Improving Medical Information Retrieval with PICO Element Detection

Florian Boudin, Lixin Shi, and Jian-Yun Nie

DIRO, Université de Montréal,
CP. 6128, succursale Centre-ville
Montréal, H3C 3J7 Quebec, Canada
{boudinfl,shilixin,nie}@iro.umontreal.ca

Abstract. Without a well formulated and structured question, it can be very difficult and time consuming for physicians to identify appropriate resources and search for the best available evidence for medical treatment in evidence-based medicine (EBM). In EBM, clinical studies and questions involve four aspects: Population/Problem (P), Intervention (I), Comparison (C) and Outcome (O), which are known as PICO elements. It is intuitively more advantageous to use these elements in Information Retrieval (IR). In this paper, we first propose an approach to automatically identify the PICO elements in documents and queries. We test several possible approaches to use the identified elements in IR. Experiments show that it is a challenging task to determine accurately PICO elements. However, even with noisy tagging results, we can still take advantage of some PICO elements, namely I and P elements, to enhance the retrieval process, and this allows us to obtain significantly better retrieval effectiveness than the state-of-the-art methods.

1 Introduction

Physicians are educated to formulate their clinical questions according to several well defined aspects in evidence-based medicine (EBM): Population/Problem (P), Intervention (I), Comparison (C) and Outcome (O), which are called PICO elements. The PICO structure is commonly used in clinical studies [7]. In many documents in medical literature, one can find the PICO structure, which is, however, often implicit and not explicitly annotated. To identify documents corresponding to a patient's state, physicians also formulate their clinical questions in PICO structure. For example, in the question "*In children with an acute febrile illness, what is the efficacy of single-medication therapy with acetaminophen or ibuprofen in reducing fever?*" one can identify the following elements: P ⇒"*children with acute febrile illness*", I ⇒"*single-medication therapy with acetaminophen*", C ⇒"*ibuprofen*" and O ⇒"*efficacy in reducing fever*".

Using a well-formulated question according to the PICO structure facilitates searching for a precise answer within a large medical citation database [10]. However, using PICO structure in Information Retrieval (IR) is not as straightforward as it seems. It requires first the identification of the PICO elements in

the documents, as well as in the question if these elements are not explicitly separated in it. Several studies have been carried out on identifying PICO elements in medical documents, and to use them in IR [5, 4]. However, these studies are limited in several aspects. First, many studies on identification of PICO elements are limited to some segments of the medical documents (e.g. Method) [4], and in most cases, the test collection is very small (a few hundreds abstracts). It is difficult to see whether one can easily identify PICO elements in all parts of medical documents in a large collection. Secondly, there have been very few tests on IR using PICO elements [5]. This is due to the lack of a standard test collection with questions in PICO structure. IR tests have been carried out on small test collections, and in many cases, not compared to the traditional IR methods. It is not clear whether IR based on PICO structure is more effective than traditional IR approaches.

In this paper, we propose an approach to perform IR using PICO elements. The identification of these elements is cast as a classification task. A mixture of knowledge-based and statistical techniques is employed to extract discriminant features that once combined in a classifier will allow us to identify clinically relevant elements in MEDLINE abstracts. Using these detected elements, we show that the information retrieval process can be improved. In particular, it turns out that the I and P elements should be enhanced in retrieval. The remainder of this paper is organized as follows. In the next section, we give an overview of the related work. Then, we present our classification approach to identify PICO elements in documents. Next, IR experiments using these elements are reported. Finally, we draw some conclusions.

2 Previous work

The first aspect of this study concerns the identification of PICO elements in medical documents. Several previous approaches have already proposed to categorize sentence types in medical abstracts using classification tools. [8] showed that Machine Learning can be applied to label structural information of sentences (i.e. Introduction, Method, Results and Conclusion). Thereafter, [5] presented a method that uses either manually crafted pattern-matching rules or a combination of basic classifiers to detect PICO elements in medical abstracts. Prior to that, biomedical concepts are labelled by Metamap [2] while relations between these concepts are extracted with SemRep [9], both tools being based on the Unified Medical Language System (UMLS). Using these methods, they obtained an accuracy of 80% for Population and Intervention, 86% for Problem and between 68% and 95% for Outcome. However, it is difficult to generalize this result, as the test was done on a very small dataset: 143 abstracts for outcome and 100 abstracts for other elements.

Recently, supervised classification was proposed by [6] to extract the number of trial participants. Results reported in this study show that the Support Vector Machine (SVM) algorithm achieves the best results with an f-measure of 86%. Again, it has to be noted that the testing data, which contains only

75 highly topic-related abstracts, is not representative of a real world task. In a later study, [4] extended this work to I and O elements using Conditional Random Fields (CRF). To overcome data sparseness, PICO structured abstracts were automatically gathered from MEDLINE to construct an annotated testing set (318 abstracts). This method showed promising results: f-measure of 83% for I and 84% for O. However, this study has been carried out in a limited context: elements are only detected within the Method section, while several other sections such as Aim, Conclusion, etc. are discarded. It is not clear whether the identification of PICO elements in the whole document can lead to the same level of performance. In this study, we do not restrict ourselves to some of the sections in documents, but try to identify elements in the whole documents.

On the retrieval aspect, there have been only a few studies trying to use PICO elements in IR and compare it to traditional methods. [5] is one of the few such studies. The method they describe consists in re-ranking an initial list of retrieved citations. To this end, the relevance of a document is scored by the use of detected PICO elements, in accordance with the principles of evidence-based medicine (i.e. quality of publication or task specificity are taken into consideration). Several other studies aimed to build a Question-Answering system for clinical questions [1]. But again, the focus has been set on the post-retrieval step, while the document retrieval step only uses a standard IR approach. In this paper, we argue that IR has much to gain by using PICO elements.

Although the retrieval effectiveness is reported in some studies using PICO elements, it is yet to be proved that a PICO-based retrieval approach will always produce better effectiveness than the traditional IR methods. In this study, we will examine the effect of using PICO elements in the retrieval process in several ways and compare them to the traditional IR models. In the next section, let us start with the first step: identifying PICO elements in medical documents.

3 Identification of PICO elements in documents

PICO elements are often implicitly described in medical documents. It is important to identify them automatically. One can use linguistic patterns for this. However, a pattern-based approach may require a large amount of manual work, and the robustness has yet to be proved on large dataset. In this study, we will rather use a more robust statistical classification approach, which requires a minimal amount of manual preparation. There may be two levels of classification: one can identify each PICO element in the document, whether it is described by a word, a phrase or a complete sentence; one can also make a coarser-grain annotation – to annotate a sentence as describing only one of the PICO elements. The second method is much simplified. Nevertheless, while the first classification is very difficult, the second one is easier to implement. Moreover, for the purpose of IR, a coarse-grain classification may be sufficient.

3.1 Construction of training and testing data

Even for a coarse-grain classification task, we are still lack of a standard test collection with PICO annotated elements. This increases the difficulty of developing and testing an automatic tool that tags these elements. This is also the reason why previous studies have focused on a small set of abstracts for testing. We notice that many recent documents in PubMed[1] do contain explicit headings such as "PATIENTS", "SAMPLE" or "OUTCOMES", etc. The sentences under the "PATIENT" and "SAMPLE" headings describe the P elements, and those under the "OUTCOMES" heading describe the O elements. Below is a segment of a document extracted from PubMed (pmid 19318702):

*... **PARTICIPANTS:** 2426 nulliparous, non-diabetic women at term, with a singleton cephalic presenting fetus and in labour with a cervical dilatation of less than 6 cm. **INTERVENTION:** Consumption of a light diet or water during labour. **MAIN OUTCOME MEASURES:** The primary outcome measure was spontaneous vaginal delivery rate. Other outcomes measured included duration of labour ...*

We collect a set of roughly 260K abstracts from PubMed by stating the limitations: *published in the last 10 years, Humans, Clinical Trial, Randomized Controlled Trial, English.* Then, structured abstracts containing distinctive sentence headings are selected and these sentences marked with the corresponding PICO elements. We notice that both Intervention and Comparison elements belong to the same semantic group and are often described under the same heading. We then choose to group the corresponding segments into the same set. From the entire collection, three sets of segments have been extracted: Population/Problem (14 279 segments), Intervention/Comparison (9 095) and Outcome (2 394). Note that abstracts can also contain sentences under other headings, which we do not include in our extraction process. Therefore, it is possible that no Outcome is extracted from a document by our process. This conservative extraction approach allows us to obtain a dataset with as little noise as possible.

3.2 Features used in classification

Prior to classification, each sentence undergoes pre-processing treatments that replace words into their canonical forms. Alpha-numeric numbers are converted to numeric numbers while each word appearance in a series of manually crafted cue-words/verbs lists is investigated. The cue-words and cue-verbs are determined manually. Some examples are shown below:

Cue-verbs: *recruit* (P), *prescribe* (I), *assess* (O)
Cue-words: *group* (P), *placebo* (I), *mortality* (O)

On top of that, three semantic type lists, generated from the MeSH[2] ontology, are used to label terms in sentences. These lists are composed with entry terms corresponding to a selection of subgroups belonging to semantic types "Living

[1] http://www.ncbi.nlm.nih.gov/pubmed/
[2] http://www.nlm.nih.gov/mesh/

Beings", "Disorders" and "Chemicals & Drugs". The final set of features we use to classify sentences is: sentence's position† (absolute, relative); sentence's length†; number of punctuation marks†; number of numbers† ($\leq$10, >10); word overlap with title†; number of cue-words⋆; number of cue-verbs⋆; MeSH semantic types⋆. Both statistical (marked with †) and knowledge-based (marked with ⋆) features are extracted. Using naive statistical features such as the number of punctuation marks is motivated by the fact that authors normally conceive their abstracts according to universally accepted rules that govern writing styles.

3.3 Identification process

Tagging each document consists in a three steps process. First, the document is segmented into plain sentences. Then each sentence is converted into a feature vector using the previously described feature set. Finally, each vector is submitted to multiple classifiers, one for each element, allowing label the corresponding sentence. We use several algorithms implemented in the Weka toolkit[3]: J48 and Random forest (decision trees), SVM (radial kernel of degree 3), multi-layer perceptron (MLP) and Naive Bayes (NB). For comparison, a position classifier (BL) was included as baseline in our experiments. This baseline method is motivated by the observation that PICO statements are typically found in specific sections of the abstract, which are usually ordered in Population/Problem, Intervention/Comparison and Outcome. Therefore, the relative position of a sentence could also reasonably predict the PICO element to which it is related. Similar method to define baseline has been used in previous studies [8].

3.4 Classification experiments

For each experiment, we report the precision, recall and f-measure of each PICO classifier. To paint a more realistic picture, 10-fold cross-validation is used for each algorithm. Moreover, all sentence headings were removed from data sets converting all abstracts into unstructured ones. This treatment allows us to take a stand on a real-world scenario by avoiding biased values for features relying on cue-words lists. The output of our classifiers is judged to be correct if the predicted sentence corresponds to the labelled one. Performance of the five classification algorithms on each data set is shown in Table 1. Not one classifier always outperforms the others but the multi-layer perceptron (MLP) achieves the best f-measure scores and SVM the best precision scores. We have performed more experiments on SVM with different kernels and settings. Best scores are obtained with a radial kernel of degree 3, other kernels giving lower scores or similar performance with higher computational costs.

As classifiers perform differently on each PICO element, in the second series of experiments, we use three strategies to combine classifier's predictions. The first method (F1) uses voting: sentences that have been labelled by the majority

[3] http://www.cs.waikato.ac.nz/ml/index.html

of classifiers are considered candidates. In case of ambiguity (i.e. multiple sentences with the same number of votes), the average of the prediction scores is used to make a decision. The second and third methods compute a linear combination of the predicted values in an equiprobable scheme (F2) and using weights empirically fixed according to the observed f-measure ranking (F3) (i.e. for the P element: 5 for MLP, 4 for RF, 3 for J48, 2 for SVM and 1 for NB). Results are also shown in Table 1. Combining multiple classifiers using F3 achieves the best results with a f-measure score of 86.3% for P, 67% for I and 56.6% for O.

	P-element			**I-element**			**O-element**		
	P	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
BL	52.1	52.1	52.1	21.9	21.9	21.9	20.0	20.0	20.0
J48	79.7	75.8	77.7	57.3	54.6	55.9	49.7	42.0	45.5
NB	66.9	65.0	66.0	50.1	47.9	49.0	48.6	47.7	48.1
RF	86.7	81.3	83.9	67.2	60.2	63.5	55.7	46.2	50.6
SVM	**94.6**	61.2	74.3	**79.6**	26.1	39.3	**75.4**	10.9	19.0
MLP	86.3	**84.5**	**85.4**	67.1	**65.6**	**66.3**	57.0	**54.5**	**55.7**
F1	**89.9**	78.2	83.6	**71.2**	55.2	62.2	**62.6**	42.7	50.8
F2	86.2	85.0	85.6	66.5	64.8	65.6	57.2	54.8	56.0
F3	86.9	**85.7**	**86.3**	67.8	**66.3**	**67.0**	57.7	**55.7**	**56.6**

Table 1. Performance of each classifier and their fusing strategies in terms of precision (P), recall (R) and f-measure (F).

One can see that O or I elements are more difficult to identify than P elements. The reason is not exclusively due to the decreasing amount of training data but mainly to the task complexity. Indeed, I elements are often missdetected because of the high number of possible candidates. Terms belonging to the semantic groups usually assigned as I (e.g. drug names) are scattered throughout the abstract. Another reason is the use of specific terms occurring in multiple PICO elements. For example, although treatments are highly related to intervention, they can also occur in other elements. In the following IR experiments, we will use the F3 (best results) tagging strategy.

4 Using PICO elements in information retrieval

The language modeling approach to Information Retrieval models the idea that a document is a good match to a query if the document model is likely to generate the query. Most language-modeling work in IR use unigram language models – also called bags-of-words models– assuming that there is no structure in queries or documents. A typical way to score a document D as relevant to a query Q is to use the *Kullback-Leibler divergence* between their respective language models:

$$score(Q, D) = -KL(M_Q \parallel M_D) \propto \sum_{t \in Q} p(t \mid M_Q) \cdot \log p(t \mid M_D) \qquad (1)$$

where $p(t \mid M_Q)$ and $p(t \mid M_D)$ are (unigram) language models of the query and document respectively. Usually, the query model is simply estimated by Maximum Likelihood Estimation over the query words, while the document model is smoothed (e.g. using *Dirichlet* smoothing) to avoid zero probabilities problem.

4.1 Model definitions

We propose several approaches that extend the basic LM approach to take into consideration the PICO element annotation. According to the PICO tagging, the content of queries and documents is divided into the following four fields: Population and Problem (P), Intervention/Comparison (I), Outcome (O), and Others (X). Let us use the following notation: $Q_{All} = Q_P + Q_I + Q_O + Q_X$ for the query Q and $D_{All} = D_P + D_I + D_O + D_X$ for the document D. In case of missing tagging information, the basic bag-of-words model is used.

4.1.1 Using PICO tags in queries

We try to assign an importance (weight) to each of the PICO elements. Intuitionally, the more important is a field, the higher should be its weight. We propose the following two models by adjusting the M_Q weighting:

Model-1T: adjusting weights on PICO element (term) level. The query model is re-defined as follows:

$$p_1(t \mid M_Q) = \gamma \cdot \frac{count(t, Q)}{\mid Q \mid} \cdot \left(1 + \sum_{E \in P,I,O} w_{Q,E} \cdot \delta(Q_E, t)\right) \tag{2}$$

where $w_{Q,E}$ is the weight of query field E; $\delta(Q_E, t) = 1$ if $t \in Q_E$, 0 otherwise; γ is a normalization factor. The score function of this model, namely $score_{1T}$, is obtained by replacing the $p(t \mid M_Q)$ by $p_1(t \mid M_Q)$ in Equation (1).

Model-1F: adjusting weights on PICO field level. Four basic models for D_{ALL}, D_P, D_I and D_O are created. The final score is their weighted linear interpolation with $w_{Q,E}$:

$$score_{1F}(Q, D) = score(Q_{All}, D) + \sum_{E \in P,I,O} w_{Q,E} \cdot score(Q_E, D) \tag{3}$$

4.1.2 Using PICO tags in documents

We assume each field in the tagged document has a different importance weight $w_{D,E}$. The document model is redefined as follows:

$$p_2(t \mid M_D) = \gamma \cdot \left(p(t \mid M_{D_{All}}) + \sum_{E \in P,I,O} w_{D,E} \cdot p(t \mid M_{D_E})\right) \tag{4}$$

where γ is a normalization factor, and $p(t \mid D_E)$ uses the Dirichlet smoothing function. We denote this model by **Model-2**, and the $score_2$ is obtained by replacing $p(t \mid M_D)$ by $p_2(t \mid M_D)$ in Equation (1).

4.1.3 Using PICO tags in both queries and documents

Model-3T: enhancement at the term level. The query model is redefined as in case 1 and document model is redefined as in case 2.

$$score_{3T}(Q, D) = \sum_{t \in Q} p_1(t \mid M_Q) \cdot \log p_2(t \mid M_D) \tag{5}$$

Model-3F: enhancement at the field level. This is the combination of **Model-2** and **Model-1F**.

$$score_{3F}(Q, D) = score_2(Q_{All}, D) + \sum_{E \in P,I,O} w_{Q,E} \cdot score_2(Q_E, D) \tag{6}$$

In all our models, there are a total of 6 weighting parameters, 3 for queries ($w_{Q,P}$, $w_{Q,I}$, $w_{Q,O}$) and 3 for documents ($w_{D,P}$, $w_{D,I}$, $w_{D,O}$).

4.2 Identifying elements in queries

PICO elements may be manually marked in queries by the user. This is, however, not a realistic situation. More likely, queries will be formulated as a free sentence or phrases. Identifying PICO elements in a query is different from what we did on documents because we need to classify smaller units. In this paper, we adopt a language model classification method [3], which is an extension to Naïve Bayes. The principle is straightforward: Let P, I and O be the classes. The score of a class c_i for a given term t is estimated by $p(t \mid c_i) \cdot p(c_i)$. The probability $p(c_i)$ can be estimated by the percentage of training examples belonging to class c_i and $p(t \mid c_i)$ by maximum likelihood with *Jelinek-Mercer* smoothing:

$$p_{JM}(t \mid c_i) = (1 - \lambda) \cdot p(t \mid c_i) + \lambda \cdot p(t \mid \mathcal{C}) \tag{7}$$

where $\mathcal{C}$ is the whole collection and λ is smoothing parameter.

The above approach requires a set of classified data in order to construct the LM of each class. To this end, we use the sentences classified by the previously described approach (see Section 3). Usually, users prefer to select important terms as their queries. As a consequence, queries should contain more PICO elements than documents. Therefore, we assume that each query term belongs to one of the P, I, or O classes. Performance of the classification method is computed over a set of 52 queries (corpus described in Section 5) by comparison to a manual tagging and experimented on different values of the parameter λ. Best results are obtained for λ set to 0.5 with an f-measure of 77.8% for P, 68,3% for I and 50% for O.

5 IR experiments

We gathered a collection of 151,646 abstracts from PubMed by searching for the keyword "*diabetes*" and stating the following limitations: *Humans* and *English language*. The average length of the documents is 276 words. The tagging

time spent by our fusing strategy (see Section 3) was approximately one hour on a standard desktop computer. For queries, we use the Cochrane systematic reviews[4] on 10 clinical questions about "*diabetes*". All the references in the "Included" studies are judged to be relevant for the question. These included studies are selected by the reviewer(s) (the author(s) of the review article) and judged to be related to the clinical question. As these studies are published prior to the review article, we only try to retrieve documents published before the review's publication date. From the selected 10 questions, medical professionals (professors in family medicine) have formulated a set of 52 queries. Each query has been manually annotated according to the following elements, which extend the PICO structure: Population (P), Problem (Pr), Intervention (I), Comparison (C), Outcome (O), and Duration (D). However, in our experiments, we will use a simplified tagging: P and Pr are grouped together (as P), C and D are discarded. Below are some of the alternative formulations of queries for the question "*Pioglitazone for type 2 diabetes mellitus*":

In patients$^{(P)}$ | *with type 2 diabetes*$^{(Pr)}$ | *does pioglitazone*$^{(I)}$ | *compared to placebo*$^{(C)}$ | *reduce stroke and myocardial infarction*$^{(O)}$ | *2 year period*$^{(D)}$

In patients$^{(P)}$ | *with type 2 diabetes who have a high risk of macrovascular events*$^{(Pr)}$ | *does pioglitazone*$^{(I)}$ | *compared to placebo*$^{(C)}$ | *reduce mortality*$^{(O)}$

The resulting testing corpus is composed of 52 queries (average length of 14.7 words) and 378 relevant documents. In our experiments, we will try to answer several questions: does the identification of PICO elements in documents and/or in queries helps in IR? and in the case of a positive answer, how should these elements be used in the retrieval process?

5.1 Baseline methods

We first tested a naïve approach that matches the tagged elements in the query with the corresponding elements in the documents, i.e. each PICO tag defines a field, and terms are allowed to match within the same field. However, this approach quickly turns out to be too restrictive. This restriction is amplified by the low accuracy of PICO tagging. Therefore, we will not consider this method as baseline but the two following instead:

Boolean model: This is the search mode widely used in medical domain. Usually, a user will construct a Boolean query iteratively by adding and modifying terms in the query. We simulate this process by creating a conjunction of all the words. Queries created in this way may be longer than what a physician would construct. Boolean retrieval resulted in a MAP of 0.0887 and a P@10 of 0.1885.

Language model: This is one of the state-of-the-art approaches in current IR research. In this method, both a document and a query are considered as bag-of-words, and no PICO structure is considered. The LM approach resulted in a MAP of 0.1163 and a P@10 of 0.25. This is the baseline we will compare to.

[4] http://www.cochrane.org/reviews/

5.2 Using document tagging

In this first series of experiments, we consider the detected PICO elements in documents while the queries are considered as bag-of-words. During the retrieval process, each element E, $E \in \{P, I, O\}$, is boosted by a corresponding weight $w_{D,E}$. We begin by setting weights to 0.1 to see the impact of boosting each element alone. Table 2 shows that when these elements are enhanced, no noticeable improvement is obtained. We then try different combinations of weighting parameters from 0 to 0.9 by steps of 0.1. The best improvement remains very small ($w_{D,P} = 0.5/w_{D,I} = 0.2/w_{D,O} = 0$) and in most cases, we get worse results.

Baseline	$w_{D,P} = 0.1$	$w_{D,I} = 0.1$	$w_{D,O} = 0.1$	**Best***
0.1163	0.1168 (0.0%)	0.1161 (−0.2%)	0.1162 (−0.1%)	0.1169 (+0.5%)

Table 2. MAP scores for Model-2 (without query tagging, $\star$: $w_{D,P} = 0.5$, $w_{D,I} = 0.2$).

The above results show that it is not useful to consider PICO elements only in documents, while using a query as bag-of-words. There may be several reasons for this. First, the accuracy of the automatic document tagging may be insufficient. Second, even if elements are correctly identified in documents, if queries are treated as bags-of-words then any PICO element can match with any identical word in the query, whether it describe the same element or not. In this sense, identifying elements only in documents is not very useful.

5.3 Using both query and document tagging

Now, we consider PICO tagging in both queries and documents. For simplicity, the same weight is used for queries and documents. In this series of tests, we use manual tagging for the queries and automatic tagging for documents. Results in Table 3 show the best figure we can obtain using this method. We can see that by properly setting the parameters, the retrieval effectiveness can be significantly improved, in particular when I elements are set to a relatively high weight, P elements to a medium one, and no enhancement to O. This seems to indicate that the I element is the most important in medical search (at least for the queries we considered). This is consistent with some previous studies on IR using PICO elements. In fact, [11] suggested firstly using I and P elements to construct Boolean queries; and only if too many results are obtained that other elements should be considered.

Measure	**Model-1T**	**Model-3T**	**Model-1F**	**Model-3F**
MAP	0.1442 (+24.0%‡)	0.1452 (+24.8%‡)	0.1514 (+30.2%‡)	0.1522 (+30.9%‡)
P@10	0.3173 (+26.9%‡)	0.3404 (+36.1%‡)	0.3538 (+42.7%‡)	0.3577 (+23.0%‡)

Table 3. Performance measures for Model-1T, Model-3T ($w_{\cdot,P} = 0.3/w_{\cdot,I} = 0.9/w_{\cdot,O} = 0$), Model-1F and Model-3F ($w_{\cdot,P} = 0.1/w_{\cdot,I} = 0.3/w_{\cdot,O} = 0$) ($\ddagger$: t.test < 0.01). Increase percentage over baseline is given in parentheses.

5.4 Determining parameters

The question now is: can we determine reasonable weights automatically? We use cross-validation in this series of exepriments to test this. We have divided the 52 tagged queries into two groups: Q26A and Q26B. A grid search (from 0 to 1 by step of 0.1) is used to find the best parameters for Q26A, and test on Q26B, and vice versa. Results are shown in Table 4. The best parameters found for Q26A in **Model-1T** are $w_{Q,P} = 0.6/w_{Q,I} = 0.9/w_{Q,O} = 0$ (MAP = 0.1688, P@10 = 0.2269), and for Q26B are $w_{Q,P} = 0/w_{Q,I} = 0.9/w_{Q,O} = 0$ (MAP = 0.1301, P@10 = 0.4192). Similar for **Model-1F**, the best parameters for Q26A are $w_{Q,P} = 0.2/w_{Q,I} = 0.3/w_{Q,O} = 0$ (MAP = 0.1784, P@10 = 0.2308), and for Q26B are $w_{Q,P} = 0/w_{Q,I} = 0.3/w_{Q,O} = 0$ (MAP = 0.1350, P@10 = 0.4808). The experiments in Table 4 show that by cross-validation, we can determine parameters that lead to a retrieval accuracy very close to the optimal settings.

Cross-validation	Measure	Baseline	Model-1T	Model-1F
Q26A→Q26B	MAP	0.1221	0.1566 (+28.2%‡)	0.1596 (+30.6%‡)
	P@10	0.1846	0.2154 (+16.7%‡)	0.2308 (+25.0%‡)
Q26B→Q26A	MAP	0.1104	0.1251 (+13.4%‡)	0.1341 (+21.5%‡)
	P@10	0.3154	0.4192 (+32.9%‡)	0.4769 (+51.2%‡)

Table 4. Performance measures in cross-validation (train→test) for Model-1T and Model-1F, queries are manually annotated.

5.5 Impact of automatic query tagging

Previous results show that query tagging leads to better IR accuracy. The question is whether this task, if performed automatically, still leads to improvements. Compared to manual annotation, automatic query tagging also works well even with low tagging accuracy (Table 5). One explanation may be that the manual tagging is not always optimal. For example, the query "*In patients with type 2 diabetes*$^{(P)}$*; pioglitazone*$^{(I)}$*; reduce cardiovascular events adverse events mortality improve health related quality life*$^{(O)}$" is automatically tagged as "*patients type 2 diabetes cardiovascular health*$^{(P)}$*; pioglitazone reduce*$^{(I)}$*; events adverse events mortality improve related quality life*$^{(O)}$". The average precision for this query is improved from 0.245 to 0.298. Intuitionally, tagging *cardiovascular* as P seems to be better than O even if it is not necessarily more correct. However, one also has to consider the utilization of it. By marking *cardiovascular* as P, this concept will be more enhanced, which in this case turns out to be more beneficial.

Measure	Baseline	Manual	Automatic
MAP	0.1163	0.1514 (+30.2%)	0.1415 (+21.7%)
P@10	0.2500	0.3538 (+41.5%)	0.3038 (+21.5%)

Table 5. Performance measures for Model-1F ($w_{Q,P} = 0.1/w_{Q,I} = 0.3$)

6 Conclusion

PICO is a well defined structure widely used in many medical documents which can also be used to formulate clinical questions. However, few systems have been developed to allow physicians to use PICO structure effectively in their search. In this paper, we have investigated the utilization of PICO elements in medical IR. We first tried to identify these elements in documents and queries, then a series of models have been tested to compare different utilizations of them.

Our experiments on the identification of PICO elements showed that the task is very challenging. Our classification accuracy is relatively low. This may lead one to think that the identification result is not useable. However, our experiments on IR showed that significant improvements using PICO elements can be achieved, despite the relatively low accuracy. This shows that we do not need a perfect identification of PICO elements before using them. IR can tolerate a noisy identification result. The key problem is the correct utilization of the tagging results. In our experiments, we have found that enhancing some PICO elements in queries (and in documents) leads to better retrieval results. This is especially true for the I and P elements.

References

1. Andrea Andrenucci. Automated Question-Answering Techniques and the Medical Domain. In *HEALTHINF*, pages 207–212, 2008.
2. A.R. Aronson. Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program. In *AMIA Symposium*, 2001.
3. J. Bai, J.Y. Nie, and F. Paradis. Using language models for text classification. In *Asia Information Retrieval Symposium (AIRS), Beijing, China*, 2004.
4. G. Chung. Sentence retrieval for abstracts of randomized controlled trials. *BMC Medical Informatics and Decision Making*, 9(1):10, 2009.
5. D. Demner-Fushman and J. Lin. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103, 2007.
6. M.J. Hansen, N.O. Rasmussen, and G. Chung. A method of extracting the number of trial participants from abstracts describing randomized controlled trials. *Journal of Telemedicine and Telecare*, 14(7):354–358, 2008.
7. W.R. Hersh. *Information retrieval: a health and biomedical perspective*. Springer Verlag, 2008.
8. L. McKnight and P. Srinivasan. Categorization of Sentence Types in Medical Abstracts. In *AMIA Symposium*, 2003.
9. T.C. Rindflesch and M. Fiszman. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of Biomedical Informatics*, 36(6):462–477, 2003.
10. C. Schardt, M. Adams, T. Owens, S. Keitz, and P. Fontelo. Utilization of the PICO framework to improve searching PubMed for clinical questions. *BMC Medical Informatics and Decision Making*, 7(1):16, 2007.
11. JM Weinfeld and K. Finkelstein. How to answer your clinical questions more efficiently. *Family practice management*, 12(7):37, 2005.