Report for 11791 HW3

Name: Chao-Hung Chen
Andrew ID: chaohunc

Engineering and Error Analysis with UIMA

Error analysis

Based on my observation, there are five types of errors occurred in the causing the MRR decrease. 1. Special characters matching errors 2. Lower case and upper case matching errors 3. Stemming Errors 4. Question type Errors 5. Stopwords Errors

1. Special characters matching errors

By observing the training documents at first, we could easily find that query term and document term could not match because of special characters. For example, for the last term in some queries, there is a question mark in the end of term, which won't match the same term without question mark in the document. Like query 20, there is term "producer?" , which could not match with "producer" in documents. Therefore, it's important to deal with those special characters. Here is the list of query I find existing special characters matching errors

QID	Special characters matching errors	Should be revised to
2	Jordan--	Jordan
5	Sorrow?	sorrow
9	Moon.	Moon
11	Devil's	Devils
14	Commodores.	Commodores
17	producer?	producer
20	State?	State

2. Lower case and upper case matching errors

Since we didn't make all terms in same case, for some terms which happen as the first term of the sentence will not match with the query because it is capitalized. For example, in query 17, "Which U.S. state is the leading corn producer?". The term "corn" in query could not match with the term "Corn" in the relevant document because of case.

QID	lower case and upper error matching errors	Should be revised to
5	Sorrow	sorrow
9	Moon	moon
15	Earth	earth
17	Corn	corn

3. Stemming errors

To do linguistic normalization, stemming is important technique to match the variant forms of a word by reducing it to a common form. For example, in query 3, "Alaska was purchased from Russia in year 1867.", we could find that "purchased" is a term in passive voice. But we should not only match it with the document in passive voice. As long as the term is correlated with the query, present tense "purchase" or "purchases" is also acceptable. So we should stem the "purchased".

QID	Stemming matching errors	Should be revised to
3	purchased	purchase
5	China's	China
8	bite	bit
11	Devils	Devil
16	died	die
18	McDonalds	McDonald

4. Question type errors

For some queries, it is easy for human to find that what kinds of question the query want to ask by using “Who”,”When”, “What year”, etc. By finding those key terms, we could know the answer should be related to the type of questions. For instance, in query 6, “Who was the first person to run the mile in less than four minutes”, the relevant document should definitely need to contain a person name in their answers , like Roger Bannister.

QID	matching errors	Relevant document should contain
1	Name of Volcano	should contain a name of volcano
4	What year	should contain a time or date
6	Who	shoulg contain a person name
7	What year	should contain a time or date
11	Where	should contain a name of place
12	What	x (ambiguous, difficult for revising)
13	How deep	should contain a measurment
14	Who	shoulg contain a person name
16	When	should contain a time or date
18	Where	should contain a name of place
19	What	x (ambiguous, difficult for revising)

5. Stopwords Errors

If the query contains a lot of stopwords, it will also influence the query result. For instance, in query 20 “What is the Keystone State?”, we could find that “what” , “is”, “the” are all frequent words, and it’s useless for keeping those for term matching. If a irrelevant document contains a lot of “what” , “is”, “the”, it will be considered as relevant document. Therefore, we should use stop word list to remove those frequent words. Since every query has lots of stop words, I didn’t particularly list them.

4 Types of errors classification

Query Num/ Error type	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Sum
1		1			1				1		1			1			1			1	7
2					1				1						1		1				4
3			1		1			1			1					1		1			6
4	1			1		1	1				1	1	1	1		1		1	1		11
tot	**1**	**1**	**1**	**1**	**3**	**1**	**1**	**1**	**2**	**0**	**3**	**1**	**1**	**2**	**1**	**2**	**2**	**2**	**1**	**1**	**28**

1 = Special characters matching errors
2 = Lower case and upper case matching errors
3 = Stemming errors
4 = Question type errors

The table is the summary of 4 different type errors. I didn't list stopwords errors on the list since every query contains stopwords.

Experiments

1. Better tokenization algorithms

Tokenization algorithms will increase the chance for term matching, and there are some ways to implement tokenization algorithm and improve performances. In this part, I did experiments includes two kinds of tokenization algorithm, special characters removal and lower cases, stop words removal.

a. special characters removal and lower cases

b. stopwords removal

2. Better stemming algorithm

Stemming algorithm will be helpful to reduce the term into a common form. At here, I used Porter stemming algorithm, which is written by Martin Porter and was published in the July 1980 issue of the journal *Program*.

	Baseline	1.a. Special characters removal and lower cases	1.b. Stopwords removal	2. Stemming	1.a + 1.b + 2.
MRR	0.4375	0.4583	0.4917	0.4708	0.5167
one-tail p-value of t-test (Compared to baseline)	x	0.3617	0.20102	0.2824	0.12850
one-tail t of t-test (Compared to baseline)	x	0.3566	0.8475	0.5807	1.1508
Statistical Significance (95%)	x	Not significance	Not significance	Not significance	Not significance

Discussion

Although in several experiment, we couldn't find any statistical significance, we could still find that MRR increases and p-value decreases when we use better tokenization and stemming algorithms. Therefore, I tried to use different similarity functions to improve results.

3. Similarity functions

TF-IDF model

tf-idf model uses not only term frequency to build term vectors, but also uses document frequency. The key concept of idf is that if a term

appears less in each document, it will be considered as a rare term, which should have more significance than those frequent term.

$$\operatorname{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$ (equations cited from wiki)

$$\operatorname{tfidf}(t, d, D) = \operatorname{tf}(t, d) \times \operatorname{idf}(t, D)$$ (equations cited from wiki)

$$\operatorname{score}(D, Q) = \sum_{i=1}^{n} \operatorname{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})},$$

(equation citied from wiki)

BM25

Different form tf-idf models, BM 25 uses probabilistic models, instead of vector space models, to calculate similarity scores. The key idea is very similar. It also had tf and idf parts, and included document length normalizations. There are two parameters need to be determined, k1, b. Based on wikipedia, I used b = 0.75, and k1 = 1.6 to optimize the result.

Experiment result

	Baseline	**TF-IDF**	**TF-IDF (with 1.a + 1.b + 2)**	**BM25 (b = 0.75, k =1.6)**	**BM25 (b = 0.75, k =1.6) (with 1.a + 1.b + 2.)**
MRR	0.4375	0.4583	0.5167	0.4583	0.5083
one-tail p-value of t-test (Compared to baseline)	x	0.3617	0.12850	0.3617	0.1569
one-tail t of t-test (Compared to baseline)	x	0.3566	1.1508	0.3566	1.0207

	Baseline	TF-IDF	TF-IDF (with 1.a + 1.b + 2)	BM25 (b = 0.75, k =1.6)	BM25 (b = 0.75, k =1.6) (with 1.a + 1.b + 2.)
Statistical Significance (95%)	x	Not significance	Not significance	Not significance	Not significance

Discussion

Although I implement tf-idf and BM 25 models in several experiment, the performance of these two models are not very good. I couldn't find any statistical significance. I think one of the reasons is that the number of documents is not enough for training a very good model for prediction. For future work, I might include more data to improve MRR.