Working Note – FIRE 2013

Improving Accuracy of SMS based FAQ retrieval

Anwar Shaikh
(Delhi Technological University, Delhi)

Rajiv Ratn Shah
(National University of Singapore, Singapore)

Rahis Shaikh
(Vellore Institute of Technology, Vellore)

Following are the modifications and new enhancements made by our team to improve the accuracy of SMS based FAQ retrieval system.

- **Join Consecutive Terms** – Concatenation of consecutive terms is performed at two stages.

1. During preprocessing of the SMS text , Join two consecutive SMS words and perform matching with the single FAQ word. Also, match two consecutive FAQ words with single SMS word and measure the similarity.

   For each SMS term Si {
   If [! termExists(S_i) || ! termExists(S_{i+1})] && [termExists(concat(S_i , S_{i+1}))]
   S_i = concat(S_i , S_{i+1});
   }

   Eg.
   SMS : Wat r symptms of smll pox ?
   After Joining SMS tokens : Wat r symptms of smllpox ?

2. During the process of matching a SMS with FAQ –
   Eg.
   SMS-1 : Hw to get *home loan* frm bank ?
   FAQ-1: How to obtain *homeloan* from any bank ?

   In above example, during preprocessing the words home and loan are not joined together because both terms home and loan are valid (these terms exists in the domain dictionary). Calculation of the weight is shown below –

   $\omega_i = \alpha(S_i + S_{i+1} , F_i) * idf(F_i)$;
   $\omega_i = \alpha($ home + loan , homeloan$) * idf($homeloan$)$;

SMS-2 : Hw to get *homeloan* frm bank ?
FAQ-2: How to obtain *home loan* from any bank?
In SMS-2, the SMS term *homeloan* is formed by concatenation of two FAQ terms *home loan*, so it's necessary to join two FAQ terms and match it with single SMS term. Calculation of weight is performed as per the formula mentioned below -

$\omega_i = \alpha(S_i, F_i + F_{i+1}) * [idf(F_i) + idf(F_{i+1})] / 2;$
$\omega_i = \alpha(homeloan , home + loan) * [idf(home) + idf(loan)] / 2;$

Note that the average of the IDF of the i^{th} FAQ term and $(i+1)^{th}$ is taken to calculate the weight.

// Joining words while performing similarity calculation.
For each SMS token S_i {
For each FAQ token F_i {
$\alpha_1 = \alpha(S_i + S_{i+1} , F_i);$ $idf_1 = idf(F_i);$
$\alpha_2 = \alpha(S_i, F_i + F_{i+1});$ $idf_2 = [idf(F_i) + idf(F_{i+1})] / 2;$

if ($\alpha_1 > \alpha_2$){ $\omega_i = \alpha_1 * idf_1$ }
else { $\omega_i = \alpha_2 * idf_2$ }
}
}

So, while performing match between SMS token and FAQ token, we consider these words together to find the best possible match.

- **Stemming** – Porter Stemmer is used. FAQ words and SMS words are matched after stemming. Maximum similarity calculated w.r.t. Stem FAQ token and Non-stem FAQ token is considered i.e. maximum [similarity(Stem FAQ token, SMS token) , similarity(FAQ token, SMS token)].

- **Score Calculation** –
Score = W_1 * Similarity_Score + W_2 * Length_Score + W_3 * Proximity Score.

Similarity Score :

For each term t in the dictionary and each token si in the SMS query, similarity measure α(t, si) measures how closely term t matches the SMS token si. Term t is a variant of si, if α(t, si) > 0. Combining the similarity measure and the inverse document frequency (idf) of t in the corpus forms the weight function ω (t, si).

$$\text{Similarity\_Score}(Q) = \sum_{i=1}^{n} \max_{t \in Q \text{ and } t \sim s_i} \omega(t, s_i)$$

$$\omega(t, s_i) = \max(\alpha\,(t, si) * \text{idf}(t), \alpha\,(t\_stem, si) * \text{idf}(t)\ )$$

Metaphone : Algorithm is used for the calculation of the similarity.

Length Score : Revised from our previous approach –

$$\text{Length Score} = \frac{2 * \text{MatchedTokens}}{\text{totalSmsTokens} + \text{totalFaqTokens} - 2 * \text{skippedFAQtokens}}$$

Where *skippedFAQtokens* are the words skipped during the calculation of similarity score. These words are different from the Stop words.

E.g. what, when, where, which, while, who, whom, why, will, with, would, yet, you, your…

Whereas some of the Stop words are - a, an, and, are, as, at, be, but, by, for, of, on, or…

Proximity Score : The proximity score is calculated as:

$$\text{Proximity\_Score} = \frac{\text{matchedToken}}{((\text{distance} + 1) * \text{totalFaqTokens})}$$

Where,

totalFaqTokens= number of tokens in FAQ ,
matchedToken = number of matched token of SMS in FAQ.

$$\text{distance} = \sum_{k=0}^{n} \text{absolute difference between adjacent token pairs in SMS and corresponding pair in FAQ}$$

Where,

n = number of matched adjacent pairs in SMS

Results for English Mono task –

***** FIRE 2013 SMS TASK EVALUATION REPORT *****
No. of In-domain Queries :392
No. of Out of Domain Queries:148
In Domain correct:189/392 (0.48214287)
Out of Domain correct:125/148 (0.8445946)

Total Score: 0.58148146
Mean Reciprocal Rank (MRR): 0.79365075

- **Improvements for Hindi Language –**

1. **Normalization of FAQ and SMS**: some of the Hindi characters are replaced with the similar characters to make them similar. Because in many cases these characters does not change the meaning of the word.

```
replace(़, ) Makes फ़ायदा & फायदा equal
replace(ै,े)
replace(ौ,ो)
replace(ि,ी)
replace(ु,ू)
replace(्, )
replace(ई,इ)
replace(ष,श)
replace(न,ण)
replace(उ,ऊ)
replace(ॉ,ा)
```

2. **Substitution of characters**: To handle the noise in the SMS text, substitutions of few characters are made.

Substitutions	
प	फ
त	थ
श	स
क्ष	स
ष	स
ज	झ
ब	भ
क	ख
द	ध
ड	ढ
ग	घ
न	ण
उ	ऊ
र	ऋ
ई	इ
ङ	ञ
च	छ

3. Hindi Words Stemming –

Hindi light stemmer is used to stem hindi terms, the stemmer removes number, gender and case suffixes from nouns and adjectives (www.unine.ch/info/clef/)

Eg. आवश्यकताएँ after stemming becomes आवश्यकता .

4. Hindi Wordnet is used to find synonyms.

5. Score Calculation: Similar to the English language.

6. Results for Hindi Mono task:

***** FIRE 2013 SMS TASK EVALUATION REPORT *****
No. of In-domain Queries :46 No. of Out of Domain Queries:45
In Domain correct:34/46 (0.73913044) Out of Domain correct:45/45 (1.0)
Total Score: 0.8681319
Mean Reciprocal Rank (MRR): 0.9714286