FGMOL 2001 Preliminary Version

A Set-Theoretical Approach for the Induction of Inheritance Hierarchies

Wiebke Petersen [1]

Seminar für allgemeine Sprachwissenschaft
Heinrich Heine Universität
Düsseldorf, Germany

1 Introduction

Since in modern linguistic theories (e.g. HPSG) more and more information is regarded as being lexical, the lexicon grows in size and complexity. Much of the redundancy of lexical information can be eliminated by factoring out those properties which are shared by a set of signs and representing them in an inheritance hierarchy. In this way redundancy is avoided and the required memory is minimized.
In view of the size of realistic lexicons the manual construction of such hierarchies is time consuming and error-prone. Therefore techniques for automatically acquiring lexical knowledge are desirable. There are a number of approaches for automatically updating a lexicon by inserting new objects in an existing hierarchy: these attempts are known as the "insertion problem or "learning unknown words (e.g. Light 94 [13], Kilbury et al. 94 [10]). There are some general approaches to construct hierarchies automatically starting with totally flat data, like clustering methods (cf. Michalsky & Stepp 1983 [16]). In the framework of linguistic lexicons there are mainly two non-incremental approaches. Barg presents an algorithm for inducing hierarchies represented in DATR (cf. Barg 96 [2] [1]), which is a widespread formalism for representing lexical information in nonmonotonic multiple inheritance networks. Here the user has to supply criteria for the selection of intermediate hypotheses in order to restrict the search space. Depending on the criteria chosen the resulting DATR theory can be very compact. The other approach, presented by Sporleder, deals with the automatic construction of feature-structure hierarchies using the algorithm Top-Down Attribute Selection (TDAS) (cf. Sporleder 99 [21], Lüngen & Sporleder 99 [14]), which is based on top-down tree induction (TDTI), an algorithm that infers decision trees (cf. Quinlan 86 [20]).

[1] Email:petersew@uni-duesseldorf.de

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

The present paper presents a different approach for the automatic acquisition of lexical knowledge from unstructured data. The main innovation is that the mathematical theory of *formal concept analysis* (cf. Ganter & Wille 99 [6]) is used to extract the monotonic inheritance relations which are inherently given by the data. This approach is implemented and has been successfully tested by computing a hierarchy of the derivational information of German lemmas in the lexical database CELEX.

2 Inducing monotonic hierarchies from unstructured data

2.1 An approach for inducing compact hierarchies

Most problems involving the learning of hierarchies can be reduced to the following problem: Assume a given set of object-attribute pairs [2], which is to be organized in an inheritance network; which inheritance hierarchy captures the inherent relations between these data best? There are at least two general demands on representations of the given data: they have to be consistent and complete with respect to the data. Furthermore a good representation avoids redundancy by capturing generalizations; a representation is said to be redundancy-free if every attribute and every object is stated exactly once. In addition one could require the number of nodes in the constructed network to be minimal. Furthermore, some theories which use inheritance network representations require the networks to have special properties; for instance they could allow only monotonic or single inheritance relations, or they could demand that the network form a join semilattice, so that every set of nodes has a least upper bound.

Table 1 gives an example data-set (taken from Hall 2000 [9]) which is to be organized in an inheritance hierarchy. It is composed of nine objects (phonemes), for each of which a subset of the total set of five attributes (phonological features) applies. One natural way of structuring these data is to take for every object the corresponding set of attributes and order these sets with respect to the superset relation. A *top element* can be added to get a connected partial order. The hierarchy thus obtained is shown in figure 2. It is not free of redundancy, since the attributes "voiced, "continuant and "approximant have to be stated more than once in order to ensure that the objects inherit the correct attributes. This problem arises from the onesided intensional point of view that only the attribute sets are taken into account.
The data from table 1 can be ordered by the sets of attributes applying to the single objects and/or by the sets of objects which have an attribute in

[2] A feature-value pair applying to an object can be seen as one of its attributes. This is a strong simplification, since the appropriateness of a feature for an object cannot be captured anymore. A possible way out is to increase the number of attributes by adding attributes of the kind "feature x is appropriate here.

	consonantal	voiced	continuant	sonorant	approximant
/r/	x	x	x	x	
/l/	x	x		x	x
/?/				x	
/t/	x				
/j/		x	x	x	x
/s/	x		x		
/z/	x	x	x		
/h/			x	x	
/d/	x	x			

Fig. 1. input data

Fig. 2. hierarchy ordered by the attribute sets

common. The latter hierarchy is shown in figure 3, where the object sets are ordered by the subset relation and a bottom element is added to serve as a common subset of all object sets. A redundancy-free solution is received if one combines these two ways of structuring the data (see figure 4). The constructed network is a monotonic and multiple inheritance hierarchy. It is labeled in such a way that each object and each attribute name appears exactly once. This is achieved by attaching each attribute to the highest possible node and each object to the lowest possible; for more details see section 2.2.

Fig. 3. hierarchy ordered by the object sets

Inheritance hierarchies constructed in this way have the desirable property that the number of nodes extends the sum of the number of attributes and the number of objects maximally by two, and they are therefore very compact. Furthermore, they are data-consistent and complete, since a row of table 1 representing the properties of one object can be reconstructed by collecting all the attribute names labeling nodes above the node labeled with the object name. Analogously, a column corresponding to one attribute is given by the

object names which can be found below the attribute.
Some theories require that inheritance networks be join semilattices, so that any two elements have a least upper bound (cf. Carpenter 92 [3]). The partial order shown in figure 4 is not a semi-lattice, since for instance the nodes labeled with /l/ and /r/ do not have a unique least upper bound.
The following section shows how even complete lattices can be induced by using formal concept analysis, a set-theoretic approach which combines the extensional and intensional point of view in one theory.

2.2 Applying formal concept analysis to induce inheritance lattices

Formal concept analysis is a mathematical theory which was especially designed to provide a formal model of knowledge as a tool for communication (cf. Zickwolf 94 [23]). It aims at combining the advantages of a formal representation, like being machine-readable and processible, and a representation that can be presented visually in such a way that it is readable for human beings. Until now it has been applied only to the following linguistic problems: meronymy (cf. Priß 96 [19]), WordNet (cf. Priß 96 [18]), semantics of speech-act-verbs (Großkopf & Harras 99 [8]) and verb paradigms (Großkopf 96 [7]). In the context of this paper it is only possible to list the main definitions of formal concept analysis (FCA); for more details see Ganter & Wille 1999.
FCA starts with the definition of a formal context K as a triple (G, M, I), consisting of a set of objects G, a set of attributes M and a binary incidence relation $I \subseteq G \times M$, where $(g, m) \in I$ means "the formal object g has the formal attribute m. Formal contexts are typically represented in cross tables (see table 1). For any subset of objects $A \subseteq G$, their set of common attributes is defined as $A' = \{m \in M | \forall g \in A : (g, m) \in I\}$. Analogously, the set of common objects for a subset $B \subseteq M$ of attributes is $B' = \{g \in G | \forall m \in B : (g, m) \in I\}$. A formal concept is a pair (A, B), with the properties $A = B'$ and $B = A'$, where A is called the extent and B the intent of the concept. The set of all formal concepts of a context is partially ordered by $(A1, B1) \leq (A2, B2) \Leftrightarrow A1 \subseteq A2 \Leftrightarrow B1 \supseteq B2$. It can be proved that the set of formal concepts together with this partial order form a complete lattice, called the formal concept lattice (see figure 5). In figure 5 only the attribute and the object concepts are labeled, where the attribute concept $\mu(m)$ associated with an attribute m is the greatest concept whose intent contains m, and analogously, the object concept $\gamma(g)$ associated with an object g is the smallest concept whose extent contains g. Labeled as in figure 5, a concept lattice can be seen as a monotonic multiple inheritance hierarchy. Since the hierarchy is constructed on the base of the subset relation, it is excluded that conflicting attributes are inherited from parent nodes. However, the inheritance is not orthogonal in the sense that parent nodes have disjoint sets of attributes. Since orthogonality is stipulated in some theories like HPSG (Pollard & Sag 1987 [17]), this requires special attention when FCA is used as a tool to construct hierarchies.

Fig. 4. partial order of the object and attribute concepts

Fig. 5. concept lattice

The inheritance network of figure 4 is the sub-poset of the concept lattice of figure 5 consisting of the attribute and object concepts. Compared with the network of figure 4, the number of nodes increases in the concept lattice, since nodes are inserted which do not introduce any new attributes or objects but represent greatest lower bounds respectively least upper bounds [3]. If the induced hierarchy is to be used in a formalism which does not require it to be a lattice, it is probably sufficient to use the partial ordered attribute and object concepts. However, it should be considered that lattices are mathematically well-known objects, which are easy to treat computationally, and for which a lot of algorithms exist. And furthermore, since the concept lattice represents for every set of objects their common attributes, in some cases inserting new objects in the hierarchy is simplified.

In most linguistic tasks concerning the lexicon, the universe of the objects is known, and the characteristic attributes are fixed which are to be used to classify the objects, but often the set of objects is too large to check each object-attribute pair. The linguist looks for a selection of objects which provide the same relations with respect to the attributes as the whole universe. Formal concept analysis offers a tool called *attribute exploration* to find such a *complete selection context*. It asks the user if special implications are valid in the context or if a counterexample is known. The structure of a concept lattice can also be described by a set of implications of the form "for every object for which the attributes $m1, m2, ..., mk$ apply, the attributes $a1, a2, ..., ai$ apply too or $\{m1, m2, ..., mk\} \rightarrow \{a1, a2, ..., ai\}$ (cf. Ganter & Wille 99). Looking at the concept lattice, $A \rightarrow m$ is an implication of the context if and only if the attribute node m lies above the infimum of all attribute nodes n with $n \in A$. Therefore {approximant} → {sonorant, voiced} and {consonantal, sonorant} → {voiced} are implications of the example context.

[3] Under certain circumstances the number of nodes increases extremely; for instance, the number of nodes in the concept lattice capturing the derivational information of German lemmas contained in the lexical database CELEX is greater than 72.000; however, if the same information is represented in a network which only includes the attribute and object concepts, the number of nodes is less than 4.000.

3 Some notes on inducing regularities, subregularities and exceptions from flat data

Since lexical information can be structured very well in terms of regular, subregular and exceptional forms, many theories allow elements of nonmonotonic inheritance, for instance default inheritance in DATR (cf. Evans & Gazdar 96 [5]) or default unification in feature-based theories like HPSG (cf. Copestake & Lascarides 99 [4]). Automatic induction of nonmonotonic structures leads to special problems, since the data do not reveal nonmonotonic relations as openly as monotonic ones. The set of possible representations using nonmonotonicity is much larger than the set of monotonic representations. To find a "good representation using default information a method has to be found to decide which information may be overwritten in which case. One way could be to investigate the similarity of attribute classes applicable to a set of objects. If two classes are very similar, one could become a subclass of the other; normally it is more favorable to make the class applying to a larger class of objects the upper class. For applying this method, a measure of similarity is needed; in the framework of formal concept analysis similarity measures have been formalized (cf. Lengnink 96 [12]; Leischner 99 [11]). Another approach which can be used in the processing of exceptions is to allow partial implications, which are valid except for certain exceptions (cf. Stumme 96 [22], Luxemburger 94 [15]).

References

[1] Barg, P., *Automatic inference of datr theories*, in: W. Bock, H.-H. end Polasek, editor, *Data Analysis and Information Systems. Statistical and Conceptual Approaches (Proceedings of the 19th Annual Conference of the Gesellschaft für Klassifikation e. V.,University of Basel,1995-03-8/10)* (1996), pp. 506–515.

[2] Barg, P., "Automatischer Erwerb von linguistischem Wissen. Ein Ansatz zur Inferenz von DATR-Theorien," Niemeyer, Tübingen, 1996.

[3] Carpenter, B., "The Logik of Typed Feature Structures. With Applications to Unification Grammars, Logic Programs and Constraint Resolution." Cambridge University Press, Cambridge, 1992.

[4] Copestake, A. and A. Lascarides, *Default representation in constraint-based frameworks*, Computational Linguistics **25** (1999), pp. 55–105.

[5] Evans, R. and G. Gazdar, *Datr: A language for lexical knowledge representation*, Computational Linguistics **22** (1996), pp. 167–216.

[6] Ganter, B. and R. Wille, "Formal Concept Analysis. Mathematical Foundations," Springer, Berlin, 1999.

[7] Groß kopf, A., *Formal concept analysis of verb paradigms in linguistics*, in: E. Diday, Y. Lechevallier and O. Opitz, editors, *Ordinal and symbolic data analysis*, Springer, Berlin, 1996 pp. 70–79.

[8] Groß kopf, A. and G. Harras, *Begriffliche erkundung semantischer strukturen von sprechaktverben*, in: G. Stumme and R. Wille, editors, *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*, Springer, Berlin, 1999 pp. 273–295.

[9] Hall, T. A., "Phonologie. Eine Einführung," Walter de Gruyter, Berlin, 2000.

[10] Kilbury, J., P. Naerger and I. Renz, *Simulation lexikalischen erwerbs*, in: S. W. Felix, C. Habel and G. Rickheit, editors, *Kognitive Linguistik: Repräsentation und Prozesse*, Westdeutscher Verlag, Opladen, 1994 pp. 251–271.

[11] Leischner, L., *Additive similarity measures*, Technical Report FB4 preprint 2022, Technische Universität Darmstadt, Fachbereich Mathematik (1999).

[12] Lengnink, K., "Formalisierungen von Ähnlichkeiten aus der Sicht der Formalen Begriffsanalyse," Shaker Verlag, Aachen, 1996.

[13] Light, M., *Classification in feature-based default inheritance hierarchies*, in: H. Trost, editor, *Proceedings of KONVENS '94: Verarbeitung natürlicher Sprache. Österreichische Gesellschaft für Artificial Intelligence* (1994), pp. 220–229.

[14] Lüngen, H. and C. Sporleder, *Automatic induction of lexical inheritance hierarchies*, in: J. Gippert, editor, *Multilinguale Corpora. Codierung, Strukturierung, Analyse*, Prague, 1999 pp. 42–52.

[15] Luxenburger, M., *Partial implications*, Technical Report FB4 preprint 1612, Technische Universität Darmstadt, Fachbereich Mathematik (1994).

[16] Michalski, R. S. and R. E. Stepp, *Learning from observation: Conceptual clustering*, , **1**, Morgan Kaufmann, Los Altos, 1983 .

[17] Pollard, C. and I. Sag, *Information-based syntax and semantics*, Technical Report 13, Center for the Study of Language and Information (1987).

[18] Priß, U., *The formalization of wordnet by methods of relational concept analysis*, in: C.Fellbaum, editor, *WordNet - An electronic lexical database and some of ist applications*, MIT-Press, 1996 .

[19] Priß, U., "Relational concept analysis: semantic structures in dictionary and lexical databases," Shaker Verlag, Aachen, 1998.

[20] Quinlan, J., *Induction of decision trees*, Machine Learning **1** (1986), pp. 81–106.

[21] Sporleder, C., "Learning Lexical Generalisations. An Operational Evaluation of Current Machine Learning Methods," Master's thesis, Universität Bielefeld (1999).

[22] Stumme, G., *Attribute exploration with backgroundimplications and exceptions*, in: W. Bock, H.-H. end Polasek, editor, *Data Analysis and Information Systems. Statistical and Conceptual Approaches (Proceedings of the 19th Annual Conference of the Gesellschaft für Klassifikation e. V.,University of Basel,1995-03-8/10)* (1996), pp. 457–469.

[23] Zickwolff, M., *Zur rolle der formalen begriffsanalyse in der wissensakquisition*, in: R. Wille and M. Zickwolff, editors, *Begriffliche Wissensverarbeitung: Grundlagen und Aufgaben*, BI-Wissenschaftsverlag, Mannheim, 1994 pp. 173–190.