Context-dependent 'Near' and 'Far' in Spatial Databases via Supervaluation

Michael Minock and Johan Mollevik

Department of Computing Science Umeå University, Sweden
Phone: +46 90 786 6398
Fax: +46 90 786 6126
Email: mjm@cs.umu.se, johang@cs.umu.se

Abstract

Often we are interested to know what is 'near' and what is 'far' in spatial databases. For instance we would like a hotel 'near' to the beach, but 'far' from the highway. It is not always obvious how to answer such nearness questions by reducing them to their crisp counterparts 'nearer' or 'nearest'. Thus we confront the vague and context-dependent relation of *near* (and *far*). Our approach follows a supervaluation tradition with a limited representation of context. The method is tractable, learnable and directly suitable for use in natural language interfaces to databases. The approach is based on logic programs supervaluated over a set of context-dependent threshold parameters. Given a set of rules with such unconstrained threshold parameters, a fixed parameter tractable algorithm finds a setting of parameters that are consistent with a training corpus of context-dependent descriptions of 'near' and 'far' in scenes. The results of this algorithm may then be compiled into view definitions which are accessed in real-time by natural language interfaces employing normal, non-exotic query answering mechanisms.

1. Introduction

A difficulty in natural language interfaces to databases (or knowledgebases) has been an adequate treatment of vagueness. For example when we ask for "a *near by* Indian restaurant", what exactly do we mean? While related questions involving the comparative and superlative forms (e.g. "is Ghandi's *nearer* than Taj Mahal?","which Indian restaurant is the *nearest?*") have crisp answers, which restaurants qualify as 'near' seems open to interpretation and arbitrary. In short, 'near' is vague.

Figure 1 depicts the general situation we model. A speaker asks for objects of type R (e.g. *Restaurants*) that qualify as members in the vague predicate V (e.g. *Near*). As a result a subset A of R is reported back to the speaker using the description C (e.g. "Ghandi's and Taj Mahal are both within 300 meters of your current position."). One obvious truism is that the determination of A

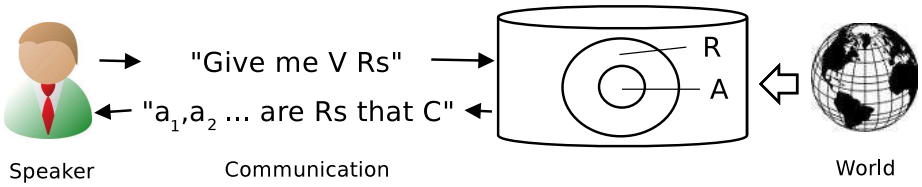

Figure 1: The basic framework

is *context dependent*. For example how much time does the speaker have for lunch? Is the answer different when it is raining? What if the speaker requests near by hospitals rather than near by restaurants? Does requesting hospitals make a difference in the determination of what distance qualifies as near? Does the number of restaurants in the vicinity of the speaker influence the distance threshold of what qualifies as *Near*? These questions hint at the strong role of context in the interaction depicted in figure 1. Of course context is a rather broad notion so let us stipulate the following three types:

> **Speaker context** involves the speaker's goals, capacities and preferences. In the case of 'near', we may ask if the speaker is walking or driving, if they are in good health, if they are under time pressure, etc.
>
> **World context** involves every thing that holds in the world external to the speaker's mind. In practice this will either be recorded in the modeled reality of the database or not be explicitly modeled. For example our database might track the location of the speaker and restaurants, but not track the weather.
>
> **Communication context** involves the actual request of the speaker. Because we restrict the form of communication so rigidly in figure 1, this context is simply what type of objects the speaker is requesting (i.e. R) and the vague predicate (i.e. V).

While representing communication context has been largely handled by restricting our attention to only types of communication depicted in figure 1, we still have the daunting task of representing speaker and world context. Surely we must feel some trepidation [23]. Are we really prepared to build rich models of the user's wants, needs and capabilities? Likewise are we prepared to attempt to formally describe how our database relates to the greater world? Perhaps we could try, but then again perhaps we should just give up and refer to separate contexts in the simplest possible way. That is by merely stipulating that contexts (e.g. c_1) exists, whatever they are, and by giving them descriptive names (e.g. c_1 is named "a 5 minute walk").

In essence we elect this simple approach to context by extending our vague predicate $V(x)$ to $V(x, c)$. Thus, assuming that I am standing at the capital building in Washington DC, we might assert $\neg Near(WhtHouse, \text{'5-min-walk'})$

but $Near(WhtHouse, \text{'afternoon-day-trip'})$. We might elect to say neither $Near(WhtHouse, \text{'15-min-walk'})$ nor $\neg Near(WhtHouse, \text{'15-min-walk'})$ if it is not clear either way. Other questions might be to ask whether there exists a context c_i where $Near(Tokyo, c_i)$? Sure, when $c_i =$ 'inter-galactic-space-travel'. In fact it seems that for any vague predicate, we can say that there is some context that makes it true and some context that makes it false.

Now that we have largely side-stepped context, this paper will focus on representing vagueness of 'near' and 'far' in spatial databases. There has been a wealth of prior work in vagueness stretching from antiquity to comprehensive modern treatments (see [20] for an overview). The present work is informed by these developments, but adopts, in the terms of [6], a *computational* rather than a *cognitive* perspective. That is we seek to support simple aspects of vagueness through leveraging modern relational database systems and theorem provers limited to tractable classes of first-order logic. Our long term goal is to robustly and efficiently support important and well circumscribed classes of vagueness without necessary recovering all the nuances of the phenomena. The work described in this paper is part of this project and addresses the special case of representing 'near' and 'far' in spatial databases. The practical motivation for focussing on 'near' and 'far' is that questions to GISs are often couched in such terms (e.g. "which Indian restaurants are near to the university?", "which hotels are near to the white house but far from a highway?") and answers or descriptions of spatial scenes could be described using such vague spatial predicates (e.g. "It's the Starbucks near to the Chinatown metro stop."). Practically all natural language interfaces to GIS eschew this problem and instead focus on qualitative relations (e.g. give the objects that overlap one another) or they work hard to answer 'near' questions using their crisp counterparts of 'nearer' and 'nearest'.

1.1. Organization of this article

This article is an extension of an earlier conference article [14] and holds a similar structure, but presents at greater depth and with a wider discussion of alternative and future work. The work is also more focussed on the case of 'near' and 'far', and does not follow up on the more extended cases of 'next-to' and 'between' that were briefly entertained in the conference paper. Section 2 provides a review of work in vagueness and in particular vague relations in spatial databases. Section 3 presents our approach basing it on a concrete example for the North Western Washington DC portion of the OpenStreetMap database. The example illustrates the essence of our approach. Section 4 discusses our prototype implementation. Section 5 presents a wide ranging discussion of the work as well as criticisms and future directions. Section 6 summarizes and concludes.

2. Background

2.1. General accounts of vagueness

Vague relations are characterized by *borderline cases* and *inquiry resistance*. Borderline cases means that there are cases that don't seem to be clearly in or out of the relation and inquiry resistance means that no amount of further information can decide the case. Inquiry resistance distinguishes vagueness from *ambiguity* for, in general, ambiguity can be resolved with further dialogue. If a user requests "list the buses that travel down D st." does this include buses that don't actually stop on D street? Once that issue is decided the question essentially becomes crisp, thus it does not resist inquiry. But in a given context if a restaurant is neither 'near', nor 'far' from me, then giving a more accurate measure of the distance does not often help. Although it can be argued that making the context of the question more specific might bring the relation closer to being crisp, we assume here that even under precise contexts vague relations are still inquiry resistant.

A comprehensive treatment of vagueness is Kees van Deemter's recent book *Not Exactly: In Praise of Vagueness* [20]. The book is an informative and entertaining look at vagueness in many of it guises and it reviews the main theoretical approaches to representing the phenomena. The book bases its definition of vagueness on the sorites paradox of Eubulides of Miletus of the fourth century B.C.E. The sorites paradox (also known as the paradox of the stone heap) asks how many stones make a heap. Since one stone does not make a heap, and since in general adding one stone to a collection should not change its status, then paradoxically we should be able to continue adding stones to the collection with it never attaining the status of a 'heap'. A more modern version of this paradox that makes explicit the notion of perceived differences, starts with assuming that we have a person that is 151 centimeters tall that we refer to as 'short'. If we stand this person next to a person that is only a hair's width taller (perceptually the two subjects appear to be the same height), then we must state that the other person is also 'short'. Of course by this line of reasoning we will conclude that people of arbitrarily large height are 'short'. Clearly somewhere the induction step must break down.

More abstractly, in van Deemter's terminology, there are three fundamental properties of vague relations (such as $tall(x)$ or $near(x, y)$) based on gradable measures (such as height and distance): *admissibility*, *tolerance* and *non-transitivity*. In the example of nearness:

> **Admissibility** states that if some object is 'near' at a given distance, then if it were moved to a shorter distance away, then it too would be 'near'.
>
> **Tolerance** states that if an object is 'near', then if it is moved an imperceptible distance away, it still remains 'near'.
>
> **Non-transitivity** states that if object x is 'near' to object y and object y is 'near' to object z, then it is not necessarily the case that object x is 'near' to object z.

We will adhere easily to admissibility and non-transitivity, though as we will see, we will struggle later with tolerance.

Van Deemter's book does a thorough job of presenting and contrasting the linguistic and semantic approaches to handling sorites problems, with special focus on the example of tall. This includes the naive method of specifying thresholds, supervaluation, Kamp's notion of incoherent contexts, introspective accounts, fuzzy logic based approaches, and finally probabilistic logic based approaches. We refer the interested reader to [20].

2.2. Supervaluation

The tradition which guides the present work, is *supervaluation* (see [7, 12] for a philosophical description and [8, 18, 15, 3] for practical approaches that, broadly speaking, can be classified as employing supervaluationistic techniques). From [12],

> According to [supervaluationist] theory, a sentence is true if and only if it is true on all ways of making it precise. This yields borderline case predications that are neither true nor false, but classical logic is preserved almost entirely.

In this paper this is captured by parameterizing the definition of vague predicates with various thresholds constants which are not explicitly set. A *precisification* of the vague predicate is a setting of the relevant threshold parameters to numerical values that are consistent with observations. A statement is *supertrue* if it is true over all possible precisifications. To illustrate, let us define 'near' as $(\forall x)(x < n_1 \Leftrightarrow near(x))$ and assume observations $near(2)$ and $\neg near(10)$. For simplicity, let us assume that our universe of distances is the whole numbers between 0 and 20. Given the observations we have 8 consistent precisifications of our rule. But no matter what the actual setting of n_1 is, the rules above are sufficient to deduce that $near(1)$. Thus $near(1)$ is supertrue, $near(11)$ is superfalse and $near(5)$ is neither supertrue or superfalse, being either true or false with different consistent settings of the parameter n_1. This formulation enforces the property of admissibility and non-transitivity, but does not directly address tolerance. Issues of tolerance aside, what we have is a fairly reasonable formulation for practical application. While one can solve for consistent settings of n_1 in a straight forward way (e.g. $n_1 = 3$ is consistent with the observations), this is not necessary if one wishes to simply use the system to deduce membership or non-membership in vague relations.

There is a limitation of the above system that should be mentioned. Although the formulas given do not determine a specific value for n_1, each model does set a specific value for the threshold. Where this becomes tricky is when we introduce another parameter that determines when a distance is definitely not 'near'. Let us say for example that we introduce the rule: $(\forall x)(x > n_2 \Leftrightarrow \neg near(x))$. Under this system, because for each model a given distance must be either $near$ or $\neg near$, we unwittingly induce the constraint that $n_2 = n_1 - 1$. This is unfortunate, because once we determine parameter settings, we would

like for the resulting logic to model inquiry resistance by remaining agnostic about the determination of the vague predicates for values in some gap. For example in the most conservative setting[1] of parameters for the above example is $n_1 = 3$ and $n_2 = 9$. Instead of embracing partial logic, a work around is to simply introduce an 'opposite' vague predicate $far(x)$ so that we have the rule, $(\forall x)(x > n_2 \Leftrightarrow far(x))$ and the rule that determines mutual exclusion: $(\forall x)(near(x) \rightarrow \neg far(x))$. Thus we can now have consistent settings of threshold parameters under classical first-order logic where a distance is provably *near*, provably far or neither. This gives us our 'gaps' that are needed to model inquiry resistance.

There has been a series of practical works that model vagueness via supervaluation. The work in [8] was one of the first attempts to fix parameters in rule based systems by letting domain knowledge interact with rules to give tighter bounds on intervals. The work is similar to the work carried out here, but no attempt to identify tractable classes of problems was undertaken. Other important work in supervaluation is Bennett's work on VAL (Vague Adjective Logic) [2] and Pulman's work on vague predicates and degree modifiers [15].

2.3. Vague relations in spatial databases

Typically spatial databases represent regions by adding geometric types (`POINT`, `LINE`, `POLYGON`, etc.) to the basic attribute types (e.g. `INT`, `VARCHAR`, `DATE`, etc.). Such geometric types (e.g. as proposed in SQL/MM) have a host of well-defined operators (e.g. *overlaps*, *contains*, *disjoint*, *strictly-above*, *does-not-extend-to-the-right-of*, etc.) and functions (*distance*, *area*, etc.) that can be used as conditions or terms in queries. The canonical types of crisp queries are *point queries* (e.g. "what park am I currently in"), *range queries* (e.g. "what are the Chinese restaurants between H and F streets and 9th and 11th streets?"), *nearest neighbors* (e.g. "Where is the nearest ATM from the corner of 9th street and F street?") and *spatial joins* (e.g. "give Indian restaurants within 100 meters of a metro stop."). The use of R-tree indexes [10] and their generalization ([11]) give log-based access to spatially arranged objects, by indexing on the bounding rectangle of polygon regions and line segments. While spatial databases provide a standard set of basic spatial operators and functions we seek to support spatial predicates 'near' and 'far' which are of a vague, context-dependent nature.

While many researchers have pursued fuzzy logic approaches [5, 1, 21] in spatial databases, it should be remarked that most of these approaches have focussed on what could be called *indeterminate* spatial relations [17] of fuzzy objects, rather than focus on the vague relations *near* (and far) that obtain between crisp objects. An indeterminate relation considers an object that has unknown shape, such as a river that may widen during rainy season or a forest whose boundary may taper off into a clearing. While such objects have unknown exact shape, there does exist some shape and thus, in principle, there

[1] A conservative setting of parameters selects the widest gaps between opposite relations consistent with observations.

are answers to basic spatial predicates (e.g. RCC8 relations [16]) and functions (e.g. distance) evaluated over them. A common way to represent such indeterminate regions is through the so-called *egg yolk* method [4]. The greatest possible extent of a region is represented as well as the most compact possible region and the calculus of the exact spatial predicates and functions gives rise to three-valued logics and intervals of possible values. Some recent work that flirted more directly with vague spatial relations is [13]. While they explicitly side step "vague, ambiguous and context dependent expressions" via a controlled language approach, they provide an interesting definition of 'between' that can be thought of as being on the cusp of becoming a vague relation.

3. An Approach to Context-Dependent *Near* and *Far*

Our approach to represent $Near$ is based on supervaluation over multiple contexts. Returning to figure 1, given a user's question for Rs that are V (e.g. "*Restaurants* that are *near*") under context c, we apply the vague predicates at the intensional level. That is the set of answers A is:

$$\lambda c.\{x|R(x) \wedge V(x,c)\} \text{ where } V(x,c) \Leftrightarrow n(c) < val(x)$$

We focus on the case where R is a basic type predicate (for example $Church$, Pub or $Museum$ and V is a binary (as opposed to unary) vague predicate $near$ (or far) extended with a context argument. Specifically $near(x,y,c)$ and $far(x,y,c)$ are true when object x is 'near' to (or 'far' from) object y in context c. The val function represents the 'distance' between the geometries of two objects. For now let us assume this distance function is the straight line distance metric Δ_{sld} and that the function geo maps from object ids to their associated geometries (i.e. points, polygons or lines).

3.1. The definition of the context-dependent Near and Far

Ignoring types for now, the main rules that define the vague predicates are:

$$(\forall x)(\forall y)(\forall c)(near(x,y,c) \Leftrightarrow \Delta_{sld}(geo(x),geo(y)) < low(c))$$
$$(\forall x)(\forall y)(\forall c)(far(x,y,c) \Leftrightarrow high(c) < \Delta_{sld}(geo(x),geo(y)))$$

These rules provide the necessary and sufficient conditions[2] for $near(x,y,c)$ and $far(x,y,c)$. The necessary conditions state that if two objects are 'near' to (or 'far' from) one another in a given context, then the distance between their associated geometries must be less than some threshold low (or greater than some threshold $high$) for the context. The sufficient conditions state that if the distance between two objects associated geometries is less than some threshold low (or greater than some threshold $high$) for a given context, then two objects are 'near' to (or 'far' from) one another in that context. Nowhere

[2] An error in the conference paper [14] was that only sufficient conditions were included.

do we explicitly set these parameters $low(c)$ and $high(c)$, for they are to be constrained by observations.

Note that the functions we are using in our system are partial and have rigid signature restrictions: $geo(x)$ maps object ids to geometries, $\Delta_{sld}(g_i, g_j)$ maps two geometries to a numeric value, and $low(c)$ and $high(c)$ map contexts to numerical values. The relation $<$ is defined over a finite set of numerical values under active domain. The function signature requirements give a finite Herbrand universe for any system specifying a finite set of contexts and a finite set of observations. This along with the fact that rules are limited to the Horn property gives us a tractable algorithm[3].

Finally we model the 'opposite' relation between $near$ and far in a given context via:

$$(\forall x)(\forall y)(\forall c)(near(x, y, c) \Rightarrow \neg far(x, y, c)).$$

3.2. Calculating thresholds from context-dependent observations

By way of example let us consider observations over the contexts c_1 = '10 minute stroll', c_2 = 'five minute sprint in rain' and c_3 = 'short bicycle ride'. In the scene[4] in Figure 2 we may have observations of the following sort:

$$near(m_2, s_1, c_1) \wedge far(m_2, s_4, c_1) \wedge \neg near(m_4, s_5, c_2) \wedge far(s_6, s_2, c_2) \wedge ...$$

Based on a set of such observations, the most conservative setting of the thresholds could be $low(c_1) = 240$, $high(c_1) = 600$, $low(c_2) = 750$, $high(c_2) = 5000$ and $low(c_3) = 780$, $high(c_3) = 4500$. As will be discussed in Section 4, we have a tractable algorithm to calculate these parameters.

3.3. Representing and comparing contexts

While the $near(x, y, c)$ and $far(x, y, c)$ predicates include a context argument, we also wish to associate facts as being true (or false) in contexts. To achieve this we include a predicate $ist(p, c)$ (standing for 'is true') that records that a proposition $p \in \mathcal{P}$ holds in a particular context c. These facts are propositions from an arbitrary finite set $\mathcal{P}$. For example let us say that the propositions are on-foot, raining, day-time, on-bike. Thus we might have the following facts: $ist(\text{on-foot}, c_1)$, $ist(\text{on-foot}, c_2)$ $\neg ist(\text{on-foot}, c_3)$, $\neg ist(\text{on-bike}, c_1)$, $\neg ist(\text{on-bike}, c_2)$, $ist(\text{on-bike}, c_3)$, $ist(\text{raining}, c_2)$. If a context c_i makes no claims about $raining$, then neither $ist(raining, c_i)$ nor $\neg ist(raining, c_i)$ will be asserted.

Given the ist predicate and a set of context dependent observations, we are able to calculate several useful relationships between contexts:

[3]Technically this is a fixed parameter tractable algorithm based fixing the arities of the functions and predicates. See formal definitions and proofs developed in [14]

[4]We use OpenStreetMap data in PostgreSQL extended with PostGIS. The scene in 2 represents a section of Washington, DC and consists of over a thousand unique objects. Each of these has an associated geometry type (e.g. a polygon, line or point) and a set of descriptive attributes that associates name and types.

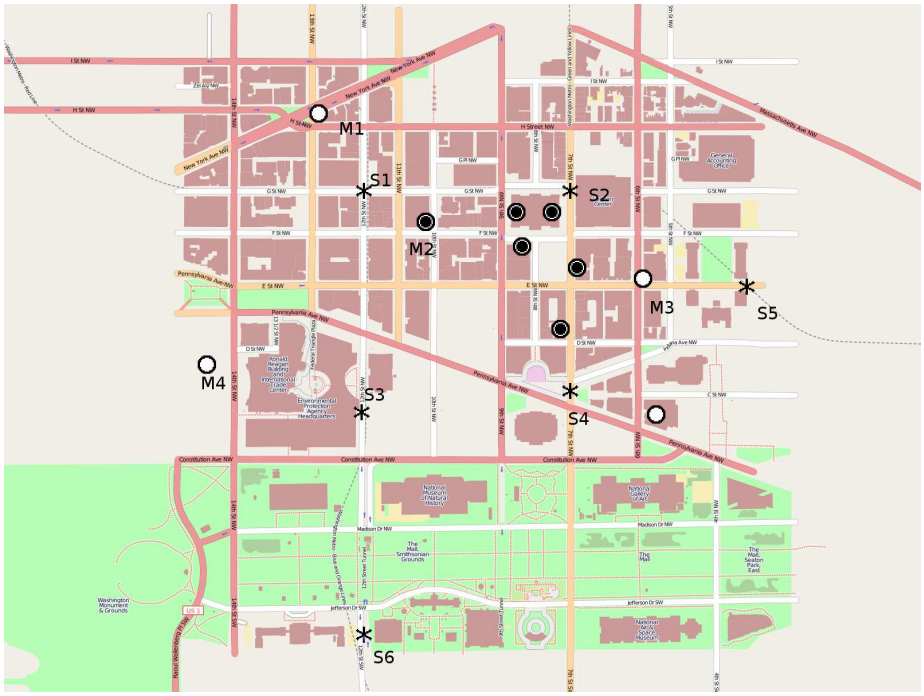

Figure 2: Museums definitely near (•) and possibly near (○) to a metro (∗) in the context of a '10 minute stroll'

Possibly Equal ($c_i \approx c_j$)

When we take into account both observations as well as *ist* facts we may calculate whether two contexts are *possibly equal*. We say that two contexts c_i and c_j are possibly equal (denoted $c_i \approx c_j$) if their equality is consistent with the observations and *ist* facts. This can be tested by merely stating their equality and testing whether the observations and *ist* facts are consistent. In our example *ist* and observations in Section 3.2, none of the contexts can possibly be equal to one another.

Generality ($c_i \sqsupset c_j$)

Ignoring observations we can speak of context c_i being at least as general as a context c_j (denoted $c_i \sqsupseteq c_j$) if and only if $(\forall p)(p \in \mathcal{P} \wedge ist(p, c_j) \Rightarrow ist(p, c_i))$ and $(\forall p)(p \in \mathcal{P} \wedge \neg ist(p, c_j) \Rightarrow \neg ist(p, c_i))$. This can be tested efficiently for each context pair by testing the consistency of these rules with the *ist* facts. A context c_i is more general than c_j (denoted $c_i \sqsupset c_j$) if $c_i \sqsupseteq c_j$, but **not** $c_j \sqsupseteq c_i$. In our example $c_1 \sqsupset c_2$.

More Discerning ($c_i \prec c_j$)

Ignoring *ist* facts, a relationship we calculate is what we call a context being *more discerning* than another context. A context c_i is at least as discerning as

c_j (denoted $c_i \preceq c_j$) if everything that is 'near' in c_i is 'near' in c_j and everything 'far' in c_i is 'far' in c_j. A context c_i more discerning as c_j (denoted $c_i \prec c_j$) if $c_i \preceq c_j$, but **not** $c_j \preceq c_i$. Formally we can compute if $c_i \preceq c_j$ if we add the following rules to above rules and observations and test for consistency:

$$(\forall x)(\forall y)((near(x, y, c_j) \Rightarrow near(x, y, c_i)) \wedge (far(x, y, c_j) \Rightarrow far(x, y, c_i)))$$

In our example above, it is the case that $c_1 \prec c_2$.

3.4. Generating views

Based on the set of contexts, observations, and then the calculated high and low parameters, we can generate views that capture what is 'near' and 'far' in a given context. Thus the view definition of `NEAR(XID, YID, CONTEXT)` provides the objects that are 'near' to one another in the given context. Its form is a straightforward result of pairs $\langle c_i, low(c_i)\rangle$:

```
CREATE VIEW NEAR(id1,id2,context) AS
  (SELECT x.osm_id,y.osm_id,'10 minute stroll' FROM
    planet_osm_point AS X, planet_osm_point AS Y
    WHERE ST_Distance(x.way,y.way) < 240)
  UNION
  (SELECT x.osm_id,y.osm_id,'10 minute stroll' FROM
    planet_osm_polygon AS X, planet_osm_point AS Y
    WHERE ST_Distance(x.way,y.way) < 240)
   ...
  UNION
  (SELECT x.osm_id,y.osm_id,'5 sprint in rain'
   ...
```

This view can be accessed via natural language through the natural language interface system. An analogous view is defined for `FAR(XID, YID, CONTEXT)`

3.5. Multiple distance metrics

In practice, it is often the case that a straight line distance measure is an unreliable metric of 'real' distance. Certainly the existence of a subway system warps the notion of what is 'near'. Modern GIS treatments are getting more sophisticated in calculating such alternative distances. For example taking into consideration the road network and approximate travel times, it is fairly straightforward to develop a metric $\Delta_{driving}$, it is even feasible that we can develop $\Delta_{walking}$ or Δ_{metro}. Given this we could then imagine composing these to generate metric functions such as $\Delta_{walking\&metro}$.

Although we have not yet performed any experiments with these alternative distance metrics, we note here that the impact of this will be to make the distance function definition based on input context. Thus the basic vague predicate definition rules from above become:

$$(\forall x)(\forall y)(\forall c)(near(x, y, c) \Leftrightarrow \Delta(geo(x), geo(y), c) < low(c))$$
$$(\forall x)(\forall y)(\forall c)(far(x, y, c) \Leftrightarrow high(c) < \Delta(geo(x), geo(y), c))$$

This will then necessitate the assignment of distance metrics to various contexts:

$$(\forall x)(\forall y)(\Delta(geo(x), geo(y), c_1) = \Delta_{sld}(geo(x), geo(y)))$$
$$\ldots$$
$$(\forall x)(\forall y)(\Delta(geo(x), geo(y), c_{10}) = \Delta_{driving}(geo(x), geo(y)))$$

While this requires slightly more knowledge engineering, it does not alter the tractability of the approach.

3.6. Type dependence

Based on our intuition that a person being 'near' to a waste paper basket and a person being 'near' to a park should be accorded different distance thresholds, let us entertain the possibility of extending our approach with types (e.g. a restaurant, ATM, park, road, etc.). With respect to the definition of *near* (and far) this would lead to a proliferation of rules including:

$$(\forall x)(\forall y)(\forall c)(near(x, y, c) \Leftrightarrow ATM(X) \wedge Restaurant(Y) \wedge \Delta(geo(X), geo(Y), c) < low(ATM, Restaurant, c))$$

Note that the low parameters now take types as well as the named context as arguments. This is a straight forward extension to the basic case although it results in many more rules – quadratic in the number of types.

A serious limitation with this approach is that monotonicity of logic will result in the most restrictive threshold bounds to be induced between types. This will likely quickly lead to hard to explain inconsistencies that render the approach unworkable. A possible fix would be to stipulate that pairwise disjoints between types. For example the types could included restaurants, ATMs, etc. For example:

$$(\forall x)(Restaurant(x) \Rightarrow \neg Road(x) \wedge \neg ATM(x) \wedge ...)$$
$$...$$

Even if we were able to work with a set of mutually exclusive types, the whole approach suffers from requiring many more observations to fully constrain the thresholds across all type combinations. For this reason we largely dismiss the idea (originally proposed in [14]) of including types in the rules.

4. Prototype Implementation

The approach of section 3 is implemented in an initial LISP prototype integrated with the theorem prover SPASS and PostgreSQL extended with PostGIS over OpenStreetMap data. The implementation consists of two web-based tools: the *teaching tool* and the *query tool*. The teaching tool lets an administrator build and manage contexts (see Figure 4). The output of the learning tool is a set calculated relationships between contexts and the view expressions that define the context-dependent `NEAR` and `FAR` relations. The query tool lets casual users obtain answers to 'near' and 'far' queries via limited natural language dialogue (see Figure 5).

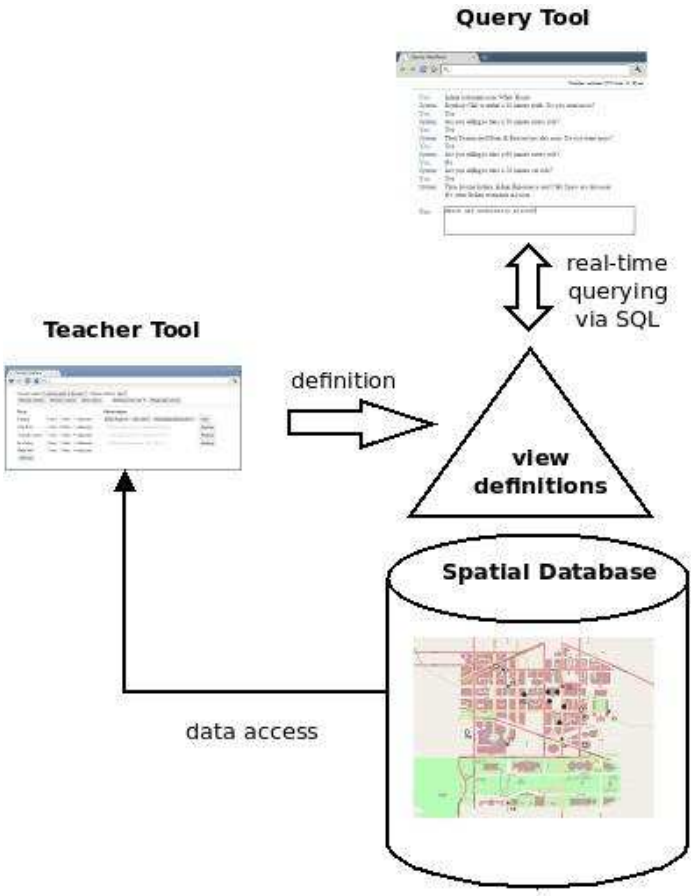

Figure 3: The overall architecture

4.1. The teacher tool

The teacher tool enables an administrator (hereafter referred to as the teacher) to add and name contexts, set associated propositions in contexts to $true$, $false$ or *unknown*, and then to make statements of what is 'near' and 'far' in contexts. Additionally the teacher must specify which distance metric is relevant in the context and must also specify a descriptive name for the context. These descriptive names (e.g. a '5 minute sprint in the rain') will be used to identify the context to casual users.

As the teacher builds up a library of contexts, the system alerts the teacher to related contexts that are possibly equal (see $c_i \approx c_j$) above to a new context that they are asserting. The teacher is encouraged to refrain from defining a context c_i if there already exists a context c_j, where $c_i \approx c_j$ and $c_j \sqsubset c_i$ and $c_j \prec c_i$ (see Section 3.3 for a formal definition of these terms). This captures

Figure 4: The interface to the teacher tool

the intuition that if a context is more specific, but in being so it widens a gap between opposite predicates, then it is probably irrelevant. Likewise a new context c_i that is more general than a prior context c_j ($c_i \sqsubset c_j$) and more discerning $c_i \prec c_j$ should probably supersede c_j and c_j should be removed from the library. These are just recommendations to the teacher however. The teacher can author contexts as they see fit.

Once the teacher has defined a library of contexts, they generate a view definition for the context library. This starts by calculating the most conservative settings of the parameters, followed by a translation of the entire body of contexts into view definitions. This is achievable in polynomial time (see [14]).

Core reasoning services

The core reasoning services in the teaching tool are carried out by the first order resolution theorem prover SPASS [22]. We encode the vague observations, the *ist* statements, the unique names assumption and $<$ over numerical constants (under active domain) in first order logic limited to Horn clauses. These encodings are confirmed to be consistent via a satisfiability check. One tricky issue worth mentioning around the definition of $<$, is that must discretize the active domain of ordered numerical values $d_1, ..., d_n$. This includes Horn-based constraints such as:

$$x = d_i \Leftarrow x < d_i + 1 \wedge d_i - 1 < x$$

Assuming consistency of the defined contexts and observations, based on the reasoning question we are seeking to answer, we encode constraints such as the value of a parameter, or whether two contexts are equal, or any of the other questions from Section 3.3, and test for consistency. The view compilation routine which finds consistent settings of all the *high* and *low* function values is the most expensive of the operations. However this does not require interaction and, again based on the Horn case it is tractable. Further experiments are underway to improve our prototype's performance.

4.2. The query tool

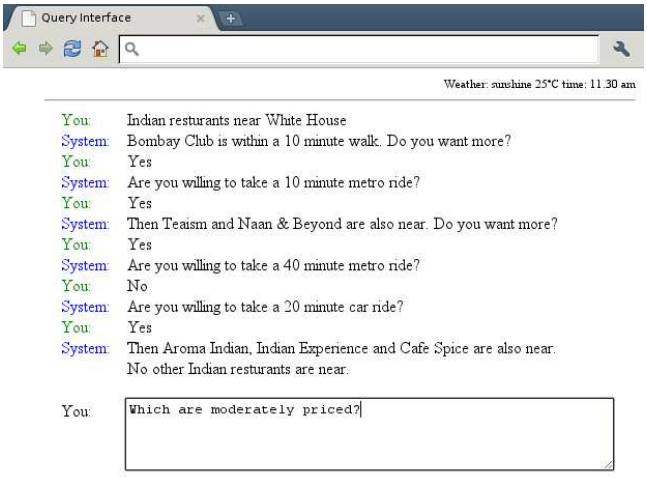

Figure 5: A dialog over the query tool.

Given that we have the view, materialized or otherwise, that defines the vague predicate `NEAR` and `FAR`, the question is how do we use such a view in querying the underlying database. A common use case is that we are unaware of the user's context and they then ask a nearness query. Now given the assumption that the true context of the user matches a context defined in the context library, this will mean that a family of possible thresholds will be possible.

To resolve a more discerning notion of *near* the system will need to obtain from the user their context. This may involve setting predicates based on world state (e.g. the weather or time of day) or via explicit questions (e.g. "Are you driving, walking, or biking?", "Can you take a metro?", etc.) and using this to narrow the set of possible contexts. This could also be based on users stating that objects are definitely near or far, or the user even suggesting the actual threshold values explicitly. The key point is that as the set of possible contexts becomes more constrained, the boundary *low* and *high* parameters will become more and more constrained. The dialogue in Figure 5 follows a strategy where the set of answers common to all possible contexts are identified first, followed by questions that systematically rule out contexts and progressively report all answers in the next less restrictive context, etc. until all possible contexts are addressed or ruled out.

5. Discussion

The original ambition of this article was to cover a wide class of vague spatial preposition including 'between','on','next to','at', etc. As it turned out, we decided to focus our attention on the more limited case of 'near' (and 'far').

Now that we have developed what we feel to be a concrete and practical approach to this simpler case we will be generalizing our approach to more complex and varied cases.

The general interpretation of vague language in this article has been as 'convention'. In general this leads to difficulties. For example consider the non-spatial request for "large cities in Alaska." If we assume that the only relevant gradable measure for a large city is it's population, where would we set the threshold? Obviously this depends on the comparison set. For example do we want a large city by Alaskan standards, or do we want large cities by more conventional standards – in which case perhaps the correct answer is that Alaska does not have any large cities. A possible way to extend vague predicates is with definitions such as "tall means above $n1$ standard deviations from the mean." The $n1$ here remains as a parameter, but one that becomes folded up within a complex calculation that takes the distributional context into account. Formally, in relation to figure 1, this would be:

$$\lambda c.\{x|x \in V^*(\{y|R(y)\}, c)\} \text{ where } x \in V^*(S, c) \text{ when } val(x) \text{ is at least } n^*(c) \text{ standard deviations above the mean } val \text{ for members of } S$$

This paper has based its approach on relational database technology and theorem provers applied over tractable cases of Horn clauses with a finite Herbrand universes. No doubt our approach to context can be criticized for its simplicity given the more sophisticated options [9, 19]. While we slightly extended our notions of context to make certain propositions true or false in context, adding a context argument to predicates is essentially our approach to context in this paper. The main virtue of this is its simplicity and its representation in tractable first-order logic.

Conventional wisdom says that limiting approaches to vagueness to classical first-order logic is too restrictive. In fact Kees van Deemter concludes his book [20] with the analogy of giving up classical logic with the expulsion of Adam and Eve from paradise. It's a painful, but necessary. While we acknowledge that this is probably ultimately true to capture advanced cognitive aspects of vagueness, in this paper we are fighting the expulsion on computational grounds.

6. Conclusions

The approach detailed in this paper treats vagueness in spatial databases as a set of conventions, based on context. The approach is based on definite logic programs with contexts represented as first-class objects and a type of supervaluation over a set of threshold parameters. Given a set of context-dependent rules with open threshold parameters, a tractable algorithm finds a setting of the parameters that are consistent with a training corpus of vague spatial statements. The results of this algorithm may then be compiled into view definitions that may be integrated into normal SQL-based databases. And

in turn such view definitions may be exploited by natural language interfaces employing, normal, non-exotic relational query answering mechanisms.

The need to map vague spatial descriptions to precise logical formulas over spatial databases is a problem that is quite relevant as we develop natural language interfaces for communicating with anything, anywhere. This article has presented a scalable approach to support vague spatial relations for querying spatial databases using natural language phrases such as 'near' and 'far'. In doing so, it has brought to bear work in supervaluation based approaches to vagueness and treatments of context. Experiments have been encouraging and efforts are underway to turn our prototype into a system.

References

[1] D. Altman. Fuzzy set theoretic approaches for handling imprecision in spatial analysis. *Journal of Geographical Information System IBM*, 8(3):271–289, 1994.

[2] B. Bennett. A theory of vague adjectives grounded in relevant observables. In *KR*, pages 36–45, 2006.

[3] B. Bennett, D. Mallenby, and A. Third. An ontology for grounding vague geographic terms. In *FOIS*, pages 280–293, 2008.

[4] A. Cohn and N. Gotts. Representing spatial vagueness: A mereological approach. In *KR*, pages 230–241, 1996.

[5] A. Dilo, R. de By, and A. Stein. A system of types and operators for handling vague spatial objects. *International Journal of Geographical Information Science*, 21(4):397–426, 2007.

[6] M. Duckham and M. Worboys. Computational structure in three-valued nearness relations. In *COSIT*, pages 76–91, 2001.

[7] K. Fine. Vagueness, truth and logic. *Synthèse*, 30:263–300, 1975.

[8] N. Goyal and Y. Shoham. Reasoning precisely with vague concepts. In *AAAI*, pages 426–431, 1993.

[9] R. V. Guha. *Contexts: A Formalization and Some Applications*. PhD thesis, Stanford University, 1991.

[10] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *SIGMOD'84, June 18-21, 1984*, pages 47–57. ACM Press, 1984.

[11] J. Hellerstein, J. Naughton, and A. Pfeffer. Generalized search trees for database systems. In *VLDB*, pages 562–573, 1995.

[12] R. Keefe. Vagueness: Supervaluationism. *Philosophy Compass*, 3(2):315–324, 2008.

[13] S. Mador-Haim, Y. Winter, and A. Braun. Controlled language for geographical information system queries. In *Proceedings of Inference in Computational Semantics*, 2006.

[14] M. Minock. Vague relations in spatial databases. In *Proc. of Applications of Natural Language to Data Bases(NLDB)*, pages 203–208, Cardiff, Wales, 2010.

[15] S. Pulman. Formal and computational semantics: a case study. In *Proceedings of the Seventh International Workshop on Computational Semantics: IWCS-7, Tilburg, The Netherlands, 2007*, pages 181–196, 2007.

[16] D. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In *KR*, pages 165–176, 1992.

[17] A. Roy and J. Stell. Spatial relations between indeterminate regions. *Int. J. Approx. Reasoning*, 27(3):205–234, 2001.

[18] P. Santos, B. Bennett, and G. Sakellariou. Supervaluation semantics for an inland water feature ontology. In *IJCAI*, pages 564–569, 2005.

[19] L. Serafini and P. Bouquet. Comparing formal theories of context in ai. *Artif. Intell.*, 155:41–67, May 2004.

[20] K. van Deemter. *Not Exactly: In Praise of Vagueness*. Oxford University Press, 2010.

[21] F. Wang. Handling grammatical errors, ambiguity and impreciseness in GIS natural language queries. *Transactions in GIS*, 7(1):103–121, 2003.

[22] C. Weidenbach, U. Brahm, T. Hillenbrand, E. Keen, C. Theobalt, and D. Topić. SPASS version 2.0. In *18th International Conference on Automated Deduction*, pages 275–279, Kopenhagen, Denmark, 2002.

[23] T. Winograd and F. Flores. *Understanding computers and cognition - a new foundation for design*. Addison-Wesley, 1987.