# Syntactic Similarity for Ranking Database Answers obtained by Anti-Instantiation

Lena Wiese

Institute of Computer Science
University of Göttingen
Goldschmidtstrasse 7
37077 Göttingen
Germany
lena.wiese@uni-goettingen.de

**Abstract.** Flexible query answering can be implemented in an intelligent database system by query generalization to obtain answers close to a user's intention although not answering his query exactly. In this paper, we focus on the generalization operator "Anti-Instantiation" and investigate how syntactic similarity measures can be used to rank generalized queries with regard to their closeness to the original query.

## 1 Introduction

Searching for data in a conventional database is a tedious task because a correct and exact formulation of the query conditions matching a user's query intention is often difficult to achieve. This is why users need the support of intelligent and flexible query answering mechanisms. *Cooperative* (or *flexible*) query answering systems internally revise failing user queries and return answers to the user that are more informative for the user than just an empty answer. In this paper, we devise a ranking based on similarity of conjunctive queries that are generated by a generalization procedure. With this ranking the database system has the option to only answer the queries most similar to the original query.

In this paper we focus on flexible query answering for conjunctive queries. Throughout this article we assume a logical language $\mathscr{L}$ consisting of a finite set of predicate symbols (for example denoted *Ill*, *Treat* or $P$), a possibly infinite set *dom* of constant symbols (for example denoted Mary or $a$), and an infinite set of variables (for example denoted $x$ or $y$). A query formula $Q$ is a conjunction of atoms with some variables $X$ occurring freely (that is, not bound by variables); that is, $Q(X) = L_{i_1} \wedge \ldots \wedge L_{i_n}$. The CoopQA system [1] applies three generalization operators to a conjunctive query (which – among others – can already be found in the seminal paper of Michalski [2]). In this paper we focus only on the **Anti-Instantiation** ($AI$) operator that replaces a constant (or a variable occurring at least twice) in $Q$ with a new variable $y$.

*Example 1.* As a running example, we consider a hospital information system that stores illnesses and treatments of patients as well as their personal information (like address and age) in the following three database tables:

| *Ill* | *PatientID* | *Diagnoses* |
|---|---|---|
| | 8457 | Cough |
| | 2784 | Flu |
| | 2784 | Bronchitis |
| | 8765 | Asthma |

| *Treat* | *PatientID* | *Prescription* |
|---|---|---|
| | 8457 | Inhalation |
| | 2784 | Inhalation |
| | 8765 | Inhalation |

| *Info* | *PatientID* | *Name* | *Address* |
|---|---|---|---|
| | 8457 | Pete | Main Street 5, Newtown |
| | 2784 | Mary | New Street 3, Newtown |
| | 8765 | Lisa | Main Street 20, Oldtown |

The example query $Q(x_1, x_2, x_3) = \mathit{Ill}(x_1, \mathit{Flu}) \wedge \mathit{Ill}(x_1, \mathit{Cough}) \wedge \mathit{Info}(x_1, x_2, x_3)$ asks for all the patient IDs $x_1$ as well as names $x_2$ and addresses $x_3$ of patients that suffer from both flu and cough. This query fails with the given database tables as there is no patient with both flu and cough. However, the querying user might instead be interested in the patient called Mary who is ill with both flu and bronchitis. For $Q(x_1, x_2, x_3) = \mathit{Ill}(x_1, \mathit{Flu}) \wedge \mathit{Ill}(x_1, \mathit{Cough}) \wedge \mathit{Info}(x_1, x_2, x_3)$ an example generalization with AI is $Q^{AI}(x_1, x_2, x_3, y) = \mathit{Ill}(x_1, \mathit{Flu}) \wedge \mathit{Ill}(x_1, y) \wedge \mathit{Info}(x_1, x_2, x_3)$. It results in an non-empty (and hence informative) answer: $\mathit{Ill}(2748, \mathit{Flu}) \wedge \mathit{Ill}(2748, \mathit{Bronchitis}) \wedge \mathit{Info}(2748, \mathit{Mary}, \text{‘}\mathit{New\ Street\ 3}, \mathit{Newtown}\text{‘})$.

## 2 Similarity Measures

Based on feature sets of two objects $a$ and $b$, similarity between these two objects can be calculated by means of different similarity measures. That is, if $A$ is a feature set of $a$ and $B$ is the corresponding feature set of $b$, then $A \cap B$ is the set of their common features, $A \setminus B$ is the set of features that are only attributed to $A$, and $B \setminus A$ is the set of features that are only attributed to $B$. We obtain the cardinalities of each set: $l = |A \cap B|$, $m = |A \setminus B|$, and $n = |B \setminus A|$ and use them as input to specific similarity measures. In this paper, we focus on the ratio model [3] (in particular, one of its special cases called Jaccard index).

**Definition 1 (Tversky's Ratio Model [3], Jaccard Index).** *A similarity measure sim between two objects a and b can be represented by the ratio of features common to both a and b and the joint features of a and b using a non-negative scale f and two non-negative scalars $\alpha$ and $\beta$. The* Jaccard index *is a special form of the ratio model where $\alpha = \beta = 1$ and f is the cardinality $|\cdot|$:*

$$sim_{jacc}(a, b) = \frac{|A \cap B|}{|A \cap B| + |A \setminus B| + |B \setminus A|} = \frac{|A \cap B|}{|A \cup B|} = \frac{l}{l + m + n}$$

Ferilli et al [4] introduce a novel similarity measure that is able to also differentiate formulas even if $l = 0$; this measure is parameterized by a non-negative scalar $\alpha$. We call this similarity measure $\alpha$-similarity and let $\alpha = 0.5$ by default.

**Definition 2 ($\alpha$-Similarity [4]).** *The $\alpha$-similarity between two objects a and b consists of the weighted sum (weighted by a non-negative scalar $\alpha$, and adding*

*1 to the numerators and 2 to the denominators) of the ratios of shared features divided by the features of a alone and the features of b alone whenever $a \neq b$:*

$$sim_\alpha(a,b) = \alpha \cdot \frac{|A \cap B| + 1}{|B| + 2} + (1-\alpha) \cdot \frac{|A \cap B| + 1}{|A| + 2} = \alpha \cdot \frac{l+1}{l+n+2} + (1-\alpha) \cdot \frac{l+1}{l+m+2}$$

*In case $a = b$ the similarity is 1: $sim_\alpha(a,a) = 1$.*

## 3 Similarity for Anti-Instantiation

We calculate the similarity between the original query $Q$ and a query $Q^{AI^*}$ obtained by the AI operator. We concentrate on the following sets of features:

**Predicates in the query:** The predicates of $Q$ and $Q^{AI^*}$ are identical: $Pred(Q) = Pred(Q^{AI^*})$ leading to similarity 1 on the predicate feature.
**Constants in the query:** The set of constants in $Q^{AI^*}$ might be reduced compared to $Q$: $Const(Q^{AI^*}) \subseteq Const(Q)$; we have $l \leq 0$, $m \leq 0$ and $n = 0$.
**Variables in the query:** Because each AI step introduces a new variable, we have $Vars(Q) \subseteq Vars(Q^{AI^*})$ and hence $l \leq 0$, $m = 0$ and $n \leq 1$.
**Star of a literal:** For each literal $L_i$ of $Q$ the amount of connections to other literals is always greater or equal to the amount of connections in $Q^{AI^*}$. We borrow the definition of a star of a literal [4] that contains all predicate symbols of other literals that share a term with the chosen literal. We denote $Terms(L_i, Q)$ the set of terms of literal $L_i$ in $Q$.

**Definition 3 (Star of a literal [4])).** *For a literal $L_i$ in a given query $Q$ we define the* star *of $L_i$ to be a set of predicate symbols as follows*

$$\begin{aligned} Star(L_i, Q) = \{P \mid\ & \text{there is } L_j \in Q, i \neq j, \text{such that } L_j = P(t_1, \dots t_k) \text{ and} \\ & Terms(L_j, Q) \cap Terms(L_i, Q) \neq \emptyset\} \subseteq Pred(Q) \end{aligned}$$

Hence, $Star(L_i, Q^{AI^*}) \subseteq Star(L_i, Q)$ and $l \leq 0$, $m \leq 0$ and $n = 0$.
**Relational positions of a term:** Lastly, we borrow the notion of relational features from [4]. Such a relational feature of a term is the position of the term inside a literal $L_j = P(t_1, \dots t_k)$: If a term $t$ appears as the $h$-th attribute in literal $L_i$ (that is, $t_h = t$ for $1 \leq h \leq k$), then $P.h$ is a relational feature of $t$. Let then $Rel(t, Q)$ denote the multiset of all relational features of a term $t$ in query $Q$. For a term $t$ in $Q$ some its positions might be lost in $Q^{AI^*}$. Hence, $Rel(t, Q^{AI^*}) \subseteq Rel(t, Q)$ and $l \leq 0$, $m \leq 0$ and $n = 0$.

*Example 2.* The example query $Q(x_1, x_2, x_3) = Ill(x_1, Flu) \wedge Ill(x_1, Cough) \wedge Info(x_1, x_2, x_3)$ can be generalized (by anti-instantiating cough with a new variable $y$) to be $Q_1^{AI}(x_1, x_2, x_3, y) = Ill(x_1, Flu) \wedge Ill(x_1, y) \wedge Info(x_1, x_2, x_3)$. Another possibility (by anti-instantiating one occurrence of $x_1$ with a new variable $y$) is the query $Q_2^{AI}(x_1, x_2, x_3, y) = Ill(y, Flu) \wedge Ill(x_1, Cough) \wedge Info(x_1, x_2, x_3)$. Summing all features (predicates, constants, variables, stars and relational) and dividing by 5 gives us the overall average for each similarity measure and for each formula: The first query $Q_1^{AI}$ (with an average Jaccard index of 0.81 and an average $\alpha$-similarity of 0.84) is ranked very close to the second query $Q_2^{AI}$ (with an average Jaccard index of 0.80 and an average $\alpha$-similarity of 0.84) because while more constants are lost in $Q_1^{AI}$ more joins are broken in $Q_2^{AI}$.

Next, we analyze the effect of multiple applications of the AI operator on the similarity values. We have the following monotonicity property: if $A$ is a feature set of the original $Q$, $B$ is the corresponding feature set of $Q^{AI^*}$, and $C$ is the corresponding feature set of a query $Q^{AI^+}$ such that $Q^{AI^+}$ can be obtained from $Q^{AI^*}$ by applying more AI steps, then we have that either a) more variables are added in $Q^{AI^+}$ (that is, $B \setminus A \subseteq C \setminus A$) or b) (in case of all other feature sets) more features lost (that is, $A \setminus B \subseteq A \setminus C$). If one of these inclusions is proper, then the similarity of $Q^{AI^*}$ to $Q$ is higher than the similarity of $Q^{AI^+}$. More formally, for $n = |B \setminus A|$ and $n' = |C \setminus A|$ as well as $m = |A \setminus B|$ and $m' = |A \setminus C|$ and postulating that $n < n'$ or $m < m'$ for any feature, we have that $sim(Q, Q^{AI^*}) > sim(Q, Q^{AI^+})$. Due to this monotonicity property, queries with more anti-instantiations are ranked lower as shown in the following example.

*Example 3.* We consider two steps of Anti-Instantiations on our example query $Q(x_1, x_2, x_3) = Ill(x_1 Flu) \wedge Ill(x_1, Cough) \wedge Info(x_1, x_2, x_3)$. One such generalized query can be $Q^{AI,AI}(x_1, x_2, x_3, y, z) = Ill(y, Flu) \wedge Ill(x_1, z) \wedge Info(x_1, x_2, x_3)$ with two new variables $y$ and $z$ (which is a combination of the two AI steps of $Q_1^{AI}$ and $Q_2^{AI}$). The query with two anti-instantiations is ranked below the queries with one anti-instantiation: 0.63 for the Jaccard index and 0.73 for $\alpha$-similarity. Queries with one anti-instantiations would hence preferably answered.

## 4 Discussion and Conclusion

We applied two similarity measures (Jaccard index and $\alpha$-similarity) to evaluate the syntactic changes that are executed on conjunctive queries during anti-instantiation and can hence support the database system to intelligently find relevant information for a user. A comprehensive similarity framework that respects all possible combinations of the operators DC, GR and AI (as introduced and analyzed in [1]) is the topic of future work as well as a comparison to related approaches and the consideration of semantic (term-based) similarity.

## References

1. Inoue, K., Wiese, L.: Generalizing conjunctive queries for informative answers. In: Flexible Query Answering Systems, Springer (2011) 1–12
2. Michalski, R.S.: A theory and methodology of inductive learning. Artificial Intelligence **20**(2) (1983) 111–161
3. Tversky, A.: Features of similarity. Psychological review **84**(4) (1977) 327–352
4. Ferilli, S., Basile, T.M.A., Biba, M., Mauro, N.D., Esposito, F.: A general similarity framework for horn clause logic. Fundamenta Informaticae **90**(1-2) (2009) 43–66