Honours Thesis

Generating Clinical Queries from Patient Narratives

Liam Cripwell - 08857181
liam.cripwell@connect.qut.edu.au

Supervisor - Guido Zuccon
g.zuccon@qut.edu.au

October 9, 2017

Contents

1 Data Collection

1.1 Patient Narratives

1.2 Document Corpi

1.3 Evaluation Pipeline

2 Query Reduction

Balasubramanian et al. [?] provide a formal definition of the query reduction task to find the set P^*:

$$P^* = \arg\max_{P \in \mathcal{P}^Q} T_f(P), \tag{1}$$

where Q is the original query, $\mathcal{P}^Q$ is the set of possible subqueries of Q, and T_f is the measure of retrieval effectiveness for a given query. However, because it is not viable to calculate T_f for every P, estimations must be made; this gives us the revised definition of P^*:

$$P^* = \arg\max_{P \in \mathcal{P}^Q} \widehat{T_f}(P) \tag{2}$$

The query reduction task thus becomes the problem of finding an effective method of estimating values of T_f in order to produce the best possible value of P^*.

2.1 Health-Terms Model

This query reduction model utilises a Wikipedia dump in order to calculate term weights. Each Wikipedia page has been classified as either *health-related* or *non-health-related* depending on whether or not their *infobox* contains one or more UMLS medical concepts, respectively. TODO: investigate what methods jimmy used to identify medical concepts Query terms are then scored with the following formula:

$$OR(c_l) = \frac{Pr\{P \text{ is health-related } | c_l \in P\}}{Pr\{P \text{ is not health-related } | c_l \in P\}} \tag{3}$$

A term is retained as part of the generated reduction if $OR(c_l) \geq \delta$, where δ is a tuning parameter. This method was tested with δ set to various different values in order to identify an optimal setting.

2.2 Key-Concept Model

2.3 Concept-Based Retrieval Model

3 Query Expansion

Various PRF models have been investigated, namely; Kullback-Leibler Divergence, Rocchio.

3.1 Pseudo-Relevance Feedback

3.1.1 Kullback-Leibler Divergence PRF Model

The Kullback-Leibler formula used to score expansion term candidates is as follows:

$$p(t|R) \cdot \log \frac{p(t|R)}{p(t|C)}, \tag{4}$$

where $p(t|R)$ and $p(t|C)$ are the probability of term t occurring in the set of pseudo-relevant documents R and entire document collection C, respectively.

3.1.2 Rocchio PRF Model

TODO: rephrase this as the rocchio model and remove medical components. Soldaini et al. [**?**] adopted a variation of PRF in their investigation of CDS search tasks, whereby instead of altering the term weights, they determined boosting coefficients for each term. This difference is fairly trivial, but was done in order to fit their specific experimental setup. They expand the query by building the root set $\mathcal{R}_Q$, which consists of the union of the set containing all the terms in the query Q with the set of all the terms in the top-k documents returned for Q. The boost coefficient b_j is then calculated for $t_j \in \mathcal{R}_Q$ as:

$$b_j = \log_{10}(10 + w_j)$$

where

$$w_j = \alpha \cdot I_Q(t_j) \cdot tf_j + \beta/k \sum_{i=1}^{k} I_{D_i}(t_j) \cdot idf_j \tag{5}$$

and where $I_Q(t_j)$ is an indicator of the presence of term t_j in Q; $I_{D_i}(t_j)$ is an indicator of the presence of term t_j in the document D_i; idf_j is the inverse document frequency of the j-th term in the top k documents; and α and β are smoothing factors. Once all weights have been calculated, the terms in $\mathcal{R}_Q$ are ranked by their boost coefficient and the top m terms are added to the query. The following parameter values were used: $\alpha = 2$, $\beta = 0.75$, $k = 10$, $m = 20$.

3.1.3 Health-Terms PRF Model

3.1.4 Health-Terms+Rocchio PRF Model

This model follows the *HT+PRF* model described by Soldaini [].

3.2 Method

We experimented with the performance of these methods within the context of the clinical trials document collection. PRF functionality was applied to the existing UMLS reduction model in order to identify additional terms that may be relavent given the retrieval results of the UMLS reduced queries. Experiments were run with the number of pseudo-documents k set to 3, 5, and 7, and for each of these the number of pseudo-relevant expansion terms j was set to values between 1 and 15, in increments of 2.