An Approach for Query-focused Text Summarisation for Evidence Based Medicine

Abeed Sarker[1], Diego Mollá[1], and Cécile Paris[2]

[1] Centre for Language Technology
Department of Computing, Macquarie University
Sydney, NSW 2109, Australia
{abeed.sarker,diego.molla-aliod}@mq.edu.au
URL: http://www.clt.mq.edu.au
[2] CSIRO – ICT Centre,
Locked Bag 17, North Ryde, Sydney, NSW 1670, Australia
cecile.paris@csiro.au
URL: http://www.csiro.au

Abstract. We present an approach for extractive, query-focused, single-document summarisation of medical text. Our approach utilises a combination of target-sentence-specific and target-sentence-independent statistics derived from a corpus specialised for summarisation in the medical domain. We incorporate domain knowledge via the application of multiple domain-specific features, and we customise the answer extraction process for different question types. The use of carefully selected domain-specific features enables our summariser to generate content-rich extractive summaries, and an automatic evaluation of our system reveals that it outperforms other baseline and benchmark summarisation systems with a percentile rank of 96.8%.

Keywords: Automatic Text Summarisation, Medical Natural Language Processing, Evidence Based Medicine, Query-focused Summarisation

1 Introduction

Evidence Based Medicine (EBM) is a practice that requires practitioners to incorporate the best evidence from published research, when answering clinical queries. Due to the plethora of electronically available medical publications (e.g., PubMed[3] indexes over 22 million articles), practitioners generally face the problem of information overload. Research has shown that practitioners often fail to comply with EBM guidelines because of time constraints, particularly at point-of-care [7]. As such, there is a strong motivation in this domain for systems that can summarise text, according to the information needs expressed by practitioners. In this paper, we present an extractive, query-focused, single document summarisation system that relies on statistics generated from a specialised corpus. In particular, our system incorporates novel, domain-specific statistical

[3] http://www.ncbi.nlm.nih.gov/pubmed/ (Accessed on: 5th March, 2013)

features involving query types, medical semantic types, and associations between semantic types. We show that our approach outperforms other summarisation systems, with a percentile rank of 96.8%, in this challenging domain.

2 Related Work

While automatic text summarisation research has made significant progress in various domains (e.g., news), the medical domain still lacks complete end-to-end summarisation systems. This domain is particularly challenging because of a number of reasons including the complex nature of medical text and the large volume of domain-specific terminologies, concepts, and relationships that must be taken into account [1]. Some of the work on summarisation for this domain has been carried out under the broader research area of Question Answering (QA). [10] present a QA system whose summarisation component relies on the classification of information present in medical abstracts into PICO (**P**opulation, **I**ntervention, **C**omparison and **O**utcome) elements [14]. Text segments classified as *Outcome* are presented as the final summary. [13] perform polarity identification of medical sentences and show that summarisation can be improved with the use of this information. More recently, [2] proposed the AskHermes[4] system that performs multi-document summarisation via key-word identification and clustering of information. Our recent pilot study on query-focused summaristion [15] revealed that the content of extracted summaries can be improved via the use of target-sentence-specific statistics and specialised corpora.

In terms of automatic evaluation of summarisation systems, the most popular tool is perhaps ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [9], which provides several evaluation metrics that have been shown to have strong correlation with human judgements. Recently, [4] have shown that ROUGE scores for extractive summaries within a domain follow a normal distribution with most combinations of sentences giving a ROUGE score that is very close to the mean. The relative performance of a system can be measured by computing its percentile rank from the score distribution.

3 Data and Methods

We used a publicly available corpus that is specialised for the task of summarisation for EBM [11]. The corpus is collected from the Journal of Family Practice[5]. The corpus consists of a set of records, $R = \{r_1 \dots r_m\}$. Each record, r_i, contains one clinical query, q_i, so that we have a set of questions $Q = \{q_1 \dots q_m\}$. Each r_i has associated with it a set of one or more bottom-line answers to the query, $A_i = \{a_{i1} \dots a_{in}\}$. For each bottom-line answer of r_i, a_{ij}, there exists a set of detailed justifications (single-document summaries) $L_{ij} = \{l_{ij1} \dots l_{ijo}\}$.

[4] http://www.askhermes.org

[5] http://www.jfponline.com

Each detailed justification l_{ijk} is in turn associated with at least one source document d_{ijk}. Thus, our corpus has a set of source documents, which we denote as $D = \{d_{ij1} \dots d_{ijo}\}$, each d_i consisting of a set of n sentences $S_i = \{s_{i0} \dots s_{in}\}$ and a title t_i. In the research work described in this paper, we use the set of questions from our corpus (Q), the set of human authored summaries (L), and the set of referenced documents (D); and divide the corpus into two sets: training (R_{TRAIN}: 1388 documents) and evaluation (R_{EVAL}: 1319 documents).

3.1 Generation of Ideal Summaries

Our intent is to generate a query-focused summary from each d_i by extracting three sentences from it which most closely resemble the associated human authored summary (l_i). We define the ideal extractive summary of d_i to be a set of three sentences, $S_{BEST,i} \subseteq S_i$, from d_i, that produce the highest ROUGE-L f-score when compared with l_i. We choose three as our target number of sentences in line with past research in this area [10]. To identify $S_{BEST,i}$ for d_i, we generate all possible three sentence combinations for d_i, $S_{combs,i}$, and then perform an exhaustive search to select the combination that has the best ROUGE-L f-score. We thus have a set of ideal summaries from each d_i, $S_{BEST} = \{S_{BEST,0} \dots S_{BEST,1388}\}$, and we use this set to derive much of the required statistics. The target of our summarisation task is therefore to use statistics derived from S_{BEST} to attempt to select a set of sentences, S_{sel}, from an unseen document d, such that S_{sel} has the highest scoring ROUGE-L f-score among all the three-sentence combinations in d. We select a summary containing a set of three sentences, $S_{summary,i} = \{s_{first}, s_{second}, s_{third}\}$, from document d_i, using separate statistics, wherever appropriate, for each of s_{first}, s_{second} and s_{third}. We now discuss the features for which we derive statistics.

3.2 Generation of *Question Type* Independent Statistics

We apply two broad categories of statistics for the summarisation process. The first category involves features that are independent of the *types* of the questions:

Relative Sentence Position. Given a document d_i, we want to assign three scores to each sentence in S_i. Each score assigned to a sentence is an estimate of a probability measure, and depends on which target summary sentence (tn = 1, 2, or 3) we are attempting to select. The score for a sentence with relative position j is:

$$RP_{s_{ij}} = P(s_{ij}|tn) \tag{1}$$

i.e., the probability estimate of a sentence with relative position j to be chosen as the target sentence, *tn*. We first create normalised histograms of each of the three relative sentence position distributions for S_{BEST}. Then, for a sentence in a document with a relative position j, the score assigned is equal to the normalised frequency of the *jth* bin of the histogram. Since we have a separate distribution for each target sentence position, the same sentence gets a different score based on which distribution is used for scoring (e.g., when selecting the first target summary sentence, sentences earlier in the documents get high scores, because of higher likelihood). Thus, the scoring is not biased towards a predetermined region of text, but is determined by probability distributions from seen data.

Sentence Length. We use the equation in [15] and reward longer sentences:

$$LEN_{s_{ij}} = \frac{len(s_{ij}) - avg(len(all))}{len(d_i)} \tag{2}$$

where $len(s_{ij})$ is the length of sentence s_{ij}, $avg(len(all))$ is the average sentence length over the whole training set, and $len(d_i)$ is the length of the document.

Sentence-query Similarity. We assign a score to each sentence based on its similarity with the question, since our analyses suggest that answers tend to contain similar contents as the associated questions. We use Maximal Marginal Relevance (MMR), which has been used for summarisation in the past [3]:

$$MMR_{s_{ij}} = \lambda(CosSim(s_{ij}, q_i)) - (1 - \lambda)max_{s_k \in S_s}(CosSim(s_{ij}, s_k)) \tag{3}$$

where $CosSim()$ is a *cosine similarity* function that returns a score ranging from $1 - 0$ ($1 =$ complete match; $0 =$ no match). Our cosine similarity metric represents the two sets as vectors of word and the UMLS medical semantic type *tf.idf* features. Incorporation of the medical semantic types ensures that we don't only consider word-level similarity, but also concept level similarity.

Sentence Type Statistics. We use two probabilistic measures involving the *type* of each sentence. We use the system proposed by [8] to classify all the sentences of the abstracts in our corpus into PIBOSO (population, intervention, background, other, study, outcome) elements. We generate frequency distributions of the PIBOSO elements in R_{TRAIN} and S_{BEST} and use the normalised frequency distributions for making probability estimates. The first score, which we call the Position Independent PIBOSO Score (PIPS) is computed as:

$$PIPS_{s_{it}} = \frac{P(s_t|S_{BEST})}{P(s_t|R_{TRAIN})} \tag{4}$$

where $P(s_t|S_{BEST})$ is estimated as the proportion for PIBOSO element t among the sentences in S_{BEST}, and $P(s_t|R_{TRAIN})$ is estimated as the proportion of that PIBOSO element among the sentences in R_{TRAIN}. Thus, this score is higher for sentences belonging to PIBOSO categories that have a higher proportion among the best sentences compared to all sentences. The second score is computed as follows:

$$PDPS_{s_{it}} = \frac{P(s_t|tn = x)}{P(s_t|S_{BEST})} \tag{5}$$

where $P(s_t|S_{BEST})$ is as before, and $P(s_t|tn = x)$ is the proportion for PIBOSO element t in a target sentence-specific distribution. We call this the Position Dependent PIBOSO Score (PDPS). For example, when selecting the first sentence (i.e., $tn = 1$), a sentence classified as Background is given a much higher score compared to a sentence classified as Outcome, because of the greater frequency.

There can be six possible *PIPS* scores as there are six types of sentences. We normalise these scores by dividing each score by the sum of all six scores. Similarly, for each target sentence, there can be six possible *PDPS* scores, giving a total of 18 possible scores (six for each tn). We normalise these scores in the same way. The intuitions behind these scores have been explained in [15].

3.3 Generation of *Question Type* Dependent Statistics

Our preliminary analyses suggest that the content of a summary is influenced by the type of question. For example, the content of the answer to a question that asks about the treatment of a disease is generally different from that of a question that asks about a diagnostic procedure. Our intent is to categorise the questions in our corpus into *types*, analyse the medical concepts that are prevalent in the answers to each of the question types, and devise techniques that reward sentences by taking into account the question types and the domain-specific concepts present in the sentences. We define two scores: $SEMTYPE_{s_{ij}}$ and $ASSOC_{s_{ij}}$. We classify the questions in our corpus using the same categories, training data, and approach as [17]. There are 12 possible classes, and each question can have multiple categories or none. In our corpus, 216 questions have a single category, 167 have 2 categories, 61 have 3, 9 have 4, and 3 have no categories. *Treatment and Prevention*, *Pharmacological*, *Diagnosis*, and *Management* are the four most frequent question types in the data set.

Semantic Types for Sentence Scoring. We use the categorised questions of our corpus to identify the UMLS semantic types that are important for each question type. Similar to some of our previous scoring approaches, we rely heavily on probability estimates from frequency distributions. We generate a frequency distribution of all the UMLS semantic types present in the human authored summaries (l_i) of R_{TRAIN}. Next, we generate separate frequency distributions of l_i for each question type. The two sets of distributions illustrate how the UMLS semantic types are distributed over the whole training set and for each type of question. If a semantic type st has a high frequency in the distribution for question type t, but a low frequency in the overall distribution, it indicates that st is an important semantic type for answers to all questions of type t. The score for a semantic type st is calculated as follows:

$$semtype\_score(st, t) = \frac{P(st|t)}{P(st)} \tag{6}$$

where $P(st)$ is the probability estimate of semantic type st in the complete set of questions and $P(st|t)$ is the probability estimate of st in the questions of type

t. Thus, the *semtype_score()* is large for semantic types that are more frequent for question type t than the whole training set and vice versa. When scoring the sentences of an abstract, each sentence receives a score ($ST_{s_{ij}}$) based on the set of UMLS semantic types it contains ($SEMT_{s_{ij}}$). This score is the sum of the normalised $semtype\_score()$ for the UMLS semantic types contained in that sentence, as shown below:

$$ST_{s_{ij}} = \sum_{st \in SEMT_{s_{ij}}} semtype\_score(st, t) \tag{7}$$

Semantic Associations for Sentence Scoring. The intuition behind this score is that medical terms in the questions generally have some relationships with the terms in the summary sentences. For example, if a question has a term representing a disease and the summary contains a term that acts as the cure for a disease, we can assume that there is a *is_treated_by* relationship between the disease term and the cure term. In our domain, the disease and cure terms are represented by the UMLS semantic types. The UMLS semantic network also provides associations between semantic types, and we attempt to use these associations to identify sentences in the source texts that are related to the associated questions.

To identify important associations for each type of question, we first identify: (i) important question semantic types, and (ii) important answer semantic types. (i) is identified from the questions in R_{TRAIN}, while (ii) is identified from the manual summaries (l_i) in R_{TRAIN}. We use an approach identical to the one described in the previous subsection, and remove semantic types that have relative frequencies below a given threshold (we empirically chose 0.01 as the threshold). Once both sets of semantic types are identified, we identify the important associations that exist within a question type by applying yet another frequency distribution. For each question type t, we compute a normalised frequency distribution of all the associations between the important question and answer UMLS semantic types. Given a question q_i of type t, the probability estimate of the answer to that question having an association $assoc_l$ is the relative frequency of $assoc_l$ in the association frequency distribution for t. When scoring a sentence s_{ij}, we first identify the set of all associations s_{ij} has with the question ($AS_{s_{ij}}$), find the relative frequencies of the associations, and sum the relative frequencies. We use the function $assoc\_freq(assoc, t)$, which, given an association type and a question type, computes the relative frequency of *assoc* for t. The score assigned to the sentence is the sum of the relative frequencies, normalised by dividing the value by the total number of unique semantic types present in the question and the sentence. For questions with multiple types, the association frequency distributions for all the types are combined and normalised before computing sentence scores. The following equation summarises the scoring process:

$$ASSOC_{s_{ij}} = \sum_{assoc \in AS_{s_{ij}}} \frac{assoc\_freq(assoc, t)}{|st_{q_i} \cup st_{s_{ij}}|} \tag{8}$$

where st_{q_i} and $st_{s_{ij}}$ represent the semantic types present in the question and the sentence being scored respectively.

3.4 Combining Statistics for Sentence Extraction

We use the following *Edmundsonian* [6] equation to compute the score for s_{ij}:

$$\begin{aligned} SCORE_{s_{ijt}} = {} & \alpha RP_{s_{ij}} + \beta LEN_{s_{ij}} + \gamma PIPS_{s_{it}} + \delta PDPS_{s_{it}} \\ & + \epsilon MMR_{s_{ij}} + \zeta ST_{s_{ij}} + \eta ASSOC_{s_{ij}} \end{aligned} \tag{9}$$

where $SCORE_{s_{ijt}}$ is the score for candidate sentence s_{ijt}; i represents the document number, j represents the sentence position, and t represents the question type. When extracting the first sentence, we replace the MMR score with the cosine similarity score in the equation.

To automatically find good approximations for optimal values of the weights (α, β, γ, δ, ϵ, ζ and η), and the λ parameter in MMR, we perform a grid search through all values from 0.0 to 1.0 using step sizes of 0.1. Our intent is to find a combination of weights that maximises the chances of selecting the sentences, from an abstract, that belong to S_{BEST}. Therefore, for each combination of weights, we compute the recall values for the first, second and last sentences over R_{TRAIN}. The combination producing the best combined recall is chosen. We also apply and alternative regression based approach for comparison. In this approach, separate weights are learned for each target sentence using an SVM regression algorithm [5]. For each sentence, all the above mentioned scores are derived, along with an additional score for the degree of overlap between the sentence and the human summary. Our intuition is that the higher the overlap score, the more likely is the sentence to be in the final summary. We use the *jaccard similarity* measure to compute overlap, and compute values for the weights using the overlap scores as the dependent variables.

4 Evaluation and Results

We are interested in assessing the performance of our system relative to those of other systems, on this data set. We use the ROUGE tool for this, and compare the ROUGE-L f-scores of different systems using the percentile-rank based approach proposed by [4]. Figure 1 shows the probability distribution (*pd*) obtained using this technique for all abstracts in R_{EVAL}. The *pd* shows the range of possible scores an extractive summarisation system can have given this data set. The distribution is long-tailed, meaning that the scores for most of the extracts in the summary space are clustered around the mean. This suggests that most systems are likely to produce scores that are around the mean of the *pd*. The two ends of the distribution are shown on Figure 1 via the short vertical lines. The longer vertical line shows the best score achieved by our system.

Using the *pd*, the percentile rank for a ROUGE-L f-score, *sc*, is given by the cumulative distribution funtion for *pd* evaluated at *sc*. The baselines we use

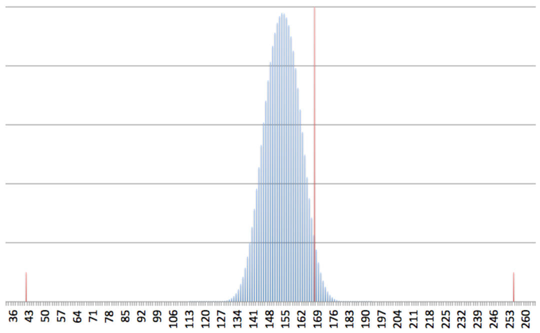

Fig. 1. The normalised histogram for all ROUGE-L f-scores in R_{EVAL}.

are: Last three sentences, last three PIBOSO *outcome* sentences (this is comparable to the summarisation component presented in [10]), random, first three sentences, all PIBOSO *outcome* sentences, SumBasic [12], FastSum (modified) [16], Sentence Position Independent (SIP), and Naïve Bayes. For the Naïve Bayes summariser, a separate classifier is trained for each target sentence using the abstracts in R_{TRAIN} with the features mentioned in the previous section. For the FastSum system, modifications were made (e.g., no redundancy removal step) to customise it to single document summarisation. The SIP system is our system without any target sentence-specific features. Table 1 presents the ROUGE-L f-scores for our system and the baselines, the 95% confidence intervals for the f-scores as reported by ROUGE, and the percentile rank for each score. In the table, QSpec represents our system, which outperforms all systems with a percentile rank of 96.8%[6]. Learning the weights via regression results in a slight degradation of performance (not statistically significant), but is still better than the other systems. The next best performing baselines are: SIP, and the *Outcome*-based systems, one of which is our implementation of the system proposed by Lin and Demner-Fushman [10]. The poor performances of SumBasic and FastSum indicate that word-frequency based approaches are perhaps not suited for this domain. Figure 2 shows a sample summary produced by the QSpec system.

To assess the contribution of each feature towards the ROUGE scores, we performed two simple experiments: (i) sentence scoring using single features only, and (ii) sentence scoring by leaving out one feature. All the single features scores indicate statistically significant improvements over the score that is obtained using no features (i.e., first three sentence summaries). Importantly, none of the single feature scores are better than the score obtained by the combination of features. The same is true for the leave-one-out scores. None of the scores are

[6] The percentile ranks for some systems in this paper are different to the ones presented in our pilot study [15]. This is because a different implementation of the same algorithm for generating the *pd* is used, which gives very slightly different values after the thousands of computations that must be performed. Relative performance comparison among the systems, however, is not different for the different *pd*s.

statistically significantly lower than the best score, which indicates that the final score is not biased by the influence of a single score. *MMR* and *ST* are shown to be the most important features from this analysis, with score drops of 0.00234 and 0.00295 units respectively in the leave-one-out experiments.

System	**F-Score**	**95% CI**	**Percentile (%)**
Last Three	0.15482	0.151 - 0.158	55.9
Last Three Outcome	0.16050	0.158 - 0.164	78.1
Random	0.15251	0.149 - 0.156	46.1
First Three	0.13994	0.136 - 0.143	36.9
All Outcomes	0.15936	0.155 - 0.164	74.2
SIP	0.16019	0.157 - 0.164	78.1
Naïve Bayes	0.15551	0.152 - 0.159	55.9
SumBasic	0.15818	0.155 - 0.162	69.9
FastSum (modified)	0.15769	0.154 - 0.161	69.9
QSpec (grid search)	0.16780	0.164 - 0.172	96.8
QSpec (regression)	0.16479	0.161 - 0.169	92.5

Table 1. ROUGE-L f-scores, 95% confidence intervals and percentile ranks for our system and several baselines.

Question: What medications are effective for treating symptoms of premenstrual syndrome (PMS)?
Summary: Forty women with premenstrual tension received either placebo, 100, 200 or 400 mg danazol daily for 3 months in a pilot study arranged as a double-blind trial. In patients treated with danazol, symptom scores for breast pain during the second and third months and for irritability, anxiety and lethargy during the third month were significantly (P less than 0.05) lower than scores in those given placebo. By the end of the trial more than 75% of patients who were still taking danazol were essentially free of breast pain, lethargy, anxiety and increased appetite, but results for other common symptoms were no better than with placebo.

Fig. 2. A sample 3-sentence, query-focused, extractive summary generated by QSpec.

5 Conclusions

In this paper, we have presented an approach for query-focused, automatic, extractive summarisation that utilises target sentence-specific statistics, question type information, and novel domain-specific features. Statistics are derived from a corpus that specialises in summarisation for EBM and the sentence extraction process relies on these derived statistics. We evaluated our system against several baselines using ROUGE and show that our system outperforms all the baselines with a percentile rank of 96.8%. This shows that use of specialised corpora, target sentence-specific statistics, and the customisation of the sentence extraction procedure to query types can significantly improve the performance of such domain specific summaristion systems. The approach is fast, as it does not apply computationally expensive NLP techniques (e.g., parsing), and can therefore be readily used post retrieval. Our future work will focus on using single-document, extractive summaries to generate multi-document, bottom-line summaries.

References

1. Athenikos, S.J., Han, H.: Biomedical question answering: A survey. Computer Methods and Programs in Biomedicine pp. 1–24 (2009)
2. Cao, Y., Liu, F., Simpson, P., Antieau, L.D., Bennett, A., Cimino, J.J., Ely, J.W., Yu, H.: AskHermes: An Online Question Answering System for Complex Clinical Questions. Journal of Biomedical Informatics 44(2), 277 – 288 (2011)
3. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of SIGIR. pp. 335–336 (1998)
4. Ceylan, H., Mihalcea, R., Özertem, U., Lloret, E., Palomar, M.: Quantifying the limits and success of extractive summarization systems across domains. In: Proceedings of NAACL. pp. 903–911 (2010)
5. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
6. Edmundson, H.P.: New methods in automatic extracting. J. ACM 16(2), 264–285 (1969)
7. Ely, J.W., Osheroff, J.A., Ebell, M.H., Bergus, G.R., Levy, B.T., Chambliss, L.M., Evans, E.R.: Analysis of questions asked by family doctors regarding patient care. BMJ 319(7206), 358–361 (1999)
8. Kim, S.N.N., Martinez, D., Cavedon, L., Yencken, L.: Automatic classification of sentences to support Evidence Based Medicine. BMC bioinformatics 12(2) (2011)
9. Lin, C.Y., Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In: Proceedings of HLT-NAACL 2003. pp. 71–78 (2003)
10. Lin, J.J., Demner-Fushman, D.: Answering clinical questions with knowledge-based and statistical techniques. Computational Linguistics 33(1), 63–103 (2007)
11. Mollá-Aliod, D., Santiago-Martinez, M.E.: Development of a Corpus for Evidence Based Medicine Summarisation. In: Proceedings of ALTW. pp. 86–94 (2011)
12. Nenkova, A., Passonneau, R.: The impact of frequency on summarization. MSR-TR, Microsoft Research, Redmond, Washington (2005)
13. Niu, Y., Zhu, X., Hirst, G.: Using outcome polarity in sentence extraction for medical question-answering. In: Proceedings of the AMIA Annual Symposium. pp. 599–603 (2006)
14. Richardson, S.W., Wilson, M.C., Nishikawa, J., Hayward, R.S.: The well-built clinical question: a key to evidence-based decisions. ACP Journal Club 123(3), A12–A13 (1995)
15. Sarker, A., Mollá, D., Paris, C.: Extractive Evidence Based Medicine Summarisation Based on Sentence-Specific Statistics. In: Proceedings of the 25th IEEE International Symposium on CBMS. pp. 1–4 (2012)
16. Schilder, F., Kondadadi, R.: Fastsum: Fast and accurate query-based multi-document summarization. In: Proceedings of ACL-HLT, Short Papers. pp. 205–208 (2008)
17. Yu, H., Cao, Y.g.: Automatically extracting information needs from ad hoc clinical questions. In: AMIA Annu Symp Proc. pp. 96–100 (2008)