San Francisco State University (SFSU) at Total Recall Track of TREC 2016

Mon-Shih Chuang, Anagha Kulkarni
mchuang@mail.sfsu.edu, ak@sfsu.edu
San Francisco State University

Abstract

This paper describes the participation of San Francisco State University group in Text Retrieval Conference (TREC) 2016 Total Recall Track from National Institute of Standard and Technology (NIST).

The TREC series provide large test collections and judgements for participant to design Information Retrieval (IR) systems for different proposes. The purpose of Total Recall Track is seeking text search system which achieves high recall with minimum number of return documents.

This year, our team participates all automatic tasks, including 34 topics in athome task and 2 datasets in sandbox task.

Our system is built based on the autonomous technology-assisted review (Auto TAR) model[1], which is also the baseline of Total Recall Track. In this paper, we will introduce several approaches which have improved the evaluation metrics compare to the baseline model. Our enhanced model combines seed expansion and feature engineering including adding n-gram, eliminating stop words, and preserving words contain digits.

1 Introduction

The objectives of Total Recall Track came from the technology-assisted review (TAR) problem, which is "the iterative retrieval and review of documents from a collection until a substantial majority or all of the relevant documents have been reviewed." [1] The goal of TAR is to maximize effectiveness of creating test collection for IR evaluation. That is, a optimal document selection algorithm to (1) send all relevant documents to reviewers with minimum number of returns (2) decide when to stop reviewing for reviewers. From TAR, Cormack and Grossman came up with autonomous technology-assisted review (Auto TAR) model, which is also the baseline model of Total Recall Track. The algorithm of baseline model will be described in section 2.

In Total Recall Track, the position of human reviewer is replaced by automated relevance assessor with pre-processed relevant judgements to evaluate systems from participants.

The objectives for participants are the inherited from TAR problem, "to submit as many documents containing relevant information as possible, while submitting as few documents as possible", and "indicate when the submission is reasonable to stop, because the effort to review more documents would be disproportionate to the value of any further relevant documents that might be found."

The first objective is evaluated by recall-precision curves, gain curves, and recall evaluated at aR+b documents submitted, for all combinations of a = 1, 2, 4 and b = 0, 100, 1000. R is number of total relevant documents for each topic. The second objective requires the participants to use "call your shot" API at the point that system should stop. Then the recall and precision at this point will be evaluated using measures like F1 and other utility measures. In section 3, our methodology to achieve these two objectives will be explained. Our system, is tested on athome1, athome2, athome3 datasets and topics, which are all provided by Total Recall Track coordi-

nators. The experiment result and analysis will be demonstrated in section 4.

2 Baseline Retrieval Model

The baseline system provided by track organizer adopts a supervised classification approach within the framework of continuous active learning. We provide a brief description of the baseline system here because our proposed approach builds on it. The provided baseline model implementation (BMI) adopts an iterative approach that uses support vector machine (SVM) to learn document classification models that label each candidate document as relevant or non-relevant. Specifically, the BMI approach proceeds as follows.

Preprocessing step: For each document in the collection, parse, and transform it into a tfidf feature vector.

For each query topic:

1. Label the topic itself as the first data-point from *relevant* class, and add it to the training set.
2. Use uniform sampling to select 100 documents at random from the collection, and add them to the training set as data-points from the *non-relevant* class. Let D: set of 100 sampled documents.
3. Learn a classification model using SVM and the compiled training set.
4. Apply the learned model to predict relevance label for each document not in the training set.
5. Sort the documents using weight returned from SVM, whose value represents the distance from data-points to the decision hyperplane, and with sign '+' for *relevant* class, '−' for *non-relevant* class. Then select L documents with highest weights, and request their actual relevance labels. L is initialized to 1, and increases by L/10 at every iteration.
6. Add the reviewed documents to the training set.
7. Remove documents from set D (step 2) that were not selected for review.
8. Go back to step 2 until 100 iteration are complete.
9. Send all unreviewed documents to be judged, then stop.

3 Methodology

3.1 Reducing Increasing Rate of Batch Size

After each iteration of re-classification, BMI sends a batch of documents to the reviewer for labelling. A drawback of sending a batch of documents at-a-time is that documents with similar contents are sent for labelling in the same iteration. This wastes labelling effort. For instance, requesting labels for all the duplicates (or near-duplicates) of a document is unnecessary. If we know the label for one of the copies, then all the duplicates will have high chance to be classified with the same label. In the current system this problem of wasted labelling effort occurs because documents with similar contents are likely to have same weight, and comparable ranks, and thus they tend to be sent in the same iteration. There are several reasons why duplicate documents occur in a collection. In athome1, which is an email collection, an email may be sent to multiple receivers, or an email received may be forwarded, both of which results in multiple copies of a document. In athome3, which is a local news collection, one article can be referenced by different news site, and the same event may be reported by multiple news site in slightly different words.

The problem of wasted effort can be completed eliminated by sending only one document at a time for labelling. However, one-document-at-a-time approach is highly inefficient. For instance, 300,000 iterations of re-classification will be required to process the athome1 dataset, which translates to prohibitively high runtime.

These observations motivate our experiments where we model the batch-size (L) as a parameter, and investigate its influence on effectiveness and efficiency. The baseline approach increases the batch-size, L, by L/10 after each iteration. In our runs, we

increase the batch-size by L/12, L/15, and L/20. The rate at which the batch-size is increased is progressively slower for the three settings. We expect this to lower the wastage of labelling effort.

3.2 Seed Expansion using Wikipedia

In BMI, seed refers to the documents labeled as relevant before the first iteration. BMI uses the topic itself as the seed document, thus the typical length of a seed "document is between 2 to 5 words. Before the first relevant document is retrieved from the collection, the seed document is the only information about the users information need(s). For some topic like 109 "Scarlet Letter Law" and 2130 "Surely Bitcoins Can Be Used", the BMI retrieves more than 30 non-relevant documents before the first relevant document is retrieved, which hurts the precision significantly. In such topics, the terms in the topic do not provide enough information about the information need(s). This inspires our approach for enhancing the seed document. To expand information before any document is retrieved, we choose Wikipedia[2] for our external source. Wikipedia is an online collaborative encyclopedia which provides a wide coverage of topics and events. Since it a well curated, high-quality resource, it has been utilized for many problems such as clustering[3], question answering[4], and patent search[5]. A Wikipedia page is organized as data fields including title, url, summary, content, images, and links. In our approach, we conduct two different runs, one using Wikipedia summary, and another using the content data field as the seed document. In our approach, we send the original topic as the query to Wikipedia search API. If multiple pages are returned, only the top one Wikipedia page is used. If the search API returns an ambiguity page, then no Wikipedia source is used. The summary/content field of the Wikipedia page and original topic are combined to expanded seed. The expanded seed will remain in the training set for every iteration of reclassification.

3.3 Unigram/Bigram SVM Features

This approach is inspired by the concept of phrase search. When the query contains more than one term, often all of some of these terms form a phrase. For example, the following query *barack obama white house* contains two phrases *barack obama* and *white house*. Data analysis of the athome1 task revels that the relevant documents usually contain the query phrases, while the non-relevant documents contain only single terms, and often these single terms convey slightly different meaning from the query topic. The main reason for this pattern is word ambiguity. Individual words are often ambiguous, but the other words in the phrase help resolve the ambiguity. For instance, topic 103 "Manatee Protection" and topic 108 "Manatee County", both contain the word "Manatee", but the former refers to the name of a kind of mammal, the later refers the name of a county in Florida.

Since BMI use unigram inverted index and stores no term position information, it cannot support phrasal queries. As such, we add bigram features in our model. It is important to note that unigrams are necessary to achieve high recall. Not all the relevant documents contain the query phrases. Thus we use both, bigram and unigram features in our model. Unigrams to boost recall and bigrams to improve precision. The query topics are also convert to the seed documents with both unigram and bigram features.

While bigram model improves the precision of retrieving relevant documents, the runtime efficiency significantly drops since the feature space explodes. In our implementation, the bigram/unigram model contains 3 million features in total while the original unigram model only contains 160,000 features.

3.4 Feature Pruning

To balance the efficiency and effectiveness of our bigram implementation, we tried the following two simple feature pruning techniques. (1) Prune rare ngram. DF $< x$. We experimented with cutoff of 3 and 5. (2) Prune ngram that contain one or more English stop words. A 25 word list provided in the Introduction to Information Retrieval book.[6] was

Topic	Information Needs	Title of the top ranked Wiki page
athome100	School and Preschool Funding	Preschool
athome101	Judicial Selection	Judicial Nominating Commission
athome102	Capital Punishment	Capital Punishment
athome103	Manatee Protection	(No Page Found)
athome104	New Medical School	New Jersey Medical School
athome105	Affirmative Action	Affirmative Action
athome106	Terri Schiavo	Terri Schiavo case
athome107	Tort Reform	Tort Reform
athome108	Manatee County	Manatee County, Florida
athome109	Scarlet Letter Law	Pete Schneider

Table 1: Returned results of Wikipedia Search API using original information of topic for query.

used for feature pruning.

The 25 pruned stop words used are : "a", "an", "and", "are", "as", "at", "be", "by", "for", "from", "has", "he", "in", "is", "it", "its", "of", "on", "that", "the", "to", "was", "were", "will", "with"

3.5 Seed Expansion using Google Word2Vec

One of the major challenges for improving recall is the classic problem of vocabulary gap. Since most topics are short (2 to 5 words) many relevant documents do not contain any of the terms mentioned in the topic. That is, there is a gap in the vocabulary that generated the topic versus the one that generated such relevant documents. In order to bridge this gap we develop an approach that expands each topic with related words. To identify the set of related words for a topic we employ Word2Vec[7], a popular approach introduced by Google that computes vector representation of words/phrases using neural network.

The objective of Word2Vec neural network is to optimize prediction of nearby words by converting them to similar vectors. For example, in the training document, if the word "Paris" and the word "France" have very high possibility to occurs near to each other, the cosine of angle between the computed vectors representation of them will closer to positive 1.

In our approach, the corpus of current task was used as the training data for Word2Vec. Therefore, the prediction of surrounding words is computed according to the behavior of the dataset. Since Word2Vec doesn't apply any stemming for the training document, the dataset has to be preprocessed with porter stemmer.

To combine this approach with our bigram model mentioned in section 3.3, we add bigram information to the training text by concating adjacent words together by underscore '_'. For example, phrases *"scarlet letter law"* will be convert to *scarlet_letter, letter_law* after applying the bigram converter.

Finally, we use each topic as input of Word2Vec distance predicting tool. The distance predicting tool imports the vectors representation generated by neural network, and computes top 40 words/phrases having highest chance occurs nearby the topics.

In our experiments, we set the cosine score ≥ 0.5 as the threshold for adding terms into the seed document. If there are no terms with score ≥ 0.5, then the topic is not expanded. For topic 106, all returned terms by Word2Vec having cosine score ≥ 0.5. Table (2) shows the top 1 terms selected using this approach with bigram/stop-words-pruning training text for each topic in athome1.

From some of the words/phrases returned this way, we can see their clear relationship with the original information need(s). For example, the top ranked result from topic 105 "Affirmative Action" is "elimin_race", which is related to the information need(s). For topic 108 "Manatee County", the top ranked result is "escambia_counti", which is another county in Florida state.

Topic	Information Needs	1st result
athome100	School and Preschool Funding	(None)
athome101	Judicial Selection	judici_appoint
athome102	Capital Punishment	believ_separ
athome103	Manatee Protection	protect_plan
athome104	New Medical School	school_enter
athome105	Affirmative Action	elimin_race
athome106	Terri Schiavo	terri_schindler
athome107	Tort Reform	reform_bill
athome108	Manatee County	escambia_counti
athome109	Scarlet Letter Law	dai_scarlet

Table 2: The First Word/Phrase returned from each topic as input of Word2Vec distance tool, using bigram/stop-words-pruning athome1collection for training.

3.6 Preserve Words Contain Digits

While analyzing the relevant documents for each athome1 topic, we found the topic 109 "Scarlet Letter Law" is another name of "House Bill 141", and most of relevant documents of topic 109 contain only the phrase "HB 141" but don't contain any word in the topic.

Inspired by this characteristic of topic 109, and the fact that BMI skips words contain digits[0-9] when building the inverted index of corpus. We expect preserving words contain digits or pure numbers can improve the recall of BMI. The possible drawback of keeping numbers is that numbers can hold much more different kind of meanings than English words, which leads to retrieve non-relevant documents. For this reason, we combine this approach only with bigram model. As bigram ,the meaning of the numbers like "141" in "HB 141" can be determine from the context, we expect keeping numbers won't have much negative effect on precision.

4 Experiments and Analysis

4.1 Reducing Increasing Rate of Batch Size

Figure 1. shows the precision-recall curve before and after reducing the increasing rate of batch size. The precision at every certain recall point is improved

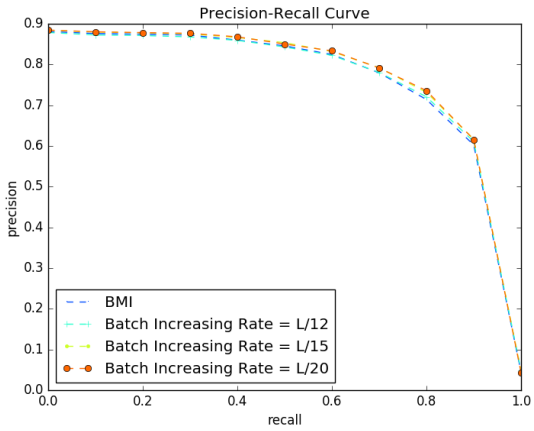

Figure 1: PR-curve of Reduce Batch Increasing Rate Experiments.

while the increasing rate gets smaller except the one at 100% recall. The reason of this approach has no improvement on the late stage is that we don't increase the number of iteration, and with smaller batch size, our runs have more unreviewed documents than BMI in the last iteration. Which result in worse precisions on the late stage.

If a model is able to retrieve most of the relevant documents before the last iteration, it will have less drawback from reducing the increasing rate of batch size. In other words, the better a model performs

on precision , the effect of tuning increasing rate of batch size becomes more viable.

4.2 Seed Expansions

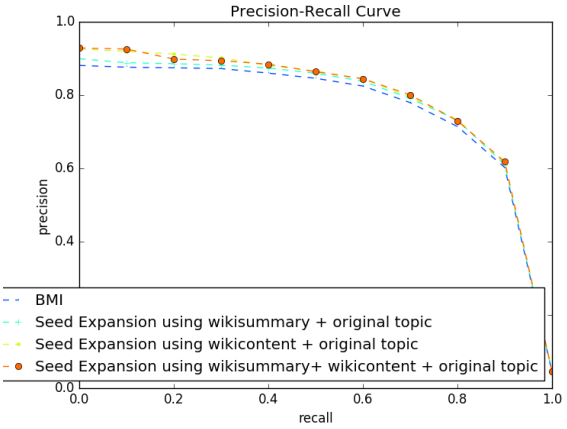

Figure 2: PR-curve of Wikipedia Experiments.

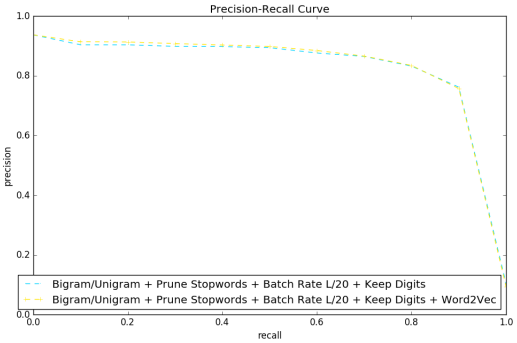

Figure 3: PR-curve of Word2Vec Experiments.

Figure 2. shows the Precision-Recall curve after combining Wikipedia summary/content and the original topic. Figure 3. shows the Precision-Recall curve after combining words/phrases returned by Word2Vec distance predicting tool and the original topic. Note we only apply the Word2Vec approach over the bigram model.

From either the experiment using Wikipedia or Word2Vec, we observed seed expansion is especially helpful to improve precision in the early stage. This observation matches our goal of the seed expansion approaches, which is expanding information need(s) before any review effort is spent. But although these 2 seed expansion approaches can retrieve the first relevant documents earlier than BMI, they don't improve the precision on late stage. The reason we is that after more documents have been reviewed and labelled, the influence of the seed documents becomes relatively smaller in the training set of classification.

4.3 Unigram/Bigram SVM Features

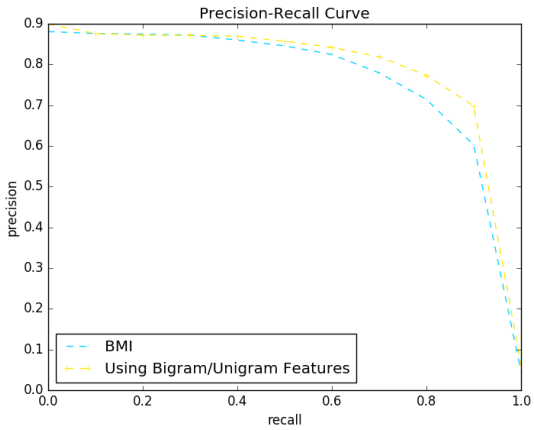

Figure 4: PR-curve of Bigram Experiments.

Figure 4. shows the precision-recall curve before and after adding bigram features. Contrast to expansing seed document, which has less effect after more documents are reviewed and labelled, the approach of enhancing SVM features start to having positive effect as more documents are added to the training set. The improvement is especially significant on the precisions after the recall is greater than 70%.

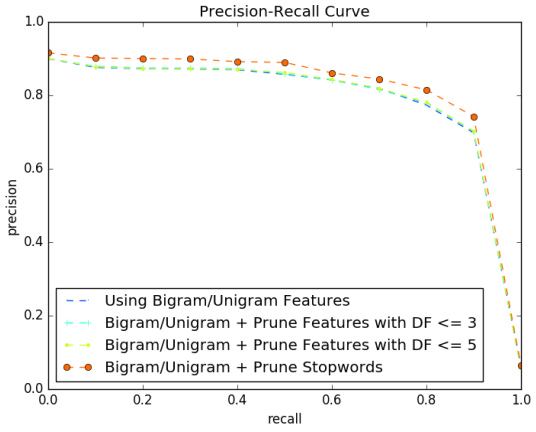

Figure 5: PR-curve of Feature Pruning Experiments.

4.4 Feature Pruning

Figure 5. shows the Precision recall curve before and after pruning features with DF$\leq$ 3, DF $\leq$ 5, and features contain stop words.

The feature pruning is not included in our approaches of improving evaluation metrics at first. Our initial objective of feature pruning is to reduce the running time of the bigram model while having least impact on the evaluate metrics. Although we have expected eliminating less representative features will result in slightly drops on all of the evaluate metrics, in our experiment runs, we found the run of eliminating stop words has better result on evaluation metrics.

By further analysis on the improved metrics topic by topic, we found the improvement on topic 109 is the main reason of the better metrics after removing stop words. As mentioned in the section 3.6, BMI performs bad on topic 109 "Scarlet Letter Law" since it fails to find documents contain "HB 141" on early stage. After removing stop words from the features, our model retrieves the documents contains phrase "HB 141" earlier than BMI. The reason is that removing stop words prevents the classifier comparing less meaningful common words, and can reduce some misclassification happened this way.

4.5 Preserving Words Contain Digits

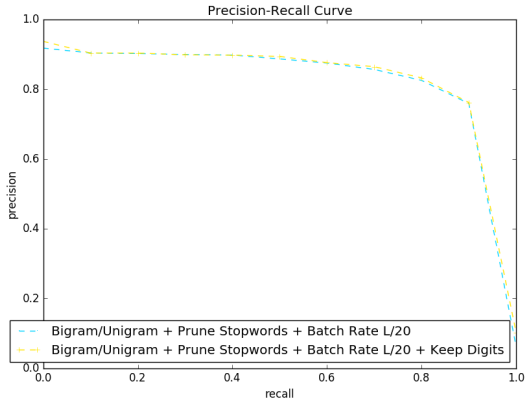

Figure 6: PR-curve of Preserving Digits experiments.

Figure 6. shows the precision-recall curve before and after preserving words contain digits. This approach leads to a small improvement of the metrics either on early stage or late stage. As we expected, since it is capable to determine the meaning of numbers from the bigram context, keeping digits doesn't have negative effects on the evaluation metrics.

4.6 The Compilation Experiment Run

Our compilation run combines (1) Reducing Increasing Rate of Batch Size to 20/L. (2) Seed Expansion using Wikipedia Summary. (3) Seed Expansion using Wikipedia Content. (4) Seed Expansion using Google Word2Vec. (5) Combining Bigram and Unigram Features (6) Pruning stop words. (7) Preserving words contain digits.

Figure 7. shows the precision-recall curve of BMI and our experiment run. Figure 8. shows the gain curve of BMI and our experiment run. Table 3. shows the recall when retrieved aR +b documents for all combinations of a = 1, 2, 4 and b = 0, 100, 1000 of BMI and our experiment run, and the percentage of improvement from BMI to our run. Figure 9. to Figure 17. show the precision-recall curve of BMI and our experiment run for each topic in athome1 task.

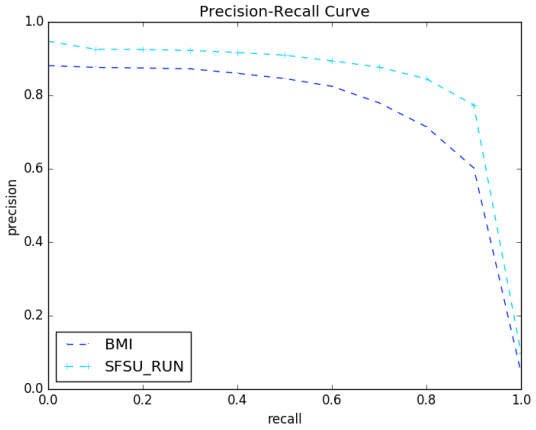

Figure 7: PR-curve of BMI and our compilation run.

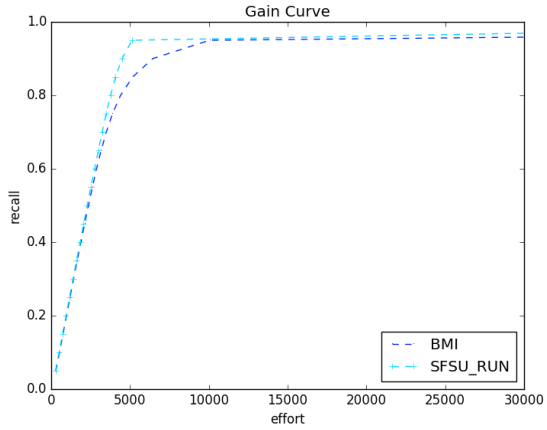

Figure 8: Gain-curve of BMI and our compilation run.

5 Conclusion

Our experiments shows combining the methodology of reducing increasing rate of batch size, seed expansion using Wikipedia source and Google Word2Vec tool sets, feature engineering include adding bigram features, removing stop words, preseving words contains digits will achieve very high recall, and outperforms BMI in every metrics used in overview paper 2015[8] for athome1 task. Our compilation methodology run is also applied to all tasks in Total Recall Track 2016, and we are expecting a good result compares with the other participants this year.

References

[1] Gordon V. Cormack and Maura R. Grossman. Autonomy and reliability of continuous active learning for technology-assisted review. *CoRR*, abs/1504.06868, 2015.

[2] Wikipedia. Wikipedia, the free encyclopedia. `https://en.wikipedia.org/`, 2001.

[3] Banerjee. S, Ramanathan. K, and Gupta. A. Clustering short texts using wikipedia. *SIGIR '07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788, 2007.

[4] Ganesh. S and Varma. V. Exploiting structure and content of wikipedia for query expansion in the context of question answering. *Proceedings of the international conference on recent advances in natural language processing(RANLP)*, pages 103–106, 2009.

[5] Al-Shboul. B and Myaeng. S. H. Wikipedia-based query phrase expansion in patent class search. *Information Retrieval*, 17:430–451, 2014.

[6] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2009.

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[8] A. Roegiest, G. V. Cormack, M. R. Grossman, and C. L.A. Clarke. Draft trec 2015 total recall track overview. 2016.

Recall	BMI	SFSU_RUN	Percentage of Improvement
a=1, b=0	0.7144	0.8064	12.88%
a=1, b=100	0.7408	0.8507	14.84%
a=1, b=1000	0.9039	0.9628	6.52%
a=2, b=0	0.9050	0.9751	7.75%
a=2, b=100	0.9191	0.9809	6.72%
a=2, b=1000	0.9548	0.9890	3.58%
a=4, b=0	0.9681	0.9917	2.44%
a=4, b=100	0.9682	0.9926	2.52%
a=4, b=1000	0.9738	0.9936	2.03%

Table 3: Recall at aR+b when a = 1, 2, 4 and b = 0, 100, 1000 of BMI and our compilation run

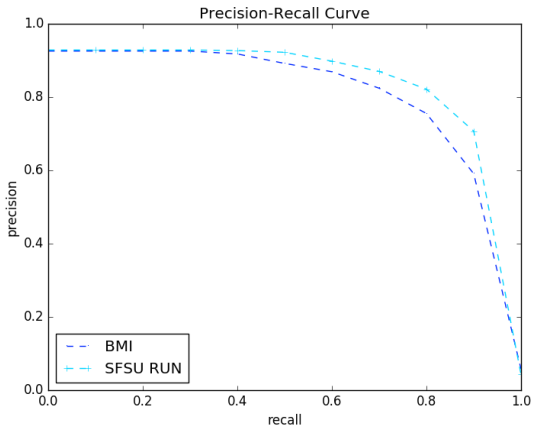

Figure 9: PR-curve of BMI and our compilation run on for topic athome100.

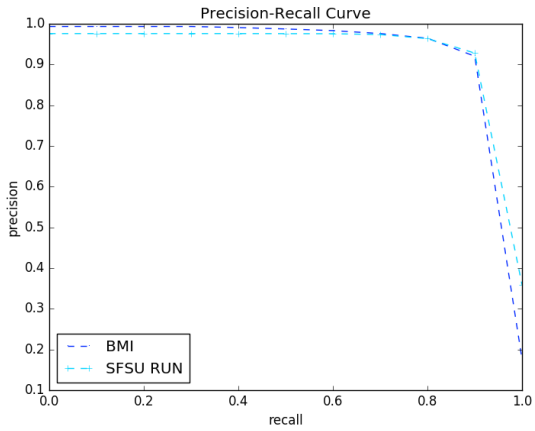

Figure 10: PR-curve of BMI and our compilation run on for topic athome101.

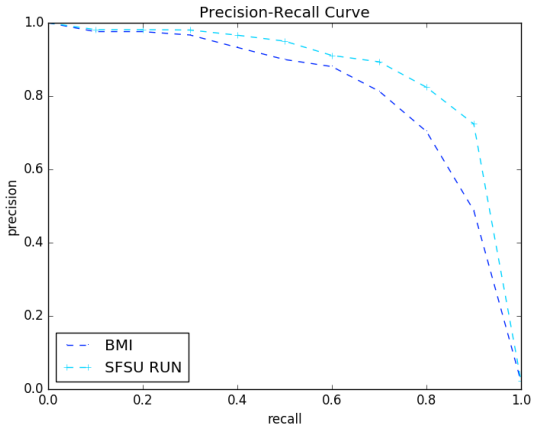

Figure 11: PR-curve of BMI and our compilation run on for topic athome102.

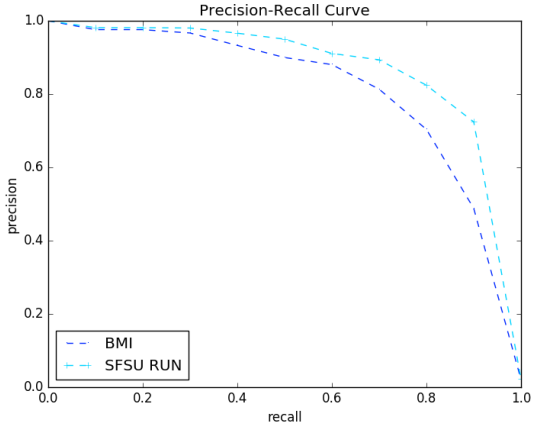

Figure 12: PR-curve of BMI and our compilation run on for topic athome103.

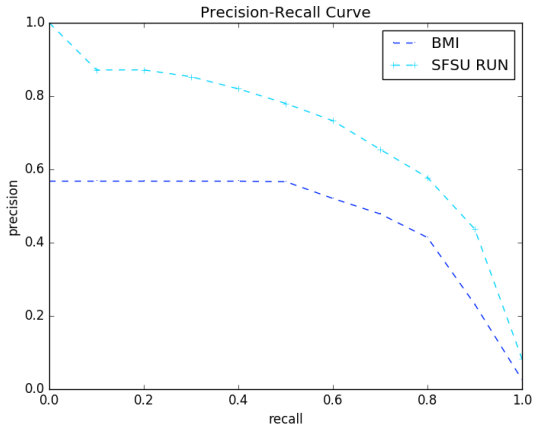

Figure 13: PR-curve of BMI and our compilation run on for topic athome104.

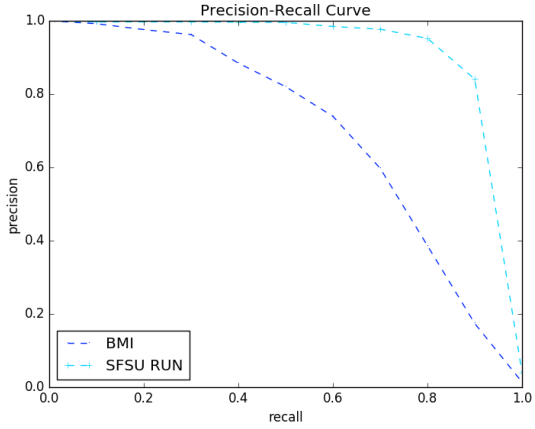

Figure 14: PR-curve of BMI and our compilation run on for topic athome105.

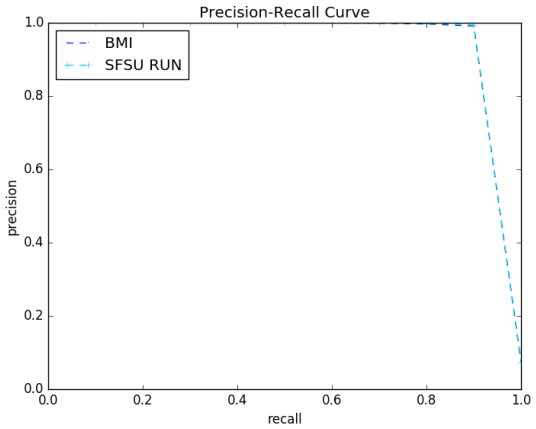

Figure 15: PR-curve of BMI and our compilation run on for topic athome106.

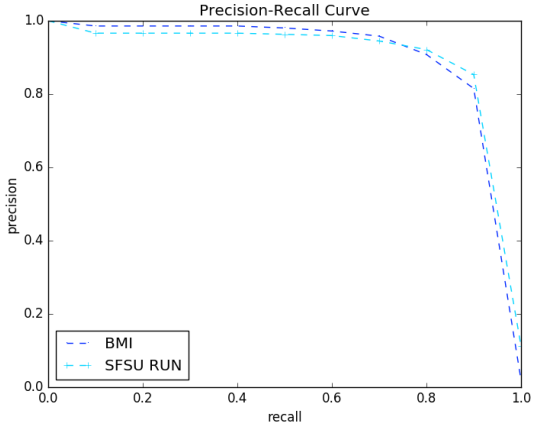

Figure 16: PR-curve of BMI and our compilation run on for topic athome107.

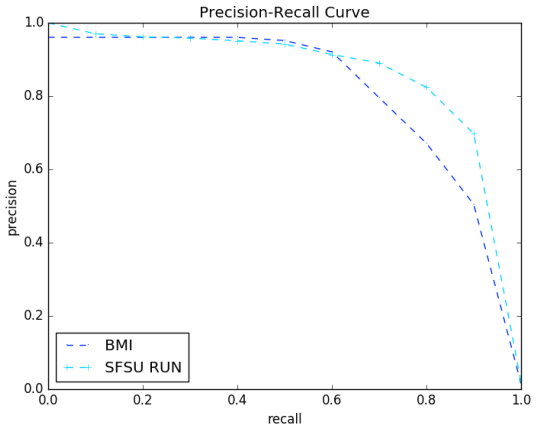

Figure 17: PR-curve of BMI and our compilation run on for topic athome108.

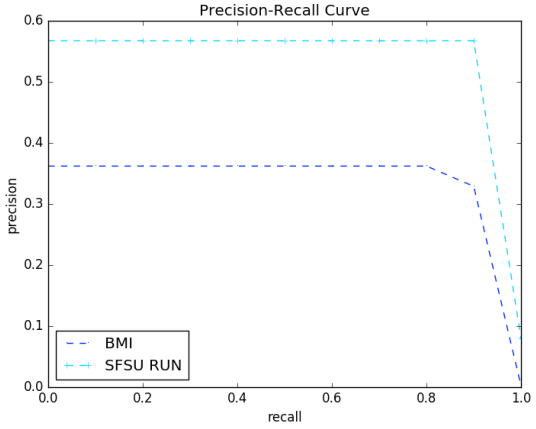

Figure 18: PR-curve of BMI and our compilation run on for topic athome109.