0.1238	-0.1918	-0.2139	0.2599	-0.3188	-0.3117	-0.0975
Neutral	0.0022	0.1043	0.2343	0.2599	-0.279	0.4203	0.4201	0.077
1200 topics								
Positive	0.0574	0.0077	-0.2099	-0.2346	0.0527	-0.3057	-0.3162	0.0217
Negative	-0.0253	-0.1218	-0.2263	-0.234	0.2897	-0.3468	-0.334	-0.0994
Neutral	0.0052	0.1216	0.3068	0.3235	-0.3149	0.4644	0.4551	0.0937
1300 topics								
Positive	0.049	0.0615	-0.1808	-0.2161	0.0362	-0.2855	-0.3023	0.0838
Negative	-0.009	-0.1339	-0.1756	-0.1821	0.2904	-0.3296	-0.3229	-0.1088
Neutral	-0.008	0.1163	0.2448	0.2639	-0.3122	0.4405	0.4394	0.0826
1400 topics								
Positive	0.0416	0.0835	-0.2674	-0.3059	0.0975	-0.2917	-0.3077	0.0926
Negative	0.0191	-0.1257	-0.2406	-0.263	0.2806	-0.3326	-0.3255	-0.1035
Neutral	-0.0326	0.0959	0.3251	0.3599	-0.3082	0.4237	0.4219	0.0711
1500 topics								
Positive	0.0932	0.0222	-0.1647	-0.2144	0.102	-0.3033	-0.3166	0.0301
Negative	0.0047	-0.114	-0.2223	-0.2507	0.2713	-0.343	-0.3324	-0.0934
Neutral	-0.0376	0.1073	0.2826	0.3289	-0.31	0.4535	0.4475	0.0837
1600 topics								
Positive	0.0231	0.0788	-0.223	-0.248	0.1292	-0.3191	-0.3261	0.0762
Negative	0.0034	-0.1138	-0.2297	-0.2374	0.2959	-0.3512	-0.3417	-0.0941
Neutral	-0.0108	0.0866	0.2975	0.3132	-0.3325	0.448	0.4408	0.068
1700 topics								
Positive	0.0258	0.0554	-0.232	-0.2549	0.1333	-0.3237	-0.3285	0.0795
Negative	0.0141	-0.1249	-0.1886	-0.2175	0.2785	-0.3577	-0.3441	-0.1049
Neutral	-0.024	0.1111	0.2807	0.3192	-0.3397	0.4908	0.4783	0.0815
1800 topics								
Positive	-0.0173	0.0815	-0.2478	-0.2806	0.1545	-0.3582	-0.356	0.072
Negative	0.0058	-0.1225	-0.203	-0.2463	0.3237	-0.3924	-0.3716	-0.0988

Neutral	0.0001	0.0984	0.2973	0.3537	-0.3896	0.5323	0.5099	0.0772
1900 topics								
Positive	0.0343	0.0874	-0.244	-0.2837	0.1303	-0.3213	-0.3351	0.0985
Negative	-0.0069	-0.1063	-0.1571	-0.1899	0.3102	-0.368	-0.3548	-0.0861
Neutral	-0.004	0.0749	0.2265	0.2704	-0.3382	0.453	0.4446	0.0521
2000 topics								
Positive	0.0063	0.1378	-0.1851	-0.1415	0.1565	-0.3719	-0.3735	0.1305
Negative	-0.0077	-0.1139	-0.0961	-0.0711	0.3313	-0.4075	-0.3905	-0.0892
Neutral	0.0055	0.0686	0.1516	0.1135	-0.3728	0.5142	0.4981	0.0467
2100 topics								
Positive	0.0131	0.0641	-0.2848	-0.3192	0.1417	-0.3691	-0.3613	0.0664
Negative	-0.0102	-0.0987	-0.2454	-0.284	0.3326	-0.3996	-0.3753	-0.0826
Neutral	0.0051	0.0709	0.3242	0.3718	-0.3581	0.4964	0.4711	0.0551
2200 topics								
Positive	-0.0112	0.1236	-0.1694	-0.2241	0.1713	-0.3789	-0.3691	0.1129
Negative	0.0096	-0.0997	-0.2006	-0.2265	0.3657	-0.4149	-0.3841	-0.0829
Neutral	-0.0051	0.0508	0.2311	0.2711	-0.379	0.4869	0.4564	0.0392
2300 topics								
Positive	0.0158	0.1206	-0.2691	-0.3119	0.1646	-0.3597	-0.3667	0.101
Negative	-0.0023	-0.1199	-0.2812	-0.3228	0.3351	-0.4124	-0.3881	-0.1018
Neutral	-0.0029	0.0787	0.3635	0.4182	-0.3826	0.5217	0.5001	0.0673
2400 topics								
Positive	-0.0366	0.1289	-0.1213	-0.1831	0.1902	-0.391	-0.3979	0.0903
Negative	-0.0356	-0.0906	-0.1414	-0.1968	0.3819	-0.4422	-0.4179	-0.0808
Neutral	0.0421	0.0401	0.16	0.2269	-0.3905	0.5038	0.4847	0.0432
2500 topics								
Positive	-0.0323	0.1555	-0.2275	-0.2484	0.1876	-0.3931	-0.377	0.1398
Negative	-0.0206	-0.0904	-0.1893	-0.2146	0.3791	-0.4519	-0.3974	-0.085
Neutral	0.0293	0.0415	0.2489	0.2794	-0.42	0.5503	0.493	0.0409
2600 topics								
Positive	0.0252	0.1638	-0.2431	-0.3026	0.1901	-0.3995	-0.4009	0.1439
Negative	-0.018	-0.0883	-0.212	-0.2761	0.3771	-0.4422	-0.4206	-0.0827
Neutral	0.0079	0.0272	0.2508	0.3225	-0.3754	0.4916	0.4738	0.0283
2700 topics								
Positive	-0.0054	0.204	-0.2805	-0.2862	0.2415	-0.4309	-0.4074	0.1612
Negative	-0.0197	-0.0661	-0.2276	-0.2456	0.4056	-0.4792	-0.4253	-0.06
Neutral	0.0183	-0.002	0.2744	0.2913	-0.415	0.532	0.4793	0.0051
2800 topics								

Positive	-0.0421	0.211	-0.2653	-0.2733	0.225	-0.445	-0.4181	0.1694
Negative	-0.0259	-0.0573	-0.2752	-0.2964	0.4377	-0.4994	-0.4373	-0.0518
Neutral	0.032	-0.0106	0.2918	0.311	-0.4113	0.5198	0.4628	-0.0039
2900 topics								
Positive	0.0105	0.2241	-0.2928	-0.3227	0.3187	-0.474	-0.4288	0.1822
Negative	-0.0628	-0.0364	-0.2269	-0.2572	0.4355	-0.5145	-0.4431	-0.0495
Neutral	0.0471	-0.0291	0.2557	0.2875	-0.4279	0.5308	0.4624	-0.0078
3000 topics								
Positive	-0.0399	0.2189	-0.0685	-0.0598	0.2637	-0.4607	-0.4526	0.1612
Negative	-0.0419	-0.0465	-0.0531	-0.0696	0.4415	-0.5125	-0.4753	-0.051
Neutral	0.0421	-0.0213	0.058	0.0683	-0.403	0.5078	0.4775	-0.003
3100 topics								
Positive	-0.0583	0.2583	-0.2781	-0.2805	0.2573	-0.4487	-0.4118	0.2135
Negative	-0.0368	-0.0509	-0.1857	-0.1809	0.4408	-0.5132	-0.4332	-0.0533
Neutral	0.0452	-0.0247	0.224	0.2207	-0.4281	0.5368	0.4616	-0.0111
3200 topics								
Positive	-0.0209	0.2076	-0.2349	-0.2706	0.2667	-0.4601	-0.4302	0.1764
Negative	-0.0421	-0.0415	-0.1673	-0.2198	0.447	-0.5216	-0.4542	-0.0507
Neutral	0.0374	-0.0227	0.1885	0.2377	-0.4093	0.5165	0.4575	-0.0076
3300 topics								
Positive	-0.0337	0.2024	-0.206	-0.2491	0.2676	-0.4544	-0.4391	0.1783
Negative	-0.0458	-0.05	-0.1281	-0.1609	0.4624	-0.5148	-0.4669	-0.0668
Neutral	0.0445	-0.0154	0.1542	0.1912	-0.4297	0.5201	0.4789	0.0041
3400 topics								
Positive	-0.0481	0.2628	-0.1855	-0.2163	0.289	-0.4898	-0.4299	0.2064
Negative	-0.0342	-0.0269	-0.1615	-0.1875	0.4593	-0.5274	-0.4475	-0.0523
Neutral	0.0376	-0.0459	0.1668	0.1938	-0.4144	0.5154	0.441	-0.0129
3500 topics								
Positive	-0.038	0.2158	-0.2682	-0.2936	0.2901	-0.4777	-0.442	0.1593
Negative	-0.0606	-0.0006	-0.2206	-0.2637	0.4822	-0.5396	-0.4648	-0.0291
Neutral	0.0517	-0.0528	0.2204	0.2568	-0.4087	0.4951	0.434	-0.0189
3600 topics								
Positive	-0.0224	0.2216	-0.0395	-0.1067	0.2926	-0.4996	-0.4341	0.1831
Negative	-0.0424	-0.0252	-0.1243	-0.1985	0.505	-0.5619	-0.4573	-0.0329
Neutral	0.0362	-0.0365	0.0997	0.1701	-0.4378	0.5302	0.4383	-0.0215
3700 topics								
Positive	-0.0544	0.2904	-0.2512	-0.2832	0.3652	-0.5467	-0.484	0.2119
Negative	-0.0392	0.0136	-0.2739	-0.3111	0.5301	-0.5938	-0.5002	-0.0055

Neutral	0.0379	-0.0748	0.2357	0.2672	-0.4286	0.5114	0.4361	-0.0445
3800 topics								
Positive	0.0165	0.2088	-0.2401	-0.2957	0.2836	-0.5058	-0.455	0.1754
Negative	-0.066	0.0207	-0.2468	-0.3062	0.505	-0.5656	-0.4781	-0.0179
Neutral	0.041	-0.0696	0.2332	0.2887	-0.4222	0.5226	0.4491	-0.0341
3900 topics								
Positive	-0.0758	0.2466	-0.2025	-0.2558	0.2492	-0.4588	-0.4525	0.1916
Negative	-0.0124	-0.0197	-0.1453	-0.1933	0.4898	-0.5479	-0.4847	-0.034
Neutral	0.0282	-0.0483	0.1563	0.2045	-0.417	0.5121	0.4648	-0.024
4000 topics								
Positive	-0.0309	0.2386	-0.1936	-0.2094	0.2711	-0.4644	-0.4335	0.1769
Negative	-0.1121	0.0033	-0.21	-0.2187	0.4937	-0.557	-0.4647	-0.0218
Neutral	0.0847	-0.0616	0.1925	0.2024	-0.4067	0.4984	0.4273	-0.0291

Appendix C. Mean infAP & infNDCG scores of 40 LDA models with different numbers of topics for the top3 retrieved documents with score weighting of the rank number to the power of 2

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0161	0.0195	**0.0233**	0.0203	0.0153	0.0186	0.0186	0.0188	**0.0236**	0.02
infNDCG	0.1515	0.1629	0.196	0.1738	0.1537	0.1766	0.1671	0.1801	**0.1827**	0.1771

no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0197	0.0189	0.0196	0.0201	0.0195	**0.0233**	**0.0243**	**0.0215**	0.0209	0.02
infNDCG	0.1698	0.1716	0.1701	0.1772	0.176	**0.1936**	**0.1876**	**0.1943**	0.1696	0.1725

no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0203	0.0199	0.0201	**0.021**	0.0172	**0.0214**	0.0194	0.0201	0.0188	**0.0228**
infNDCG	**0.1898**	0.1534	0.1745	**0.1809**	0.171	0.1728	0.1675	0.1691	0.1725	**0.1821**

no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0168	0.0188	0.0192	**0.0224**	0.0189	0.0178	**0.0232**	0.0193	**0.0229**	0.0194
infNDCG	0.1744	0.1778	0.1674	0.1746	0.1761	0.1707	0.1689	0.1733	0.1712	0.1709

* baseline run - infAP: 0.0209 & infNDCG: 0.1808

Appendix D. Mean infAP & infNDCG scores of 40 LDA models with different numbers of topics for the top4 retrieved documents with score weighting of the rank number to the power of 2

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0161	0.0195	**0.0233**	0.0203	0.0155	0.0188	0.0179	0.0182	**0.0236**	0.02
infNDCG	0.1509	0.1634	**0.1952**	0.1739	0.1545	0.1798	0.1654	0.1783	**0.1822**	0.1762

no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0198	0.0192	0.0195	0.0203	0.019	**0.023**	**0.0251**	**0.0213**	**0.0212**	**0.021**
infNDCG	0.1681	0.1735	0.1687	0.1802	0.1731	**0.1924**	**0.1925**	**0.1942**	0.1737	0.1745

no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0206	**0.0219**	0.0199	0.0208	0.0173	**0.0222**	0.0195	0.0207	0.0199	**0.0234**
infNDCG	**0.1902**	0.169	0.1772	0.1771	0.173	**0.1827**	0.1714	0.1712	0.1749	**0.1869**

no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0184	0.0197	0.0206	**0.0229**	0.0189	0.0183	**0.0235**	0.0195	**0.023**	0.0205
infNDCG	0.1695	**0.1882**	0.1691	**0.1859**	**0.1842**	0.1717	**0.1919**	0.1784	**0.1899**	**0.1845**

* baseline run - infAP: 0.0209 & infNDCG: 0.1808

Appendix E. Mean infAP & infNDCG scores of 40 LDA models with different numbers of topics for the top5 retrieved documents with score weighting of the rank number to the power of 2

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0161	**0.0195**	**0.0233**	0.0203	0.0155	0.0188	0.0179	0.0183	**0.0236**	0.0202
infNDCG	0.1509	0.1634	**0.1952**	0.1739	0.1545	0.1798	0.1654	0.179	**0.1822**	0.1774
no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0196	0.0196	0.0195	0.0203	0.0192	**0.0222**	**0.0244**	**0.0215**	**0.0212**	0.0208
infNDCG	0.1673	0.1752	0.1687	0.1802	0.1742	**0.1878**	**0.19**	**0.1951**	0.1735	0.1727
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0203	**0.0221**	0.0189	**0.021**	0.0175	**0.0221**	0.019	0.0205	0.0197	**0.0231**
infNDCG	**0.1857**	0.1697	0.1738	0.1789	0.1741	0.1802	0.1687	0.1673	0.1729	**0.183**
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0176	0.0199	0.0205	**0.0225**	0.0184	0.0187	**0.0233**	0.0197	**0.0225**	0.0199
infNDCG	0.1645	**0.1881**	0.1677	**0.1831**	**0.1809**	0.1727	**0.1901**	0.1778	**0.1888**	0.1795

* baseline run - infAP: 0.0209 & infNDCG: 0.1808

Appendix F. IR performance comparison of two binary ANN classifiers for the WSW model: 3 layers * 500 nodes vs. 2 layers * 700 nodes

A classifier can be evaluated by three metrics (accuracy, F1, or AUC), in detail, for positive/negative/neutral words. In terms of F1 and AUC, the binary classifier with 3 layers including 500 nodes per layer looks best on the training set, while the classifiers with one layer showed better performance in terms of accuracy on the validation set. Considering overall performance for three metrics on the training set, the classifier with 3 layers (500 nodes per layer) looks fine. However, the best performance over 40 LDA models was shown in the classifier with 2 layers including 700 nodes per layer, where overall scores for three metrics were good. The average mean infAP and infNDCG scores for the classifier with 3 layers and 700 nodes per layer were 0.0203 and 0.1754, respectively, which are lower than the scores of the baseline run as well as the QE model using an LDA model and the classifier with 2 layers and 700 nodes per layer. There were statistically significant differences in the average mean scores (paired t-test, alpha= 0.01, p-value = 6.45E-09 for infAP and 6.52E-09 for infNDCG).

For instance, the mean infAP and infNDCG scores for the QE model using the classifier (3 layers * 500 nodes) based on the Word Score Weighting model were listed in Table 36 and compared with the scores for the classifier (2 layers * 700 nodes, Table 20) in Figure 34 and 35.

Table 36. Mean infAP and infNDCG scores of the WSW model using an LDA model and a binary ANN (3 layers * 500 nodes) classifier for the top 2 retrieved documents (a maximum of the top 10 words for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0179	0.0224	**0.0259**	0.0194	0.0166	0.0152	0.0187	0.0171	0.0235	0.0191
infNDCG	0.1707	0.1722	**0.1993**	0.1732	0.1628	0.1602	0.1726	0.1696	0.1812	0.1723

no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0178	0.0219	0.019	0.0161	0.0214	0.0206	0.0234	0.0163	0.0192	0.0184
infNDCG	0.1603	0.1861	0.1737	0.1518	0.1796	0.1763	0.1872	0.1718	0.1675	0.1746

no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0187	0.0195	0.0211	0.0191	0.0216	0.0229	0.0213	0.0211	0.0197	0.0205
infNDCG	0.1704	0.1718	0.1719	0.1724	0.1786	0.1755	0.1807	0.1826	0.1668	0.1755

no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0214	0.0192	0.0214	0.0235	0.0216	0.0212	0.0252	0.0213	0.0221	0.0204
infNDCG	0.1671	0.1763	0.1736	0.1863	0.1902	0.1909	0.1863	0.1821	0.1808	0.1725

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

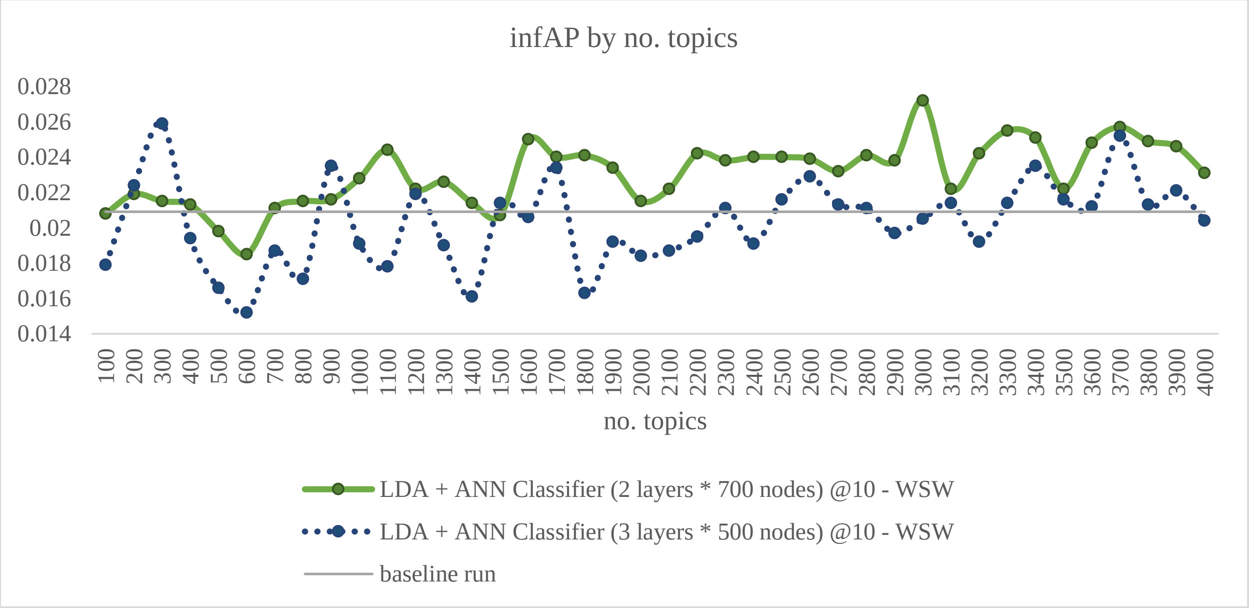

Figure 34. Mean infAP scores of 40 WSW (an LDA model + a binary classifier) models for the top 2 retrieved documents: 3 layers * 500 nodes vs. 2 layers * 700 nodes

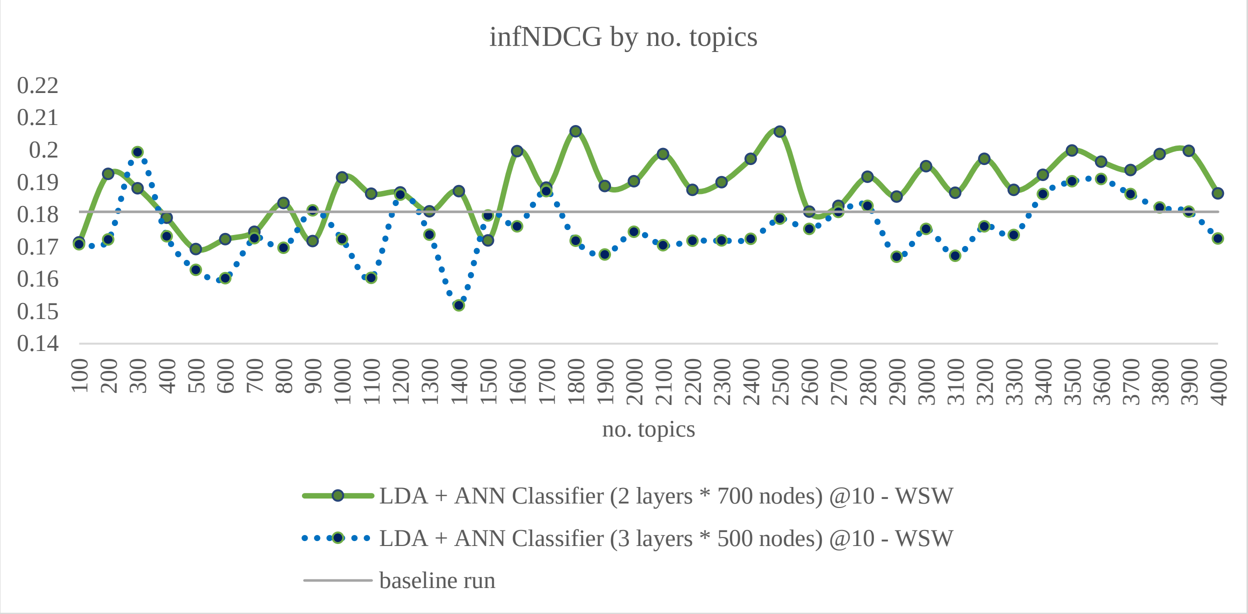

Figure 35. Mean infNDCG scores of 40 WSW (an LDA model + a binary classifier) models for the top 2 retrieved documents (3 layers * 500 nodes vs. 2 layers * 700 nodes)

Similarly, the mean infAP and infNDDCG scores for the QE model based on Positive Word Selection using the top 7 words were compared (Table 37, Figure 36 & 37). The average mean scores of the QE model based on the classifier with 3 layers and 500 nodes per layer, were 0.0208 for infAP and 0.1807 for infNDCG, which are lower than the scores of the baseline run as well as the QE model using an LDA model and the classifier with 2 layers and 700 nodes per layer. There were statistically significant differences in the average mean scores (paired t-test, alpha= 0.01, p-value = 1.54E-10 for infAP and 0.008 for infNDCG).

Table 37. Mean infAP and infNDCG scores of 40 PWS (an LDA model + a binary classifier) models for the top 2 retrieved documents (3 layers * 500 nodes, a maximum of the top 7 words for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0212	0.0211	0.0216	0.0215	0.0196	0.0202	0.0205	0.0202	0.0192	0.0216
infNDCG	0.1824	0.186	0.18	0.1852	0.1785	0.1829	0.1807	0.1748	0.1747	0.1822

no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0196	0.0217	0.021	0.0202	0.022	**0.0226**	0.02	0.0196	0.0197	0.0195

infNDCG	0.1739	0.1887	0.1812	0.1718	0.1872	**0.1892**	0.1711	0.1774	0.1738	0.1709
no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0203	0.0212	0.0203	0.0213	0.0194	0.0222	0.0199	0.0212	0.0213	0.0222
infNDCG	0.1797	0.1812	0.1723	0.1827	0.1735	0.1881	0.1821	0.1804	0.1821	0.1851
no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0213	0.0211	0.0223	0.0212	0.0212	0.0217	0.0203	0.0222	0.0199	0.0203
infNDCG	0.184	0.1841	0.189	0.1803	0.1823	0.1849	0.179	0.1855	0.1788	0.1793

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

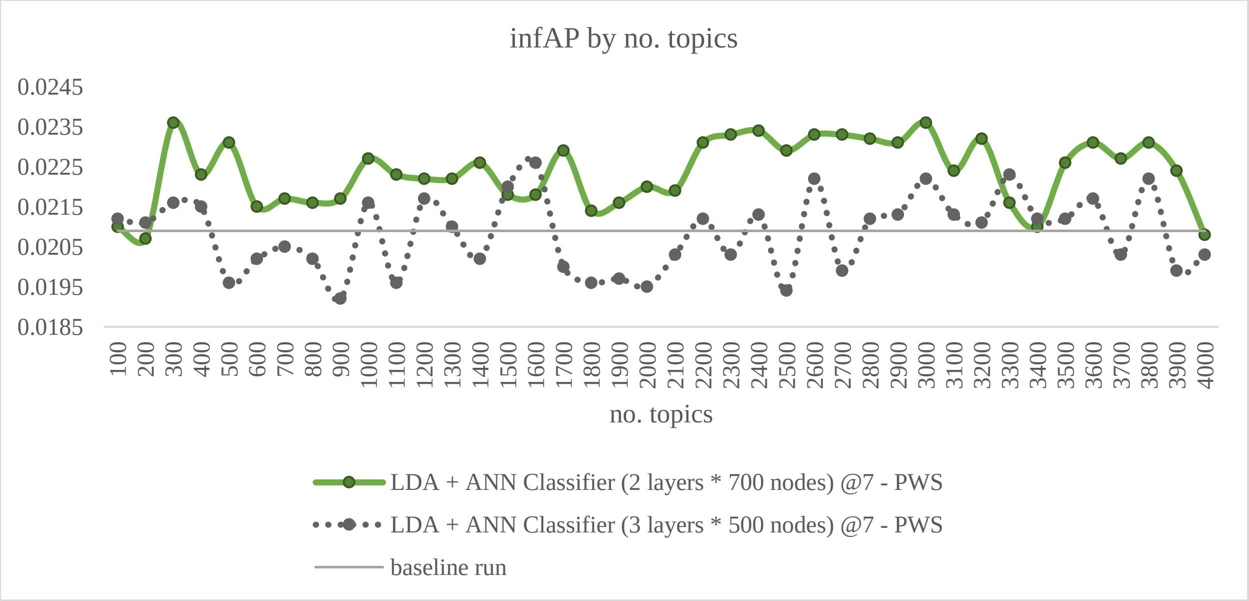

Figure 36. Mean infAP scores of 40 PWS (an LDA model + a binary classifier) models for the top 2 retrieved documents – 3 layers * 500 nodes vs. 2 layers * 700 nodes

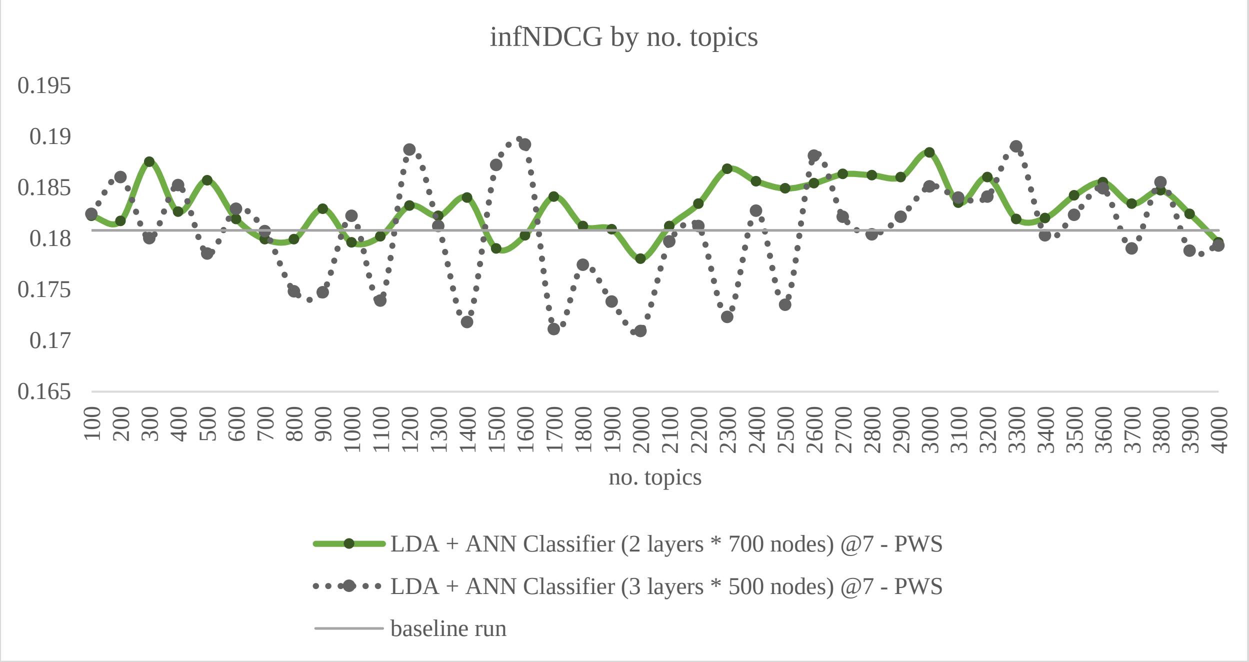

Figure 37. Mean infNDCG scores of 40 PWS (LDA + a binary classifier) models for the top 2 retrieved documents – 3 layers * 500 nodes vs. 2 layers * 700 nodes

In application, it would not be appropriate to refer to only one metric (e.g. accuracy, F1, or AUC) in selecting a classifier. Basically, developing more effective word features and collecting more training /validation data would be more effective to increase the performance of ANN classifiers rather than focusing on parameter settings.

Other factors, such as cut-off points (e.g. the number of words to be selected for QE or thresholds for TP/WP/TP*WP), the number of topics, the number of iterations for LDA model training, etc.) might affect infAP and infNDCG more critically. Anyway, the relationship between three metrics (accuracy, F1, and AUC) and (infAP & infNDCG) should be more studied in future research.

Appendix G. 30 queries followed by top 3 terms generated by the ensemble model based on Word Score Weighting for QE (mean scores - infAP: 0.0271 & infNDCG: 0.2055).

Topic No.	Query followed by three expanded terms (by the order of word score)
1	A 78 year old male presents with frequent stools and melena "rupture" "hemorrhage" "child"
2	An elderly female with past medical history of right hip arthroplasty presents after feeling a snap of her right leg and falling to the ground "osteoarthritis" "titanium" "smoke"
3	A 75F found to be hypoglycemic with hypotension and bradycardia She had UA positive for klebsiella She had a leukocytosis to 18 and a creatinine of 6 Pt has blood cultures positive for group A streptococcus On the day of transfer her blood pressure dropped to the 60s She was anuric throughout the day, awake but drowsy This morning she had temp 963, respiratory rate 22, BP 102 26 "smoke" "burns" "regression analysis"
4	An 87 yo woman with h o osteoporosis, DM2, dementia, depression, and anxiety presents s p fall with evidence of C2 fracture, chest pain, tachycardia, tachypnea, and low blood pressure "calibration" "dialysis" "x-rays"
5	An 82 man with multiple chronic conditions and previous surgeries presents with 9 day history of productive cough, fever and dyspnea "kidney" "publications" "heart"
6	A 94 year old female with hx recent PE DVT, atrial fibrillation, CAD presents with fever and abdominal pain An abdominal CT demonstrates a distended gallbladder with gallstones and biliary obstruction with several CBD stones "tomography" "research" "role"
7	A 41 year old male patient with medical history of alcohol abuse, cholelithiasis, hypertension, obesity who presented to his local hospital with hematemasis, abdominal pain radiating to the back and elevated lipase Signs of ascites, pancytopenia and coagulopathy "membrane proteins" "pancreatitis" "time"
8	A 26 year old diabetic woman, estimated to 10 weeks pregnant, presents with hyperemesis Her labwork demonstrates a blood glucose of 160, bicarbonate of 11, beta hCG of 3373 and ketones in her urine "pregnancy" "immunoassay" "chemistry"
9	Infant with respiratory distress syndrome and extreme prematurity Chest x ray shows diffuse bilateral opacities within the lungs, with increased lung volumes "infant" "paper" "work"
10	A 55 year old woman with sarcoidosis, presenting today with confusion and worsening asterixis In the waiting room, the pt became more combative and then unresponsive Ammonia level 280 on admission "prognosis" "france" "urea"
11	80 yo male with demantia and past medical history of CABG with repeated episodes of chest pain Admitted for severe chest pain episode "men" "research" "work"
12	66 yo female pedestrian struck by auto Unconscious and unresponsive at scene Multiple fractures and head CT showing extensive interparenchymal hemorrhages "mortality" "morbidity" "paper"
13	A 43 year old woman with history of transverse myelitis leading to paraplegia, depression, frequent pressure ulcers, presenting with chills, agitation, rigors, and back pain Patient has

stage IV decubitus ulcers on coccyx and buttocks, heels Admission labs significant for thrombocytosis, elevated lactate, and prolonged PT "leg" "blood" "research"

14 A 52 year old woman with history of COPD and breast cancer who presents with SOB, hypoxia, cough, fevers and sore throat for several weeks "adult" "research" "disease"

15 67 yo male smoker with end stage COPD on home oxygen, tracheobronchomalacia, s p RUL resection for squamous cell carcinoma Y stent placement was complicated by cough and copious secretions requiring multiple therapeutic aspirations Patient reports decreased appetite, 50 lb wt loss in 6 months Decreased activity tolerance PET scan revealed some FDG avid nodes concerning for recurrence Pt presents with worsening SOB with R shoulder pain and weakness "tomography" "lymph" "autopsy"

16 A 90 year old woman who was recently hospitalized for legionella PNA, with confusion and dysarthria the last few days Found down in the bathroom this morning, making non verbal utterances and with minimal movement of the right side "stroke" "ganglia" "infarction"

17 76 year old female with personal history of diastolic congestive heart failure, atrial fibrillation on Coumadin, presenting with low hematocrit and dyspnea "mortality" "exercise" "morbidity"

18 A 40 year old woman with a history of alcoholism complicated by Delirium Tremens and seizures 2 years ago, polysubstance abuse, hep C, presents with abdominal pain in lower quadrants, radiating to the back, nausea, vomitting and diarrhea Labs are significant for elevated lipase "molecular biology" "counseling" "surgeons"

19 78 year old female with PMHx HTN, dCHF, Diabetes, CKD, Atrial fibrillation on coumadin, ischemic stroke, admitted after presenting with confusion and somnolence She was recently discharged after presyncope falls Patient has had confusion at home for 3 weeks The patient denies headache, blurry vision, numbness, tingling or weakness, nausea or vomiting "morbidity" "awareness" "body weight"

20 A 87 yo female reports several days abdominal pain, worse yesterday, severe and more localized to the right, accompanied by nausea and vomitting Labs show elevated bilirubin, transaminitis, amylase and lipase "pregnancy" "pancreatitis" "pancreas"

21 A 63 year old male with biphenotypic ALL, Day 32 after BMT, h o CMV infection, aspergillus and Leggionare s disease, presents with acute onset of hypoxia accompanied by fever and two days of productive cough His CXR showed an opacification of the left basilar lobe and also right upper lobe concerning for pneumonia "morbidity" "mortality" "biopsy"

22 94 M with CAD s p 4v CABG, CHF, CRI presented with vfib arrest "mortality" "morbidity" "myocardial infarction"

23 85 yo M with PMH of colon CA s p resection now presenting with black stools and HCT drop "apoptosis" "survival" "recurrence"

24 51 years old male with multiple sclerosis and quadriplegia who presents with small bowel obstruction and low urinary output "spinal cord" "catheters" "placenta"

25 An elderly female with history of atrial fibrillation, Chronic Obstructive Pulmonary Disease, hypertension, hyperlipidemia and previous repair of atrial septum defect, presenting with shortness of breath and atrial fibrillation resistant to medication "sleep" "heart" "extremities"

26	A 79 year old female wit history of CAD, diastolic CHF, HTN, Hyperlipidemia, previous smoking history, and atrial fibrillation who presents for direct admission from home for progressive shortness of breath Patient denies recent palpitations, and reports that she has been compliant with all medications She admits to recent fatigue and 2 pillow orthopnea which has been present for months Patient underwent cardioversion and became hypotensive with a junctional rhythm requiring intubation She was placed on dobutamine Off of dobutamine, cardiac monitoring demonstrated a long QTc and an atrial escape rhythm "abstracts" "foot" "work"
27	A 96 y o female found unresponsive on ground at nursing home pressents with headache, herniation, and some neck shoulder discomfort CT head shows acute left subdural hematoma "tables" "character" "anemia"
28	An 84 year old man with a previous history of coronary artery disease, presenting with 2 days of melena and black colored emesis "morbidity" "humidity" "mortality"
29	This is a 54 year old male patient with an idiopathic pulmonary fibrosis presenting an acute dyspnea on exertion, secondary to superimposed pneumonia on patient with no pulmonary reserve Appears he has been experiencing worsening dyspnea with increased O2 requirement for the last several weeks "prevalence" "epidemiology" "heart"
30	An 85 year old woman on verapamil presents with junctional heart rhythm in 30s with associated hypotension "perfusion" "blood pressure" "calcium"

* original query texts (http://www.trec-cds.org/topics2016.xml) were listed, although there were some typos.

Appendix H. 30 queries followed by top 4 terms generated by the ensemble model based on Positive Word Selection for QE (mean scores - infAP: 0.0254 & infNDCG: 0.1939).

Topic No.	Query followed by four expanded terms (by the order of word score)
1	A 78 year old male presents with frequent stools and melena "hemorrhage" "rupture" "child" "histology"
2	An elderly female with past medical history of right hip arthroplasty presents after feeling a snap of her right leg and falling to the ground "osteoarthritis" "role" "work" "association"
3	A 75F found to be hypoglycemic with hypotension and bradycardia She had UA positive for klebsiella She had a leukocytosis to 18 and a creatinine of 6 Pt has blood cultures positive for group A streptococcus On the day of transfer her blood pressure dropped to the 60s She was anuric throughout the day, awake but drowsy This morning she had temp 963, respiratory rate 22, BP 102 26 "microbiology" "research" "role" "work"
4	An 87 yo woman with h o osteoporosis, DM2, dementia, depression, and anxiety presents s p fall with evidence of C2 fracture, chest pain, tachycardia, tachypnea, and low blood pressure "kidney" "research" "role" "work"
5	An 82 man with multiple chronic conditions and previous surgeries presents with 9 day history of productive cough, fever and dyspnea "role" "research" "work" "review"
6	A 94 year old female with hx recent PE DVT, atrial fibrillation, CAD presents with fever and abdominal pain An abdominal CT demonstrates a distended gallbladder with gallstones and biliary obstruction with several CBD stones "tomography" "research" "role" "work"
7	A 41 year old male patient with medical history of alcohol abuse, cholelithiasis, hypertension, obesity who presented to his local hospital with hematemasis, abdominal pain radiating to the back and elevated lipase Signs of ascites, pancytopenia and coagulopathy "disease" "research"
8	A 26 year old diabetic woman, estimated to 10 weeks pregnant, presents with hyperemesis Her labwork demonstrates a blood glucose of 160, bicarbonate of 11, beta hCG of 3373 and ketones in her urine "research" "role" "work" "growth"
9	Infant with respiratory distress syndrome and extreme prematurity Chest x ray shows diffuse bilateral opacities within the lungs, with increased lung volumes "research" "role" "work" "diagnosis"
10	A 55 year old woman with sarcoidosis, presenting today with confusion and worsening asterixis In the waiting room, the pt became more combative and then unresponsive Ammonia level 280 on admission "disease" "role" "research" "diagnosis"
11	80 yo male with demantia and past medical history of CABG with repeated episodes of chest pain Admitted for severe chest pain episode "pregnancy" "research" "role" "work"
12	66 yo female pedestrian struck by auto Unconscious and unresponsive at scene Multiple fractures and head CT showing extensive interparenchymal hemorrhages "mortality" "physicians" "incidence" "research"
13	A 43 year old woman with history of transverse myelitis leading to paraplegia, depression, frequent pressure ulcers, presenting with chills, agitation, rigors, and back pain Patient has

stage IV decubitus ulcers on coccyx and buttocks, heels Admission labs significant for thrombocytosis, elevated lactate, and prolonged PT "disease" "population" "research" "role"

14 A 52 year old woman with history of COPD and breast cancer who presents with SOB, hypoxia, cough, fevers and sore throat for several weeks "adult" "prevalence" "disease" "role"

15 67 yo male smoker with end stage COPD on home oxygen, tracheobronchomalacia, s p RUL resection for squamous cell carcinoma Y stent placement was complicated by cough and copious secretions requiring multiple therapeutic aspirations Patient reports decreased appetite, 50 lb wt loss in 6 months Decreased activity tolerance PET scan revealed some FDG avid nodes concerning for recurrence Pt presents with worsening SOB with R shoulder pain and weakness "tomography" "adenocarcinoma" "incidence" "research"

16 A 90 year old woman who was recently hospitalized for legionella PNA, with confusion and dysarthria the last few days Found down in the bathroom this morning, making non verbal utterances and with minimal movement of the right side "stroke" "reading" "injections" "central nervous system"

17 76 year old female with personal history of diastolic congestive heart failure, atrial fibrillation on Coumadin, presenting with low hematocrit and dyspnea "mortality" "research" "role" "association"

18 A 40 year old woman with a history of alcoholism complicated by Delirium Tremens and seizures 2 years ago, polysubstance abuse, hep C, presents with abdominal pain in lower quadrants, radiating to the back, nausea, vomitting and diarrhea Labs are significant for elevated lipase "research" "work" "review" "methods"

19 78 year old female with PMHx HTN, dCHF, Diabetes, CKD, Atrial fibrillation on coumadin, ischemic stroke, admitted after presenting with confusion and somnolence She was recently discharged after presyncope falls Patient has had confusion at home for 3 weeks The patient denies headache, blurry vision, numbness, tingling or weakness, nausea or vomiting "mortality" "morbidity" "awareness" "epidemiology"

20 A 87 yo female reports several days abdominal pain, worse yesterday, severe and more localized to the right, accompanied by nausea and vomitting Labs show elevated bilirubin, transaminitis, amylase and lipase "research" "role" "work" "methods"

21 A 63 year old male with biphenotypic ALL, Day 32 after BMT, h o CMV infection, aspergillus and Leggionare s disease, presents with acute onset of hypoxia accompanied by fever and two days of productive cough His CXR showed an opacification of the left basilar lobe and also right upper lobe concerning for pneumonia "mortality" "research" "role" "work"

22 94 M with CAD s p 4v CABG, CHF, CRI presented with vfib arrest "mortality" "heart" "research" "role"

23 85 yo M with PMH of colon CA s p resection now presenting with black stools and HCT drop "time" "research" "role" "liver"

24 51 years old male with multiple sclerosis and quadriplegia who presents with small bowel obstruction and low urinary output "mortality" "research" "role" "work"

25 An elderly female with history of atrial fibrillation, Chronic Obstructive Pulmonary Disease, hypertension, hyperlipidemia and previous repair of atrial septum defect, presenting with

	shortness of breath and atrial fibrillation resistant to medication "disease" "research" "role" "population"
26	A 79 year old female wit history of CAD, diastolic CHF, HTN, Hyperlipidemia, previous smoking history, and atrial fibrillation who presents for direct admission from home for progressive shortness of breath Patient denies recent palpitations, and reports that she has been compliant with all medications She admits to recent fatigue and 2 pillow orthopnea which has been present for months Patient underwent cardioversion and became hypotensive with a junctional rhythm requiring intubation She was placed on dobutamine Off of dobutamine, cardiac monitoring demonstrated a long QTc and an atrial escape rhythm "mortality" "research" "role" "work"
27	A 96 y o female found unresponsive on ground at nursing home pressents with headache, herniation, and some neck shoulder discomfort CT head shows acute left subdural hematoma "radiology" "surgeons" "tomography" "europe"
28	An 84 year old man with a previous history of coronary artery disease, presenting with 2 days of melena and black colored emesis "writing" "research" "role" "work"
29	This is a 54 year old male patient with an idiopathic pulmonary fibrosis presenting an acute dyspnea on exertion, secondary to superimposed pneumonia on patient with no pulmonary reserve Appears he has been experiencing worsening dyspnea with increased O2 requirement for the last several weeks "research" "lung" "role" "work"
30	An 85 year old woman on verapamil presents with junctional heart rhythm in 30s with associated hypotension "literature" "calcium" "blood" "work"

* original query texts (http://www.trec-cds.org/topics2016.xml) were listed, although there were some typos.

Appendix I. IR performance comparison of two classifiers for the WSW model: ANN vs. SVM

SVM has been a popular classifier before ANN is practically used by the development of high-performing computing resources. An SVM model was compared with an ANN classifier in terms of infAP and infNDCG. RBF (radial basis function) was applied for kernel function in training a binary SVM classifier. The SVM classifier showed a higher score slightly in the accuracy for the validation set, while F1 and AUC scores were lower (Table 38).

Table 38. Average accuracy, F1, and AUC scores for binary ANN classifiers for 30 queries

	Acc (train)	Acc (val_all)	Acc (val_pos)	Acc (val_neg)	w_F1 (train)	w_F1 (val_all)	AUC (train)	AUC (val_all)
ANN (2 layers * 700 nodes)	0.7494	0.7233	0.0549	0.9779	0.6649	**0.6414**	0.6297	**0.5772**
SVM	0.7453	**0.7290**	0.0177	0.9957	0.6429	0.6321	0.5672	0.5366

Mean infAP and infNDCG scores of 40 (LDA + binary SVM classifier) model based on Word Score Weighting were listed in Table 39. The average mean infAP and infNDCG scores were 0.02 and 0.1744, respectively, which are lower than the baseline run scores as well as the scores of the (LDA + ANN classifier) model (Figure 38 & 39).

Table 39. Mean infAP and infNDCG scores of 40 WSW (an LDA model + a binary SVM classifier) models for the top 2 retrieved documents (a maximum of the top 10 words for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0157	0.019	0.0227	0.018	0.0143	0.0175	0.0196	0.0176	0.0225	0.0198
infNDCG	0.1474	0.1596	0.1904	0.1656	0.1424	0.1676	0.172	0.1737	0.1825	0.1822

no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0187	0.018	0.0183	0.0179	0.0178	0.0244	**0.0255**	0.0228	0.0203	0.0203
infNDCG	0.1633	0.1691	0.1676	0.1692	0.176	**0.1963**	0.1944	0.1994	0.1726	0.1766

no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0182	0.0204	0.0215	0.021	0.0183	0.0224	0.0199	0.0219	0.0198	0.0214

infNDCG	0.1799	0.1551	0.1813	0.1809	0.1721	0.174	0.1722	0.1736	0.1685	0.179

no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0201	0.0198	0.0218	0.0216	0.0179	0.0199	0.02	0.0215	0.0224	0.0198
infNDCG	0.1698	0.1779	0.1727	0.1838	0.1745	0.1741	0.1821	0.1797	0.1838	0.1732

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

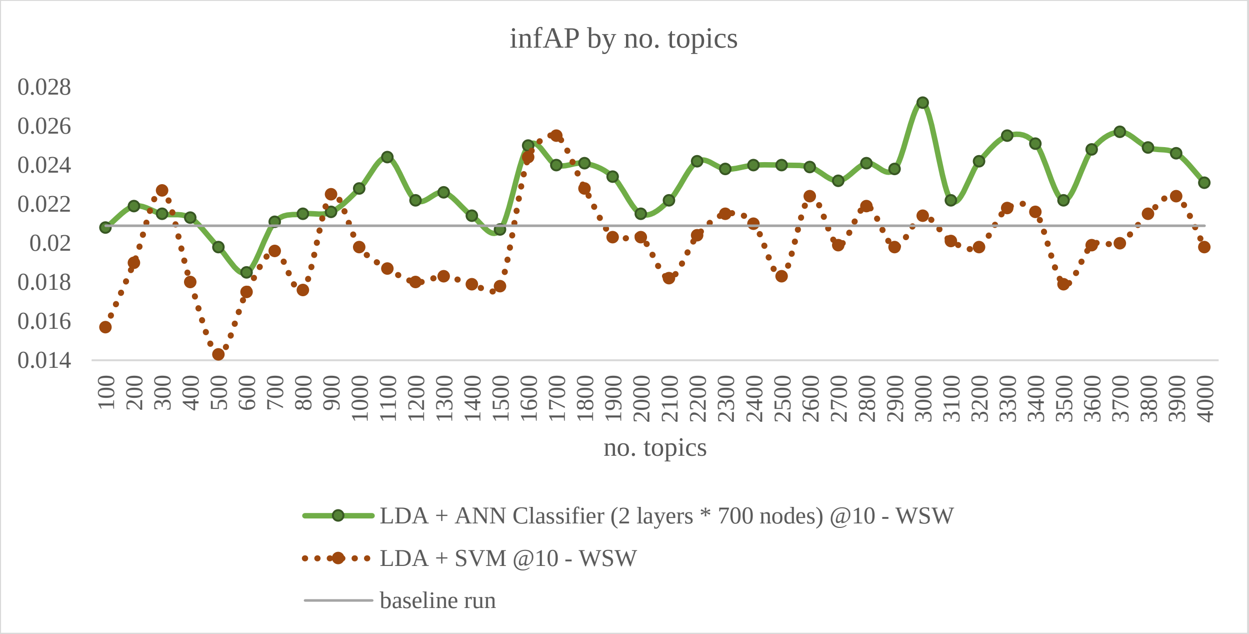

Figure 38. Mean infAP scores of 40 WSW (LDA + a binary classifier) models for the top 2 retrieved documents – SVM vs. ANN binary classifier (2 layers * 700 nodes)

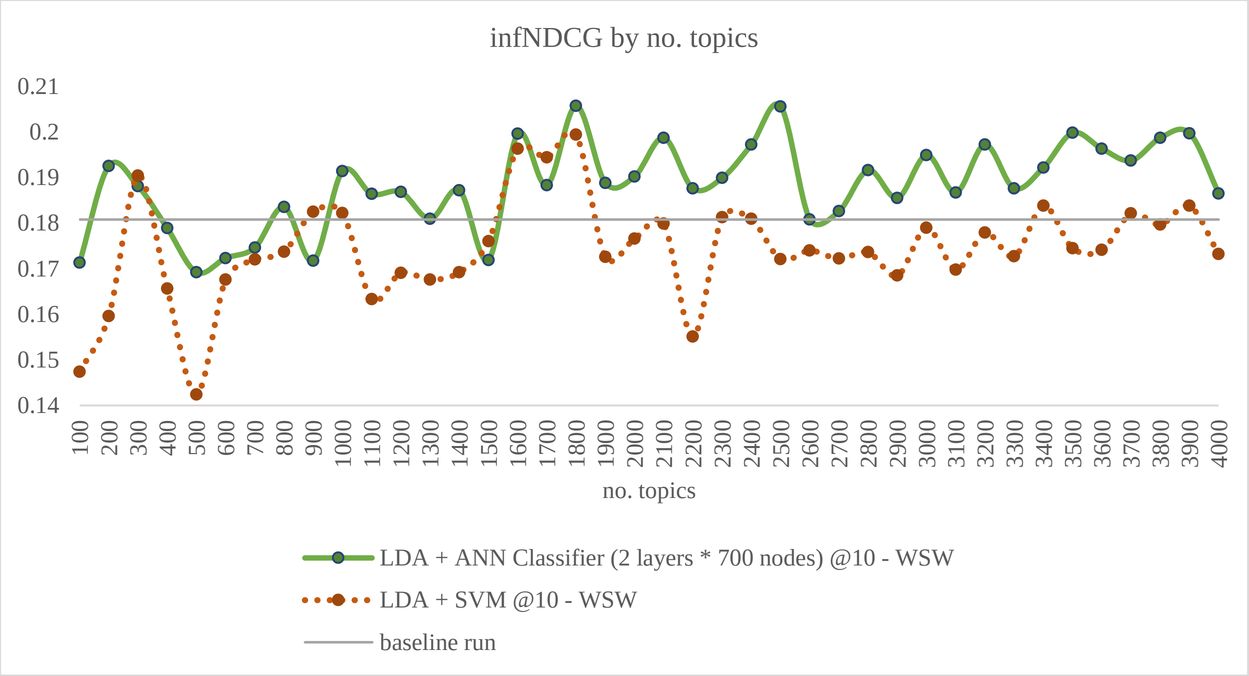

Figure 39. Mean infNDCG scores of 40 WSW (LDA + a binary classifier) models for the top 2 retrieved documents – SVM vs. ANN binary classifier (2 layers * 700 nodes)

Mean infAP and infNDCG scores of the PWS model using a binary SVM classifier (a maximum of the top 7 words for QE) were listed in Table 40 and compared in Figure 40 and 41. The average mean infAP and infNDCG scores were 0.0217 and 0.1828, respectively, which were statistically significantly higher (two-sample t-test, alpha = 0.05, p-value = 5.75E-12 for infAP and 6.06E-07 for infNDCG) than the baseline run scores even though a little bit lower than the scores (0.0224 for infAP and 0.1831 for infNDCG) of the PWS model using a binary ANN classifier. SVM also has the potential to increase infAP and infNDCG in health IR by identifying positive words.

Table 40. Mean infAP and infNDCG scores of 40 PWS (LDA + a binary SVM classifier) models for the top 2 retrieved documents (a maximum of the top 7 words for QE)

no. topics	100	200	300	400	500	600	700	800	900	1000
infAP	0.0214	0.021	0.0217	0.0211	0.0198	0.0212	0.0221	0.0207	0.0209	**0.0225**
infNDCG	0.181	0.1805	0.1839	0.1794	0.1745	0.1825	0.1835	0.181	0.1803	0.1804

no. topics	1100	1200	1300	1400	1500	1600	1700	1800	1900	2000
infAP	0.0207	0.0219	0.022	0.022	0.021	0.0222	0.0222	0.0221	0.0212	0.0212
infNDCG	0.1793	0.1839	0.1853	0.185	0.181	0.1853	0.185	0.1862	0.1855	0.1815

no. topics	2100	2200	2300	2400	2500	2600	2700	2800	2900	3000
infAP	0.0222	0.0213	0.0221	0.0221	0.0222	0.0221	0.0221	0.0221	0.0221	0.022
infNDCG	0.1853	0.1816	0.1846	0.1846	**0.1866**	0.184	0.184	0.1836	0.184	0.1828

no. topics	3100	3200	3300	3400	3500	3600	3700	3800	3900	4000
infAP	0.0221	0.0217	0.0217	0.0221	0.021	0.0217	0.0221	0.0218	0.0217	0.0209
infNDCG	0.1841	0.1829	0.1824	0.1836	0.1802	0.1829	0.1836	0.1831	0.183	0.1806

* baseline run (infAP: 0.0209 and infNDCG: 0.1808)

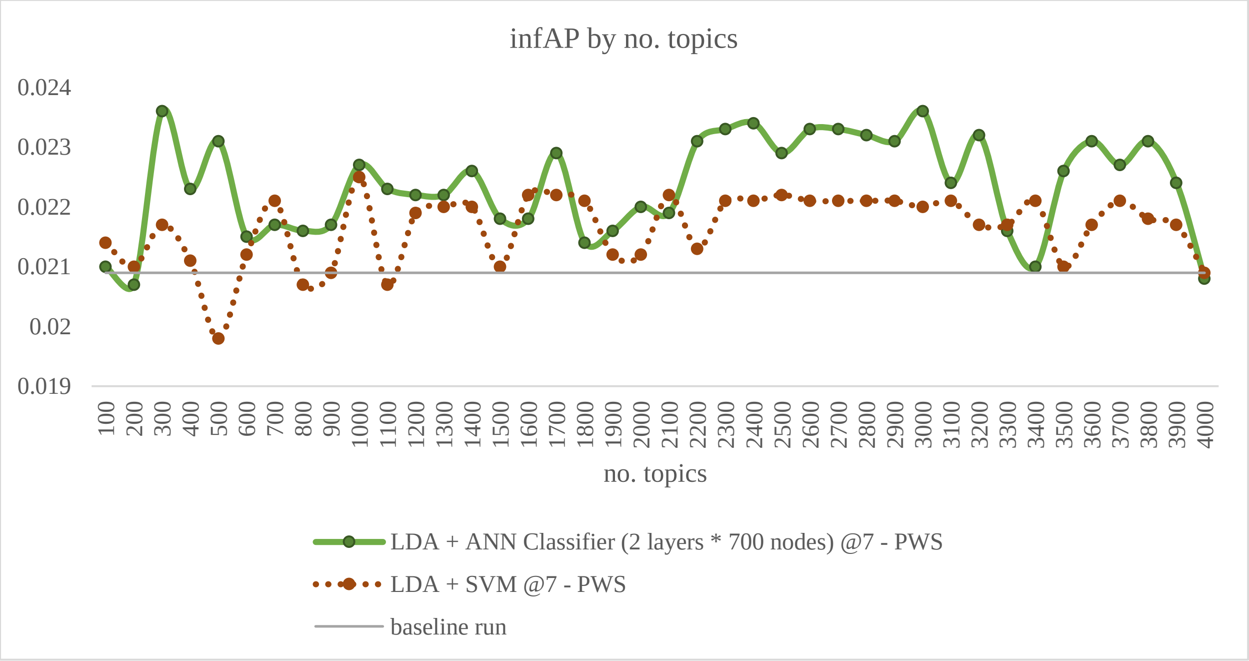

Figure 40. Mean infAP scores of 40 PWS (LDA + a binary classifier) models for the top 2 retrieved documents – SVM vs. ANN binary classifier (2 layers * 700 nodes)

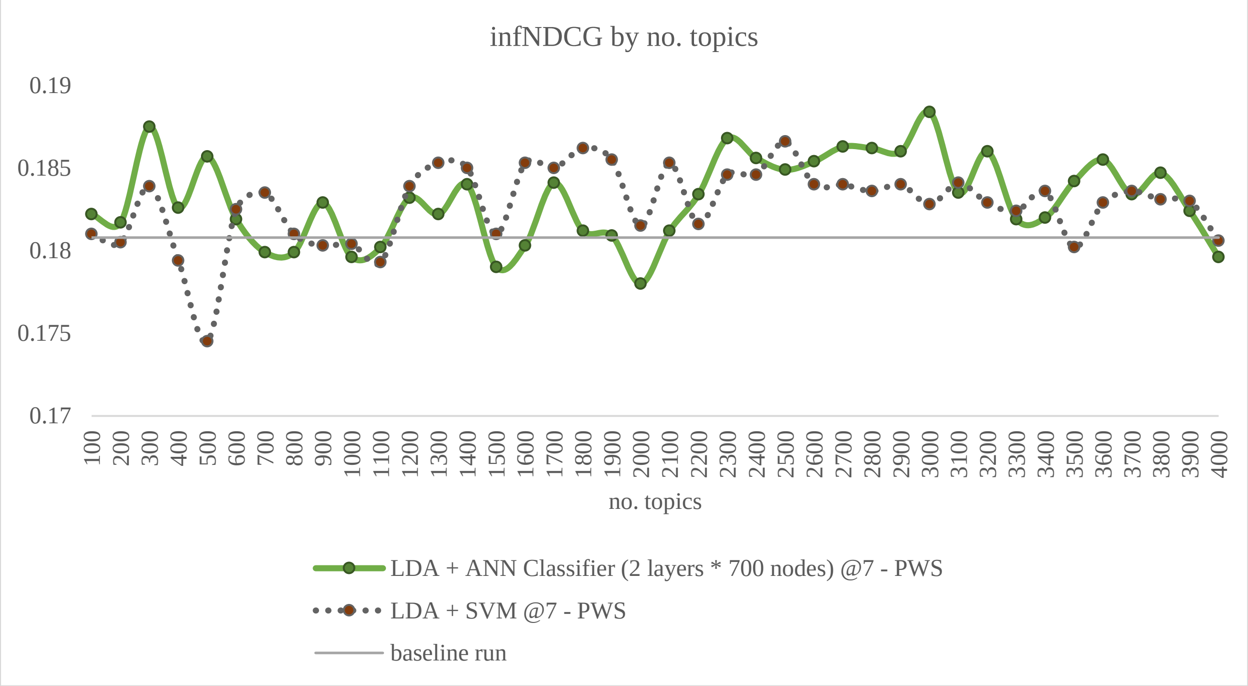

Figure 41. Mean infAP scores of 40 PWS (LDA + a binary classifier) models for the top 2 retrieved documents – SVM vs. ANN binary classifier (2 layers * 700 nodes)

CURRICULUM VITAE

Sukjin You

Place of birth: Incheon, South Korea

Education

B.S., Inha University, February 2003
Major: Computer Science & Engineering

B.A., Yonsei University, February 2012
Major: Library & Information Science

M.L.I.S., University of Wisconsin-Milwaukee, May 2014

Dissertation Title: The Ensemble MeSH-Term Query Expansion Models Using Multiple LDA Topic Models and ANN Classifiers in Health Information Retrieval

JOURNAL PUBLICATIONS

Xie, I., Babu, R., Lee, T. H., Castillo, M. D., **You, S.**, & Hanlon, A. M. (2020). Enhancing usability of digital libraries: Designing help features to support blind and visually impaired users. *Information Processing & Management*, *57*(3), 102110.

Park, H., **You, S.**, & Wolfram, D. (2018). Informal data citation for data sharing and reuse is more common than formal data citation in biomedical fields. *Journal of the Association for Information Science and Technology*, *69*(11), 1346–1354.

Wang, P., **You, S.**, Manasa, R., & Wolfram, D. (2016). Open peer review in scientific publishing: A Web mining study of PeerJ authors and reviewers. *Journal of Data and Information Science, 1*(4), 60–80.

CONFERENCE PROCEEDINGS

Xie, I., Babu, R., Castillo, M. D., Lee, T. H., & **You, S.** (2018, October). Developing Digital Library Design Guidelines to Support Blind Users. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility* (pp. 401–403).

Park, H., **You, S.**, & Wolfram, D. (2017). Is informal data citation for data sharing and re-use more common than formal data citation? *Proceedings of the Association for Information Science and Technology*, *54*(1), 768–769.

You, S., Huang, W., & Mu, X. (2015, December). Using Event Identification Algorithm (EIA) to improve microblog retrieval effectiveness. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)* (Vol. 3, pp. 122–125). IEEE.

Mu, X., & **You, S.** (2015). TREC 2015 paper submission UWM-UO@ 2015 Clinical Decision Support Track: QE by Weighted Keywords using PRF. In *TREC.*

You, S., Huang, W., & Mu, X. (2014). *UWM-HBUT at TREC 2014 Microblog Track: Using Query Expansion (QE) and Event Identification Algorithm (EIA) to improve microblog retrieval effectiveness*. WISCONSIN UNIV-MILWAUKEE. In *TREC.*

You, S., DesArmo, J., Mu, X., Lee, S., & Neal, J. C. (2014, September). Visualized Related Topics (VRT) system for health information retrieval. In *IEEE/ACM Joint Conference on Digital Libraries* (pp. 429-430). IEEE.

You, S., DesArmo, J., Mu, X., & Dimitroff, A. (2014, September). Balancing factors affecting virtual reference services: identified from academic librarians' perspective. In *IEEE/ACM Joint Conference on Digital Libraries* (pp. 477–478). IEEE.

DesArmo, J., **You, S.**, Mu, X., & Dimitroff, A. (2014). Situational virtual reference: Get help when you need it. *iConference 2014 Proceedings*.

You, S., DesArmo, J., & Joo, S. (2013). Measuring happiness of US cities by mining user-generated text in Flickr.com: A pilot analysis. *Proceedings of the American Society for Information Science and Technology*, *50*(1), 1–4.

TEACHING EXPERIENCE

Instructor (School of Information Studies at UWM)

INFOST 350 – Introduction to Application Development (Fall 2018, Onsite, undergraduate)

This course acquaints students with the core concepts of software development from an Information Studies perspective. Students learn how to develop basic software using the Python programming language that can be applied to further coursework and careers in application development and information technology.

WORK EXPERIENCES

LG Electronics, Seoul in Korea, senior research engineer from March 2003 to April 2009: Mobile application programming and middleware API implementation.

XML schema design project with Dr. Mu for IPC (Institute for Interconnecting and Packaging Electronic Circuits, 2012–2013).

Digital library project at UWM libraries using content management tools - CONTENTdm, Omeka, and WordPress (internship).