Semantic vs. Structural Resemblance of Classes

Peter Fankhauser Martin Kracker Erich J. Neuhold
GMD-IPSI
Integrated Publication & Information Systems Insitute
Darmstadt, Germany 6100
E-mail: (fankhaus,kracker,neuhold)@darmstadt.gmd.de

October 24, 1991

Abstract

We present an approach to determine the similarity of classes which utilizes *fuzzy* and *incomplete* terminological knowledge together with schema knowledge. We clearly distinguish between *semantic* similarity determining the degree of resemblance according to real world semantics, and *structural correspondence* explaining *how* classes can actually be interrelated. To compute the *semantic similarity* we introduce the notion of *semantic relevance* and apply fuzzy set theory to reason about both terminological knowledge and schema knowledge.

1 Introduction

The identification of similar or corresponding concepts forms one of the main steps when investigating different world models and relating them to each other. Apart from its long tradition in document retrieval, this issue has also been investigated in more structured frameworks such as schema independent query formulation, e.g., [Mot90], or database integration, where for a survey you may look at [SL90]. As argued in [GPN91], there should be clear demarcation between structural and semantic considerations when performing such adaption. Whereas most of the approaches to *automatically* determine the similarity of concepts utilize primarily *schema knowledge*, e.g., [NEL86, LNE89], techniques utilizing *semantic knowledge* have also been investigated for this purpose. For example, [KN89] describe the usage of a thesaurus to exploratively integrate a user query with a schema, [SG89] devise a (manual) classification strategy for class-attributes to automatically determine the semantic correspondence of classes and [YSDK91] propose the association of characteristic concepts from a generic concept space to attributes for deriving their similarity.

In this paper we present an extension of the above approaches to determine the semantic resemblance of classes utilizing *fuzzy* and *incomplete* terminological knowledge together with schema knowledge. In Section 2 we introduce a *fuzzy* knowledge model, which organizes the terminology of some application domain in an associative network. Specifically, we show how to infer not explicitly specified relationships between terms. In Section 3 we introduce our notion of *semantic* similarity and discuss the limitations of purely structural heuristics for recognizing similar classes. Finally, we present a technique to determine semantic similarity, which compensates for terms not available in the terminological knowledge base by using also the structure of classes, but still avoids counterintuitive results.

2 Terminological Knowledge

We assume that some classes or at least some of their attributes and/or relationships are assigned with meaningful names in a preintegration phase. Thus knowledge about the *terminological relationship* between the names can indicate the real world correspondence between the classes. This knowledge is represented as an associative network consisting of terms which are related by three kinds of binary relationships:

- *generalization and specialization:* A term a is related by generalization (specialization) to term b if a comprises (is comprised by) b in a taxonomic (e.g. *Person, Employee*) or partitive sense (e.g. *Name, FirstName*).
- *negative association:* Terms are related by negative association if they are complementary (e.g. *Man, Woman*), incompatible (e.g. *isEmployed, counsel*) or antonyms (e.g. *small, big*).
- *positive association:* This is the most general relationship. It relates terms, which are synonyms in some context (e.g. *Title, Heading*), and terms which are *typically* used in the *same* context (e.g. *Letter, Address*).

Since the real world semantics of terms can vary, the relationships are *fuzzy* [Zad71], i.e., they have a strength out of [0,1] assigned. Also one cannot possibly associate all terms with each other. For this reason we infer relationships that are not explicitly specified

by traversing the associative network and investigate the paths in order of their strength.

	p	n	g	s
p	p	n	g/p	s/p
n	n		g/n	s/n
g	g/p	g/n	g	p
s	s/p	s/n	p	s

Table 1: *kind* of paths

	p	n	g	s
p	τ_3	τ_3	τ_2	τ_2
n	τ_3		τ_2	τ_2
g	τ_2	τ_2	τ_3	τ_1
s	τ_2	τ_2	τ_1	τ_3

Table 2: *strength* of paths

The *kind* of a path is determined as shown in Table1. Positive association, generalization and specialization are *transitive*, thus their composition forms the same kind of relationship again. *Anti-transitive* negative associations are not composed at all. Specialization composed with (its *inverse*) generalization results in a (least specific) positive association. The kind of a positive (negative) association composed with a generalization (specialization) depends on the existence of other (possibly weaker) paths between the two terms: As for example shown in Figure 1, the path *Instructor p Student g Person* results in a *generalization* because there also exists a (weaker) generalization path *Instructor g Employee g Person*[1]. Contrarily, *Courses p Instructor p Student g Person* results in a *positive association* because there exists no generalization path between *Courses* and *Person*.

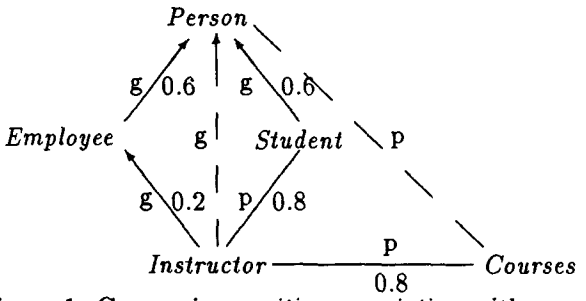

Figure 1: Composing *positive association* with *generalization*

The *strength* of a path depends on the degree of dependence between its relationships. We combine the strengths of relationships using different triangular norms (t-norms) [KF88]. T-norms are two-place functions from [0,1] x [0,1] to [0,1] that are monotonic, commutative and associative. They can be used as axiomatic sceletons for fuzzy set intersection. [SS63] specify an infinite family of t-norms parameterized by the degree of dependence ($-\infty \leq n \leq \infty$) between the membership of identified elements (in our case relationships) in intersected sets. Following [BD86], who have shown that for most practical purposes 3 to 5 t-norms provide sufficient precision while minimizing the complexity of inference, we restrict ourselves to the following t-norms (see Table2):

$$\begin{array}{lll} \tau_1(\alpha,\beta) = max(0, \alpha+\beta-1) & \text{exclusive,} & n=-1 \\ \tau_2(\alpha,\beta) = \alpha\beta & \text{independent,} & n=0 \\ \tau_3(\alpha,\beta) = min(\alpha,\beta) & \text{inclusive,} & n=\infty \end{array}$$

It can easily be shown that $\tau_1(\alpha,\beta) \leq \tau_2(\alpha,\beta) \leq \tau_3(\alpha,\beta)$ holds for all $0 \leq \alpha,\beta \leq 1$. We use the most pessimistic t-norm τ_1 to compose *specialization* with *generalization* or vice versa, assuming that the two relationships were specified according to exclusive categories unless there is also an explicit positive or negative association specified. For example, for *Woman s Person g Student*, *Woman* (neither *Man*) is not a very characteristic aspect of *Student*. The neutral t-norm τ_2 is used to compose positive (negative) associations with specialization and generalization, achieving a kind of fuzzy inheritance of characteristic aspects. Consequently, we apply the optimistic t-norm τ_3 to compose the *transitive* relationships generalization and specialization, respectively, with themselves.

The terminological knowledge model has been successfully applied for assisting schema independent query formulation [Kra91]. In the prototype Knowledge Explorer it supports the acquisition of fuzzy terminological knowledge by a direct manipulation knowledge editor, which utilizes the physical distance between two terms as a metaphor of the strength of their relationship, and allows for the adaption of knowledge bases by adapting strengths according to the subjective feedback given by individual users.

3 Semantic vs. Structural Similarity of Classes

As pointed out in the introduction, purely structural considerations do not suffice to determine the semantic similarity of classes. Consider for example the three schemas in Figure 2[2].

If we compute the percentage of common attributes proposed as heuristic for instance in [NEL86] and realized in [SLCN88] we get: CarOwner ~ ResearchInstitute (66%), CarOwner ~ Person (57%), ResearchInstitute ~ Person (57%), and 0% for the rest. However, intuitively *CarOwner* and *Person* are semantically close,

[1] In our example *Instructor* is a very special case of an *Employee*, because there can be many interim terms with many alternatives like *WhiteCollarWorker*, *Graduate*, and *Teacher*. Thus, even if these terms have not been modelled explicitly, the derived generalization path from *Instructor* to *Person* is rather weak (0.2). On the other hand, *Employee* and *Student* express roles of a *Person* at a similar level of abstraction, therefore both of them are associated to *Person* rather strongly, and so is *Instructor* via the path *Instructor p Student g Person* (0.48).

[2] We deliberately employ a very primitive syntax for class definition with only one simple domain *string* and without methods, as the focus of this paper lies in *semantic resemblance* and not in a fine grained methodology for structural correspondence described in [GPCS91].

CarOwner: *(Schema 1)*
[Name: string, Address: string,
Cars: {string}]

ResearchInstitute: *(Schema 2)*
[Name: string, Address: string,
Topics: {string}]

Letter: *(Schema 3)*
[From: Person, To: Person,
Text: string]
Person:
[Name: string, Address: string,
Send: {Letter}, Receive: {Letter}]

Figure 2: Example Schemas

and even *ResearchInstitute* could be regarded as similar to *Person* when interpreting *Person* as a *Legal Entity*. But in spite of the relatively high percentage of common attributes[3] *CarOwner* and *ResearchInstitute* intuitively do not have any direct meaningful correspondence. Finer grained structural heuristics, e.g., giving a higher relevance to matching keys [HR90], make the situation even worse, because keys tend to have rather unspecific names unless the classes have been defined adhering to very strict naming conventions.

We thus have to consider semantic knowledge also to achieve a more intuitive notion of similarity. The main missing consideration for this purpose is the *semantic relevance* of an attribute to a class. For example *Address* is usually not semantically relevant to *CarOwner*, although (almost) every *CarOwner* has an *Address*, because it is not characteristic to have an *Address* for being a *CarOwner*. The same holds for *ResearchInstitute*. Contrarily, *Address* has to be very relevant for *Letter* because it forms an integral constituent of a *Letter*, distinguishing it from other kinds of papers. It also differentiates it from its other meaning of *Character*. According to such a real world interpretation, it makes sense to look via *Address* (and *Name*) for all *Letters* a *CarOwner* (or a *ResearchInstitute*) has sent or received, but it does not make sense to match *CarOwners* with *ResearchInstitutes* via their *Address*. However, to contribute to the similarity of two classes it suffices for a common attribute (*Address*) to be relevant to just one of them (*Letter*), because it is worthwhile to enhance the information represented in a non relevant attribute (*Address* of *ResearchInstitute*) with relevant information about this attribute (*Letters* the *ResearchInstitute* has *sent* or *received*).

In terms of our fuzzy terminological knowledge

[3]Due to space limitations we restrict ourselves to "common" attributes having the same name (and the same domain - string). Common attributes with different names can be determined via a fuzzy terminological relationship also. In this case each pair of common attributes would contribute to the overall similarity multiplied by the strength of their terminological relationship.

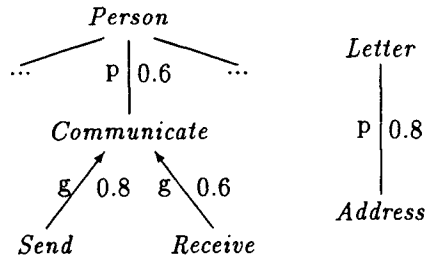

Figure 3: Terminological Knowledge Base

model, semantic relevance can be expressed by a *positive association* or by a *generalization*. For the comparison of the classes in Figure 2 we could use the very simple terminological knowledge base as depicted in Figure 3. The terminological knowledge base contains only terms from schema 3, thus neither information about *CarOwners* nor information about *ResearchInstitutes* is available. In addition, the terminological knowledge base is not completely connected. Therefore we cannot determine the semantic similarity only on a terminological basis, but also have to consider the semantically relevant parts of the structure of classes. Informally, this can be achieved by matching classes only via attributes that are relevant to at least one of the classes. We thus devise the following three steps to determine semantic similarity by combining fuzzy terminological knowledge with relevant structural information:

a) Collect the Semantic Aspects:

The semantic relevance of an attribute (or a relationship) to a class C is determined by investigating the terminological knowledge base, and, if necessary, the relevant schema context of C: An attribute is relevant to C, either if there exists a (derived) terminological relationship between the attribute and C directly, or if the attribute is semantically relevant to a class C', which can be reached via a (schema) relationship that is relevant to C. In the latter case the relevance of the attribute to C' is composed with the relevance of the (schema) relationship between C and C' by the neutral t-norm τ_2. If more than one relevant (schema) path can be used to determine the relevance of an attribute, we choose the most relevant path. If we encounter a cycle, or find only attributes or relationships with a relevance below a certain threshold, we stop. The set of all semantically relevant attributes and relationships is collected in C_{sem}. For example,

$$\begin{aligned} CarOwner_{sem} &= \{\} \\ Person_{sem} &= \{(0.48, Send), (0.48, Receive), \\ &\quad (0.43, Address)\}. \end{aligned}$$

The semantic relevance of Send (Receive) is determined by navigating in the terminological knowledge base on the path *Send (Receive) g [0.8] Communicate p [0.6] Person*. To derive the semantic relevance of *Address* to *Person* we first navigate in the schema from Person to Letter via the relevant relationships *Send* and *Receive*. According to the knowledge base

Address is relevant to *Letter*, and as *Letter* is relevant to *Person* via *Send* and *Receive*, we multiply the maximum relevance of *Send* and *Receive* to *Person* with the relevance of *Address* to *Letter*, max(0.48, 0.48) * 0.9, giving a relevance of 0.43 of *Address* to *Person*.

b) Collect the Structural Aspects:

We define C_{struct} as the set of the immediate, but not semantically relevant attributes of a class[4]:

$$CarOwner_{struct} = \{Name, Address, Cars\}$$
$$Person_{struct} = \{Name\}$$

c) Determine Semantic Similarity:

For two classes C^1 and C^2 we derive the set of terminological relationships between all pairs of elements of C^1_{sem} and C^2_{struct}, of C^1_{struct} and C^2_{sem}, and of C^1_{sem} with C^2_{sem}. The complexity of this comparison in pairs is of order $|C^1_{sem}|*|C^2_{struct}|+|C^1_{struct}|*|C^2_{sem}|+|C^1_{sem}|*|C^2_{sem}|$. Still the actual costs can be kept reasonably low because it can be assumed that at least C^1_{sem} and C^2_{sem} are not very large. The strength of each terminological relationship found in this step[5] is multiplied with the relevance of the attribute in C_{sem}. The maximum of these strengths then determines the similarity of the two classes. For *CarOwner* and *Person* the set of all pairs of elements of $CarOwner_{sem}$ and $Person_{struct}$ and of $CarOwner_{sem}$ and $Person_{sem}$ is empty, whereas the set of all pairs out of $Person_{sem}$ and $CarOwner_{struct}$ = {(0.43, Address)}. Thus the overall similarity of *CarOwner* and *Person* is 0.43. The similarity of *ResearchInstitute* to *Person* is also 0.43 (for a discussion see Section 4), whereas all other similarities are 0.

Of course this technique is still a very coarse approximation of our real world intuition, but it can significantly improve the precision of our similarity measure, without requiring an unrealistically deep and complete model of the world. Our notion of *semantic relevance* of an attribute to a class can be compared to the notion of *significance* of concepts (c.f. attributes, relationships) describing an attribute (c.f. class) in the approach of [YSDK91]. With respect to schema resemblance the main contribution of our approach is that schema knowledge can be used to compensate for missing terminological knowledge.

The actual *correspondence* between *CarOwner* and *Person* can only be determined by applying finer grained structural rules which also specify sufficient conditions for applying some merge operation, e.g., by generating a virtual relationship. In the project KODIM [FKY91], where this research has been carried out, we currently design such rules for database integration, and investigate logic meta-interpretation techniques [UYS89] for generating explanations to facilitate a user-centered, incremental establishment of integrated schemas. Rule-based integration operators which determine possible relationships between classes out of {*subsume, equivalence, overlap, disjoint*} may be found in [SSG+91].

[4] According to our real world intutition *Cars* are of course semantically relevant to *CarOwners*, but due to the incompleteness of the terminolgical knowledge base we cannot determine this.

[5] As we used only name equivalence of attributes in our example all these strengths are 1.

4 Conclusions and future work

We have presented a terminological knowledge model and an approach to determine the semantic similarity of classes on the basis of incomplete and fuzzy terminological knowledge. Our motivation to enhance a loose notion of *structural correspondence* with a notion of *semantic relevance* arouse from the following two observations:

- Structural elements of classes only contribute to semantic similarity if they are semantically relevant. Relying on structural correspondences only, would significantly reduce the *precision* of semantic similarity.
- Although database theory has developed quite powerful notions of structural equivalence, see e.g. [AH88] or [LNE89], it is obviously impossible to formulate *necessary* and sufficient rules for this purpose. In addition, some of these notions introduce and discard names arbitrarily. Thus, too fine grained structural considerations can also reduce the *recall* of a measure for semantic similarity.

Apart from applying the presented techniques for retrieving similar schema constituents in the framework of database integration, we will further develop them into the following directions:

a) Improvement of precision: In our example *CarOwner* and *ResearchInstitute* are equally similar to *Person*. From its definition in *Schema 3*, the interpretation of *Person* as a (not explicitly modelled) *LegalEntity* seems to be valid. However, if *Person* would also have a typically human attribute like *Spouse*, at least its similarity with *ResearchInstitute* should have been weaker. Strictly speaking one cannot even deduce that *CarOwners* are *Persons*, unless they are equipped with a typically human attribute. Thus the relative amount of non-matching, but relevant attributes should be taken into account. Also we will investigate the potential of slightly finer grained structural considerations. For example, although the mathematical relation between the *multiset* of (structured) values maintained by a class (with object identity) and the *set* of values maintained by a relationship is quite obvious, the "semantic distance" between these alternative representations seems to be highly dependent on the actual universe of discourse.

b) Improvement of recall: The usage of simple morphological rules to determine the semantic similarity of terms could improve the degree of covering some application domain with the terminological knowledge base and thereby the recall of our similarity measure.

Also sensible broadening strategies should be investigated: For example, if no similar classes are found, matching attributes which are not semantically relevant could be taken into account too, as long as they do not lead to an explosion of response.

5 Acknowledgements

We would like to express our warmest thanks to Amit Sheth for his concise and helpful comments to several drafts of this paper, which have significantly contributed to establishing its focus on the treatment of semantic resemblance.

References

[AH88] S. Abiteboul and R. Hull. Restructuring hierarchical database objects. *Theoretical Computer Science*, 62:3–38, 1988.

[BD86] P. Bonissone and K. Decker. Selecting uncertainty calculi and granularity: An experiment in trading off precision and complexity. *Machine Intelligence Pattern Recognition*, 4:217–247, 1986.

[FKY91] P. Fankhauser, M. Kaul, and Xu Yi. Kodim - knowledge oriented distributed information management. Projectplan, internal document, GMD, October 1991.

[GPCS91] J. Geller, Y. Perl, P. Cannata, and A. Sheth. Structural integration: A technique of view integration using the object-oriented dual model. Technical report, Bellcore, Sept. 1991. Technical Memorandum TM-STS-017628/1.

[GPN91] J. Geller, Y. Perl, and E.J. Neuhold. Structure and semantics in oodb class specifications. *this issue*, 1991.

[HR90] S. Hayne and S. Ram. Multi-user view integration system (muvis): An expert system for view integration. In *Proc. of the 6th Int. Conf. on Data Engineering*, Feb. 1990.

[KF88] G.L. Klir and T.A. Folger. *Fuzzy Sets, Uncertainty and Information*. Prentice Hall, 1988.

[KN89] M. Kracker and E.J. Neuhold. Schema independent query formulation. In F.H. Lochovsky, editor, *Proc. of the 8th Int. Conf. on Entity-Relationship Approach*, pages 233–247, Toronto, Canada, 1989.

[Kra91] M. Kracker. A fuzzy concept network model and its applications. Technical report, Arbeitspapiere der GMD, October 1991. submitted to FUZZ-IEEE'92, IEEE Int. Conf. on Fuzzy Systems.

[LNE89] J. Larson, S. Navathe, and R. Elmasri. A theory of attribute equivalence in databases with applications to schema integration. *IEEE Trans. Softw. Eng.*, 15(4):449–463, Apr. 1989.

[Mot90] A. Motro. Flex: A tolerant and cooperative user interface to databases. *IEEE Transaction on Knowledge and Data Engineering*, 2(2), June 1990.

[NEL86] S. Navathe, R. Elmasri, and J. Larson. Integrating user views in database design. *IEEE Comput.*, 19(1):50–62, Jan. 1986.

[SG89] A. Sheth and S. Gala. Attribute relationships: An impediment in automating schema integration. In *Workshop on Heterogeneous Database Systems*, Chicago, Dec. 1989.

[SL90] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3), Sept. 1990.

[SLCN88] A. Sheth, J. Larson, A. Cornelio, and S.B. Navathe. A tool for integrating conceptual schemata and user views. In *Proc. of the 4th Int. Conf. on Data Engineering*, pages 176–183, Feb. 1988.

[SS63] B. Schweizer and A. Sklar. Associative functions and abstract semi-groups. *Publicationes Mathematicae Debrecen*, 10:69–81, 1963.

[SSG+91] A. Savasere, A. Sheth, S. Gala, S. Navathe, and H. Marcus. On applying classification to schema integration. In *Proc. of IEEE 1st Int. Workshop on Interoperability in Multibase Systems*, pages 258–261, Kyoto, Japan, April 1991.

[UYS89] L. Ümit Yalçinalp and Leon Sterling. An integrated interpreter for explaining prolog's successes and failures. In H. Abramson and M.H. Rogers, editors, *Meta-programming in logic programming*, chapter 10, pages 191–203. MIT Press, Cambridge, Mass., 1989.

[YSDK91] C. Yu, W. Sun, S. Dao, and D. Keirsey. Determining relationships among attributes for interoperability of multi-database systems. In *Proc. of IEEE 1st Int. Workshop on Interoperability in Multibase Systems*, pages 251–257, Kyoto, Japan, April 1991.

[Zad71] L.A. Zadeh. Similarity relations and fuzzy orderings. *Information Sciences*, 3:177–200, 1971.