TR No. SIDL-WP-1999-0115

Approximate Query Translation (Extended Version)

Chen-Chuan K. Chang
Electrical Engineering Department

Héctor García-Molina
Computer Science Department

Stanford University
{kevin,hector}@db.stanford.edu

Abstract

In this paper we present a mechanism for approximately translating Boolean query constraints across heterogeneous information sources. Achieving the best translation is challenging because sources support different constraints for formulating queries, and often these constraints cannot be precisely translated. For instance, a query [score $>$ 8] might be "perfectly" translated as [rating $>$ 0.8] at some site, but can only be approximated as [grade = A] at another. Unlike other work, our general framework adopts a customizable "closeness" metric for the translation that combines both precision and recall. Our results show that for query translation we need to handle interdependencies among both query conjuncts as well as disjuncts. As the basis, we identify the essential requirements of a rule system for users to encode the mappings for atomic semantic units. Our algorithm then translates complex queries by rewriting them in terms of the semantic units. We show that, under practical assumptions, our algorithm generates the best approximate translations with respect to the closeness metric of choice. We also discuss how the precision and recall of the translated queries can be estimated.

1 Introduction

To enable interoperability, mediator systems [1, 2] must integrate heterogeneous information sources with different data representations and search capabilities. A mediator presents a unified context for uniform information access, and consequently must translate *original* user queries from the unified context to a *target* source for native execution. This translation problem has become more critical now that the wide range of disparate sources are just "one click away" across the Internet. Achieving the best translation is challenging because sources use different constraints for formulating queries, and often these constraints cannot be precisely translated. This paper presents a framework that finds perfect mappings if possible, or in general the "closest" approximations, taking into account differences in attribute names, operators, and data formats.

Example 1: Consider a mediator that integrate on-line shopping sites for books, audio, and videos. In particular, the mediator presents a unified view Media(name, format, $\cdots$). Suppose a user wants to find the "cassettes" by some artist "Harrison." Let us consider translating the corresponding constraints f_c = [format = cassette] and h = [name contains Harrison].

The mediator will find perfect mappings whenever possible, *e.g.*, it will translate h to [au contains Harrison] for source fatbrain.com. However, in many cases such perfect mappings simply do not exist. For instance, for source *amazon* (at amazon.com), neither f_c nor h can be translated precisely.

Consequently, some schemes focus on "minimal superset" mappings [3], which will return all the potential answers but with as few unwanted answers as possible. In particular, the mediator will map f_c to [type = audio] for *amazon*, returning cassettes as well as CDs. Unfortunately, for h the only superset mapping at *amazon* is *True* (*i.e.*, returning the whole source database), which is often unacceptable.

However, in many cases, good approximations do exist, and they may be more favorable. For instance, *amazon* can approximate h as [au = "Harrison,*"] to match Harrison as a last name. (Note that *amazon* requires at least a last name for au.) It will miss those answers with Harrison as the first name, *e.g.*, "Ford, Harrison". However, since most users will actually mean last names (e.g., "Harrison, George") in such an artist query, this mapping may be better than *True*.

In fact, even f_c may need a different approximation, say if [type = audio] returns a huge number of CDs and very few cassettes. Alternatively, mapping [desc contains "cassette"] simply looks for "cassette" in the textual descriptions. This mapping may return a lot less data than [type = audio], but may perhaps miss a few cassettes (that do not have the term "cassette" in their description). If the "false negatives" are acceptable, then the alternative mapping may be more attractive. ■

We can view a query as a Boolean expression of constraints of the form [attr1 op value] or [attr1 op attr2]. (While not explicitly discussed here, our framework can handle the latter "join" constraints [4].) These constraints constitute the query "vocabulary," and must be transformed to "native" constraints understood by the target source. For example, constraint [score > 8] may have to be mapped to [grade = A]. In this process, attribute names have to be mapped (*e.g.*, score to grade), values have to be converted (*e.g.*, score 8 corresponds to grade A), and operators have to be transformed (*e.g.*, ">" to "=").

After we first studied query translation in an earlier work [3], and implemented that machinery, we soon realized that *approximate translation* is critical for "real-world" applications. Our earlier work focused on minimal-superset mappings as the "correct" translations, because the *exact* results can be recovered by post-processing their supersets. As just illustrated, in many cases only approximations exist, and they might be even more practical than the strictly correct ones. (Analogously, a concurrent system with strict serializability may result in undesirable low concurrency.)

Furthermore, different mediation applications need different "correctness" or *closeness* criteria for mappings. It is thus essential for a translation system to flexibly support a wide range of closeness metrics. This paper presents such a framework, where the best approximate translations can be found under virtually *any* reasonable metric. In particular, the framework supports minimal-superset, maximal-subset (when extra-answers must not be returned), and other "hybrid" criteria in between. Our customizable criteria allows one to quantify "false positives" and "false negatives" that are expected to occur in a translation, in an analogous fashion to how the conventional IR parameters of precision and recall quantify "errors" that occur in executing a single query.

Our results show that, under such flexible metrics, one must cope with *interdependencies* among both query conjuncts and disjuncts. It is thus critical to note that query mapping is not simply a matter of translating each constraint separately. Some interrelated constraints can form a "semantic unit" that must be handled together. This discovery is surprising since our previous study [3] showed that query disjunctions can be translated separately, significantly simplifying the translation process. Now, in an approximate

translation scenario, interrelation depends on the particular closeness metric, as we next illustrate.

Example 2: Let us continue our audio search example. Suppose that the user is looking for both cassettes and CDs by asking the query $Q = f_c \vee f_d$, where f_d = [format = disc]. (Recall that f_c = [format = cassette].) Let us denote the closest mapping (for some closeness metric) of query Q as $\mathcal{S}(Q)$.

Suppose that the mediator adopts the minimal-superset metric, under which it will generate $\mathcal{S}(f_c)$ and $\mathcal{S}(f_d)$ both as [type = audio] (Example 1). In this case, to translate Q, the mediator can separately map the disjuncts f_c and f_d, *i.e.*, $\mathcal{S}(Q) = \mathcal{S}(f_c) \vee \mathcal{S}(f_d)$ = [type = audio], which indeed precisely translates Q, *i.e.*, $Q \equiv \mathcal{S}(Q)$.

To contrast, assume that the mediator is concerned about large result sizes, so as illustrated earlier, uses the mappings $\mathcal{S}(f_c)$ = [desc contains cassette] and $\mathcal{S}(f_d)$ = [desc contains disc]. (That is, given the mediator's closeness metric, these are the best approximate translations.) Now $\mathcal{S}(f_c) \vee \mathcal{S}(f_d)$ = [desc contains cassette] $\vee$ [desc contains disc]. This mapping is not as good as [type = audio], which in our example exactly gets all CDs and cassettes. Thus, for the closeness metric in use, translating $\mathcal{S}(f_c)$ and $\mathcal{S}(f_d)$ separately leads to a suboptimal mapping, and hence disjunction Q is not "separable." ∎

Query translation must rely on human expertise to define what constraints may be interrelated, and how to translate basic semantic units. For instance, in Example 2 we need a rule for translating the single-constraint pattern [format = F] such as f_c and f_d. But do we need a rule for composite queries, *e.g.*, $(f_c \vee f_d)$? What kind of queries must constitute such "semantic units"? In this paper we will answer these questions, identifying the essential requirements for a translation rule system.

Based on rules, our challenge is to translate arbitrary queries as Boolean expressions of constraints (we currently do not handle negation). Our approach is to *divide-and-conquer*. We present Algorithm *NFB* to "decompose" an original query into its semantic units, which can then be translated by the given rules. Note that there are many decompositions, but not all of them will lead to the closest mapping. In our running example, suppose that we are given translation rules for the semantic units (h), (f_c), (f_d), and $(f_c \vee f_d)$, and we wish to translate query $hf_c \vee hf_d$. We can decompose the query as $(h)(f_c) \vee (h)(f_d)$, or with some rewriting, as $(h)(f_c \vee f_d)$. On which expression should we apply the rules to obtain the best mapping? Is the best solution unique? How is the optimality of translation guaranteed? Again, we will answer these questions in this paper.

In summary, we make the following main contributions for approximate query translation:

- We propose a general framework, and we define the notion of translation closeness. Our framework can adopt different closeness metrics for different applications.
- We present fundamental theorems on the separability of query terms, and safeness of decompositions. These results are critical for the development of any algorithm that attempts approximate query translation.
- We present Algorithm *NFB*, which systematically finds the best translation with respect to a given closeness metric. It will find a *unique* best-mapping in the practical cases when semantic units do not "interlock."
- We study how to estimate the precision and recall parameters of a translation, and we show that reasonable formulas do exist for such estimation.

We briefly discuss related work in Section 2, and then start in Section 3 by defining a closeness criteria that combine precision and recall. Section 4 studies a basic assumption on compositional monotonicity and our results on compositional separability. In Section 5 we present our framework and Algorithm *NFB*. Finally, Section 6 concludes with simple formulas for estimating precision and recall of translated query compositions.

2 Related Work

Information integration has been an active research area [1, 2, 5, 6]; however, we believe that our focus on the constraint mapping problem is unique. Many integration systems have dealt with source capabilities, *e.g.*, Information Manifold [7, 8], TSIMMIS [9, 10], Infomaster [11, 12], Garlic [13, 14], DISCO [15], and others [16, 17, 18]. These efforts have mainly focused on generating query plans that observe the *grammar* restrictions of native queries (such as allowing conjunctions of two constraints, or disallowing disjunctions). Our work complements these efforts by addressing the semantic mapping of constraints, or analogously the translation of vocabulary. In particular, the output of our semantic mapping (which uses the vocabulary understood by the target source) can be the input to the capability mapping that others have analyzed. See reference [3] for additional details.

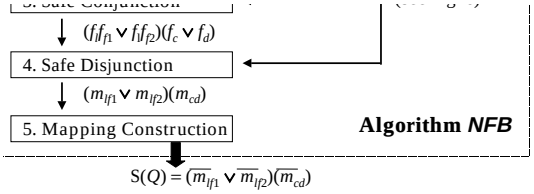
Surprisingly, while approximation is critical for query mapping (Section 1), we have seen virtually no translation efforts that stress this notion. However, approximation has been studied for query processing: First, some work aims to reduce processing cost through approximation. For instance, references [19, 20] study the approximate fixpoints of Datalog predicates, and [21] uses approximate predicates as filters for expensive ones. Second, several researchers have explored accelerated but approximated query answering to reduce response time [22, 23, 24, 25]. Finally, CoBase [26] explored query relaxation for approximate and conceptual query answering.

We define our translation metrics based on the parameters of precision and recall. Both classic notions have been commonly used for quantifying respectively false-positives and false-negatives, most notably for information retrieval [27, 28]. In addition, some single-valued measures for IR effectiveness have also been proposed, such as the well-known E-measure [27], which Section 3 will discuss.

Finally, the approximate translation discussed in this paper was motivated by our previous work [3]. As Section 1 mentioned, our earlier model of "exact" mappings significantly simplifies the translation process, but unfortunately cannot accommodate more general closeness metrics. In contrast, this paper specifically explores the notion of *approximation*, and deals with mappings under virtually *any* reasonable closeness metric.

3 Query Approximation: Accounting for Precision and Recall

Our goal for query mapping is to find the closest translation for an original query, which may not be fully expressible at the target. To quantify how closely a mapping M approximates the original query Q, we use a *closeness criterion* $\mathcal{F}[M, Q]$ that returns a normalized "rating" in $[0 : 1]$ as the *closeness* between M and Q. The higher the rating is, the closer M approximates Q. Our framework allows a wide variety of closeness functions (we will discuss some intuitive and important ones). We say that a mapping M is the *closest mapping* for Q with respect to the closeness criterion $\mathcal{F}$, if for any other mapping M' of Q,



$S(Q) = (\overline{m_{lf1}} \vee \overline{m_{lf2}})(\overline{m_{cd}})$