WaterlooClarke: TREC 2015 Total Recall Track

Haotian Zhang, Wu Lin, Yipeng Wang,
Charles L. A. Clarke and Mark D. Smucker

Data System Group
University of Waterloo

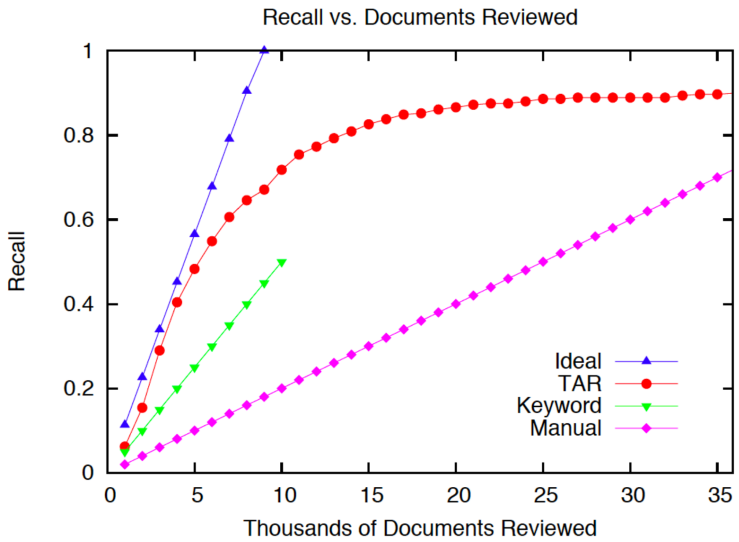
TREC, 2015

Background

Objective

- Implement automatic or semi-automatic methods to identify as many relevant documents as possible from document collections.
- Meanwhile, require as less review effort as possible. Review effort means relevance feedback from assessors.

Recall vs. Review Effort



Adam Roegiest, Charles L. A. Clarke, Gordon V. Cormack Maura R. Grossman. Total Recall Track Overview TREC 2015

Baseline

Methodology

- Cormack, Gordon V., and Maura R. Grossman. "Multi-Faceted Recall of Continuous Active Learning for Technology-Assisted Review.", SIGIR 2015.
  1. "Seed set" is constructed from the query terms.
  2. Logistic Regression classification.
  3. Select the highest-scoring documents for review.
  4. Repeat the above process until collecting a sufficient number of relevant documents.
- SAL: Simple active learning
- SPL: Simple passive learning
- Comparison: Auto-TAR $>$ SAL $>$ SPL

Potential Directions

Seed Selection

"Seed Set" can determine the trend of classification. Stronger seed set could accelerate the retrieval process.

Feature Engineering

Unigram TF-IDF based feature cannot represent the exact meaning of some phases. etc, "Deutsche Mark"

Classifier

Logistic Regression seems easy to beat.

Query Expansion

The flow of relevant documents provide informative terms to expand original query.

Seed Selection

Clustering-Based Seed Selection

1. Select Top K documents with the highest BM25 score.
2. Latent semantic indexing and dimension reduction via SVD.
3. K-Means clustering on the set of selected documents.

Sampling Strategy

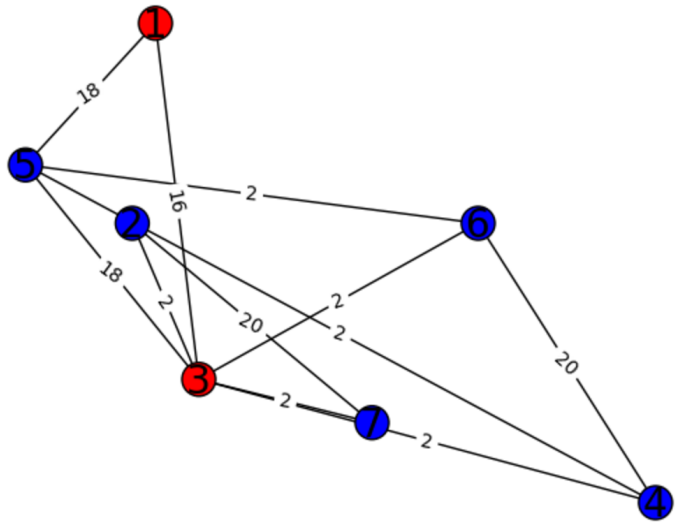
Exploration vs. Exploitation

$$l_t \in \underset{v \in 1,\ldots,k}{\operatorname{argmax}} \left\{ \frac{r_v}{t_v} + \sqrt{\frac{\mu \log(\sum_{c=1}^{|C|} t_c)}{t_v}} \right\}$$

Seed Selection

Graph Strategy

1. Documents are considered as nodes in the graph.
2. We run K-means T times to cluster these documents.
3. The weight $w_{i,j}$ of a undirected edge between node i and node j is $w_{i,j} = \sum_{t=1}^{T} \mathsf{I}_t(i,j)$.
4. Traverse the priority queue created based on the weights between documents.



Seed Selection

Jumping Strategy

- Greedy search in one cluster and switch to other cluster when not relevant document is found.

Weighted Strategy

- Assign weight for each cluster and decay the weight when encountering not relevant document.

Table: Number of relevant documents found in 50 seeds

Methods	tr0	tr1	tr2	tr3	tr4	tr5	tr6
Jumping	46	1	2	10	47	49	40
Weighted	46	0	2	10	47	49	42
Sampling	45	1	2	14	**48**	49	**46**
Graph	**47**	**2**	2	**15**	45	**50**	45

Feature Engineering

n-gram Model

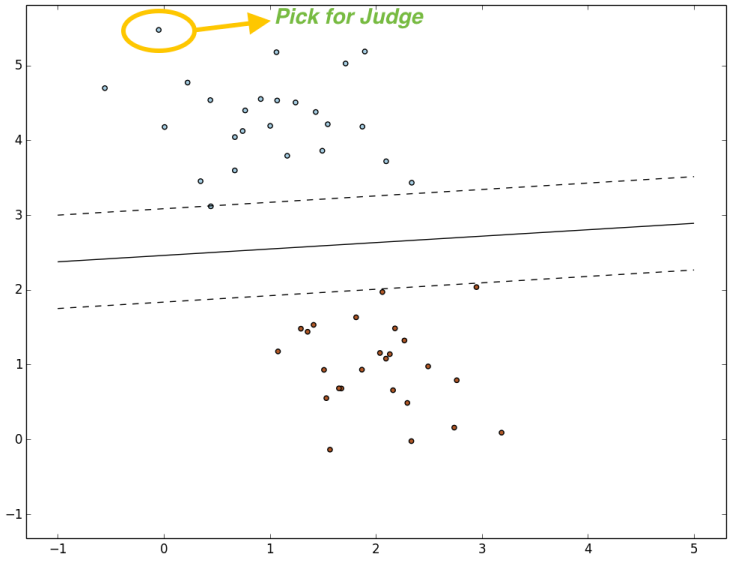
$$
\begin{aligned}
\#Rel : 1\{ & \\
& Deutsch : Weight_1 \\
& Mark : Weight_2 \\
& Deutsch\ \ Mark : Weight_3 \\
& \}
\end{aligned}
$$

- The dependency relationship between terms cannot be represented by unigram model.
- TF-IDF value of unigram, 2-gram, 3-gram. And the combination of these features.
- Other features, the entropy weighting LSI:

$$g_i = 1 + \sum_j \frac{p_{ij} \log p_{ij}}{\log n}, \text{where } p_{ij} = \frac{tf_{ij}}{gf_i} \tag{1}$$

Classifier Selection

Logistic Regression Model



- The document farthest from the decision boundary is selected for judging.
- LR and other linear model is well enough for sparse high-dimensional feature such as TF-IDF.

Classifier Selection

Classifier Comparison

Classifier	Toolbox	Feature
Logistic Regression	Sofia-ML	Unigram TF-IDF
Logistic Regression	Sofia-ML	N-gram TF-IDF
Logistic Regression	Sofia-ML	4-char TF-IDF
Linear SVM	LIBSVM	Unigram TF-IDF
Linear SVM & LR fusion	Sofia-ML	4-gram TF-IDF
RBF SVM	LIBSVM	Entropy
RBF SVM	LIBSVM	Unigram TF-IDF
Decision Tree	Scikit-Learn	Unigram TF-IDF
Naive Bayes	Scikit-Learn	Unigram TF-IDF
AdaBoost	Scikit-Learn	Unigram TF-IDF
Gradient Boosting	XGboost	Unigram TF-IDF

Table: Classifiers Applied

Classifier Selection

Cross Validation

- Though performing 5-fold cross-validation, Gaussian(RBF) kernel SVM tends to overfit in training set. Grid search for soft margin parameters: C and γ.

Other Linear Model

- Linear SVM and Linear regression performs nearly the same as LR. Linear models work with $d(dimensionality) \gg n(documents)$.
- The RRF fusion of ranking lists generated from 5 different LR classifiers can slightly improve the accuracy of classification.

Query Expansion

Simple Mixture Model - Obtain Informative Terms

Zhai, Chengxiang, and John Lafferty. "Model-based feedback in the language modeling approach to information retrieval." CIKM, 2001.

SM assumes that terms in relevant documents are generated as below:

1. Given two models θ_0 and θ_1;
2. Given a mixing coefficient, $\overrightarrow{\pi} = (1 - \pi, \pi)$;
3. For the j-th term in the i-th relevant document:
   - Firstly, independently generate a latent model indicator, $z_{ji} \sim \text{Bernoulli}(z|\overrightarrow{\pi})$;
   - Then, independently generate a term, $w_{ji} \sim d(w|\theta_{z_{ji}})$;

Query Expansion

Simple Mixture Model

The background model indicates the noise when generating a document:

$$d(w|\theta_1) = 0.5 \times d(w|\theta_{\text{corpus}}) + 0.5 \times d(w|\theta_{\text{non-rel}}) \tag{2}$$

The probabilistic model $p(F|\theta)$ generates each word in F independently according to θ is:

$$d(F|\theta) = \prod_i \prod_w d(w|\theta)^{c(w;d_i)} \tag{3}$$

Use simple mixture model, the log-likelihood of feedback documents is:

$$\log d(F|\theta_0) = \sum_i \sum_w c(w;d_i) \log((1-\pi)d(w|\theta_0) + \pi d(w|\theta_1)) \tag{4}$$

Submission

At-Home

- We ran our own system and accessed the automated assessor via the Internet. Two runs were successfully submitted: UWPAH1(without query expansion) and UWPAH2(with query expansion).

Sandbox

- We also submitted one fully automated solution (without query expansion), which the track coordinators executed as a virtual machine within a restricted environment.

Submission

Seed Selection

- Graph Strategy

Feature Engineering

- Unigram & 2-gram TF-IDF value

Classifier

- Logistic Regression

Query Expansion

- Top k terms in relevance model for each iteration

Result

Evaluation Methods

- Effort at 75%, 80% recall
- Gain curve
- "Recall@aR+b" values defined as the "Recall" that is achieved when "Effort" is equal to $aR + b$, where a and b are constant number

Table: Average review effort for each run at 75% recall

Run	Corpus	BMI	**UW**
UWPAH1	Athome1	3862	**3716**
UWPAH1	Athome2	2258	**2013**
UWPAH1	Athome3	**777**	1070
UWPAH1	Mimic	**8948**	9196
UWPAH1	Kaine	74761	**71816**
UWPAH2	Athome1	3862	**3682**

There is no statistically significant difference between our method and BMI.

Result

Table: Review effort at 75% recall in Athome1 for UWPAH1 and UWPAH2

Topic	UWPAH1	UWPAH2
athome100	4019	**3968**
athome101	4503	**4491**
athome102	**1402**	1417
athome103	4307	**4305**
athome104	**272**	291
athome105	**2898**	2981
athome106	**12861**	12892
athome107	1914	**1892**
athome108	2337	**2228**
athome109	2642	**2358**

There is no statistically significant difference between our method and BMI.

Conclusion

Conclusion

- Logistic Regression is super efficient for high dimensional sparse data.
- Feature engineering matters.
- Baseline is hard to beat.