REPRINTED FROM:

NATURAL LANGUAGE UNDERSTANDING AND LOGIC PROGRAMMING, II

Proceedings of the Second International Workshop on Natural Language Understanding and Logic Programming
Vancouver, Canada, 17–19 August, 1987

edited by

Veronica DAHL
Associate Professor
[illegible] University
[illegible] Canada

and

Patrick SAINT-DIZIER
I.R.I.S.A.
University of Rennes 1
[illegible] de Beaulieu
[illegible] France

Natural Language Understanding and Logic Programming, II
V. Dahl and P. Saint-Dizier (Editors)

Using Commonsense Knowledge to Disambiguate Word Senses

K. Dahlgren

IBM Los Angeles Scientific Center
11601 Wilshire Blvd., Third Floor
Los Angeles, California 90025
net address: csnet1@ibm.com

A word sense disambiguation algorithm was developed which uses a combination of frequently-occurring fixed phrases, syntactic contexts, and commonsense reasoning to progressively reduce the number of relevant senses. This local combined method employs complex commonsense reasoning only for the difficult cases, and keeps to simpler syntactic and phrasal tests where possible. The algorithm was 96% successful in 2193 concordance occurrences of seven nouns, and 99% successful in 1789 concordance occurrences of four verbs. A PROLOG implementation of the algorithm was written and tested, as will be described in Section 10.

1. Approaches to Word Sense Disambiguation

Computational lexical disambiguation divides into syntactic category assignment such as whether *farm* is a noun or a verb (Milne [1]) and word sense disambiguation within syntactic category. The latter problem is the subject of the present research. Assuming that word senses are listed together under one lexical entry in a given syntactic category, the problem is to select the correct one. One computational method of disambiguation is pattern matching where the surrounding words frequently associated with a sense are used to disambiguate a word. Such methods are powerful and can be used to eliminate 70% of the ambiguity(Black [2]). A second method employs a rich syntactic lexicon which includes selectional restrictions (Gross [3]). A third method uses a combination of structural and conceptual analysis for disambiguation (Black [2]). In the present work a method is proposed which combines three types of information to disambiguate: fixed and frequent phrases, syntactic information and commonsense reasoning. It is similar to Black's approach, but it differs in using a psycholinguistically motivated word meaning representation as the basis of a generalized disambiguation procedure. The advantage of the method is that it employs computationally expensive commonsense reasoning only for the difficult cases, and not for simpler cases. In the data examined, half the cases were simple, and the other half required commonsense reasoning.

Two approaches to simplifying disambiguation were considered and rejected as the starting points (though they are important aspects of the process). One was the idea that domain-specificity could be used to reduce ambiguity. In particular domains certain senses of words become more salient and others recede. For example, in a military domain the noun *company* has a salient sense meaning "a unit of soldiers".[1] In other domains, such as business, this sense is irrelevant, and may be suppressed altogether. Unfortunately, domain-specificity does not eliminate all ambiguity. Firstly, some text is very general, and readers are unsure what terminology will be used. Secondly, even in domain-specific texts, many words remain highly ambiguous. For example, the noun *hand* has sixteen senses and retains ten of them in almost any text.

It was decided to ignore domain-specific senses for the present research and concentrate upon disambiguating widely-used senses. Similar to domain-specificity is the idea of using paragraph topic to disambiguate. For example, if the paragraph topic is house-cleaning, then the sense of *bed* in "make the bed" which means "place to sleep with bedclothes" (as opposed to the sense which means "bedstead") can be selected without any further reasoning. This proposal is attractive because it is computationally efficient, once paragraph topic is determined. However, paragraph topic only partially disambiguates. Other information must be added to paragraph topic information. Since paragraph topic is difficult to determine, the method must be rejected.

2. Local Combined Ambiguity Reduction

In the present research, the hypothesis was tested that a combined method employing fixed and frequent phrases, syntactic properties and complex commonsense reasoning could be used to progressively reduce ambiguity. A computational disambiguation algorithm should take advantage of whatever relatively simple methods exist, especially in light of the exponentially large number of possible interpretations of a text. The hypothesis is that superficial word associations and syntactic structures take less cognitive processing than does commonsense reasoning. Wouldn't it be bewildering if each time the word *hand* was encountered, all sixteen senses were potential candidates in the interpretation? Perhaps frequently-used combinations of words are stored in memory, and accessed similarly to the way compounds are. Perhaps natural language is redundant in providing a number of different clues for semantic interpretation, one of which is just that certain sequences of words can have only one interpretation. If this is true, natural language is not as generative as has been believed, in that some complex phrases are memorized rather than generated. On the other hand, in the more difficult cases where fixed phrases and syntactic tests fail, people disambiguate by using a broad level of conceptual and encyclopedic knowledge. Fortunately, in a psycholinguistically based natural language understanding system, a great deal of the information needed for the difficult cases is already encoded for independent reasons. This research began with the prejudice that paragraph topic would do the lion's share of disambiguating. It turned out that half is done by fixed phrases and syntax and, and half by local commonsense reasoning.

The combined disambiguation method is local, in that it first attempts to employ the directly adjacent words as fixed phrases, then tries syntactic information in the same sentence, then tries commonsense reasoning concerning elements of the same sentence. Only after all of these have failed does it attempt to disambiguate using information in surrounding sentences. During the parse of a sentence, ambiguous nouns are flagged. Resolved ambiguity information is passed from sentence to sentence. Anaphora rules identify referents of nouns and pronouns to the extent possible. The algorithm applies cyclically, to the most deeply embedded sentence within a sentence and upward to the topmost sentence. Since fixed and frequent phrases are computationally simple, and reduce ambiguity substantially, they are tried first. Then syntactic tests are attempted. Finally more complex commonsense reasoning is invoked, in which ontological and generic knowledge is accessed. At each stage, the progressive reduction of the Sense List is checked to see whether the noun has been fully disambiguated, in which case the Sense List is reduced to one member and the algorithm is exited. The algorithm is as follows:

Assume all senses are relevant--set Sense List to all senses
Try all fixed phrases
    Conclusive? Return
    Reductive? Reduce Sense List accordingly.
Try syntactic tests
    Conclusive? Return
    Reductive? Reduce Sense List accordingly.
Use CK reasoning
    Conclusive? Return
    Reductive? Reduce Sense List accordingly.

3. Test of Hypothesis

The local ambiguity reduction method described above was tested in the following way. Seven nouns and four verbs were chosen to be tested against concordances drawn from the Hansard parliamentary corpus, which were generously supplied by Ezra Black. Highly ambiguous words were chosen because it is with these difficult cases that commonsense knowledge is most needed. These were the nouns *office, hand, company, idea, crop, people*, and *school*, and the verbs *work, support* and *use* and *move*. Psycholinguistically based generic representations of these in the Kind Types system (KT) were expanded to account for all senses, and syntactic and fixed phrase rules were written. Employing these, the algorithm fully disambiguates 96% of the 2193 instances of the nouns in the concordances, even though the concordances do not always include the entire sentence in which the word occurred. Of the 91 concordances for which the method failed, only one occurred in a full sentence. The algorithm was 99% successful for 1789 concordances of the verbs.

ENTITY	→	(ABSTRACT V REAL) & (INDIVIDUAL V COLLECTIVE)
ABSTRACT	→	IDEAL V PROPOSITIONAL V QUANTITY V IRREAL
REAL	→	(PHYSICAL V TEMPORAL V SENTIENT) & (NATURAL V SOCIAL)
PHYSICAL	→	(STATIONARY V NONSTATIONARY) & (LIVING V NONLIVING)
NON-STATIONARY	→	(SELFMOVING V NONSELFMOVING)
COLLECTIVE	→	MASS V SET V STRUCTURE
TEMPORAL	→	RELATIONAL V NONRELATIONAL
RELATIONAL	→	(EVENT V STATIVE) & (MENTAL V EMOTIONAL V NONMENTAL)
EVENT	→	(GOAL V NONGOAL) & (ACTIVITY V ACCOMPLISHMENT V ACHIEVEMENT)

Figure 1. Ontological Schema

3.1. Commonsense Knowledge

The disambiguation algorithm is designed to work in the text-understanding system KT. KT is based upon a theory which identifies the form of mental representations of concepts with the form of encyclopedic knowledge about the entities to which these concepts refer (Dahlgren and McDowell [4], [5]). The heart of the KT system is the knowledge representation scheme which encodes word meanings AS commonsense knowledge. Each noun and verb is attached to a commonsense ontology (eg., ABSTRACT vs REAL) and also has an associated generic representation which lists typical and inherent features of entities (Rosch, et al [6], Dahlgren [7]). The theory is similar to other work in commonsense knowledge (Hobbs, et al [8]), but differs in employing cognitively based representation aimed at text understanding rather than at robot control.

In KT nouns and verbs are attached to an ontology of concepts which reflects the way the average person classifies real world entities. Each of these reflects a major class distinction in the commonsense view. The ontology was developed on the basis of verb selectional restrictions (Chomsky [9]), basic cognitive distinctions described in the psychological literature, and philosophical concerns. At each node in the ontological hierarchy entities may cross-classify. For instance, the node REAL cross-classifies as either PHYSICAL, SENTIENT, or TEMPORAL and as either NATURAL or SOCIAL. Thus, from the REAL node alone there are parallel NATURAL and SOCIAL ontologies branching from REAL to PHYSICAL, SENTIENT, and TEMPORAL. Moreover, any individual entity can be multiply attached to the ontology. That is, an individual John might attach to all three of the following: the HUMAN node under ANIMAL, the node PERSON under SENTIENT, and the node CARPENTER under ROLE. Thus the hierarchy is a lattice rather than a tree. The Ontology is given in Figure 1. Figure 2 illustrates the way a few sample words attach to leaves of the ontology.

	Rule
bush	PLANT ← INDIVIDUAL & NATURAL & STATIONARY & LIVING
bear	ANIMAL ← INDIVIDUAL & NATURAL & LIVING & SELFMOVING
John	PERSON ← INDIVIDUAL & NATURAL & SENTIENT
book	ARTIFACT ← INDIVIDUAL & SOCIAL & NONLIVING
car	VEHICLE ← ARTIFACT & SELFMOVING
office	INSTITUTION ← STRUCTURE & SENTIENT & SOCIAL
move	NATURAL & TEMPORAL & NONRELATIONAL & GOAL AND ACHIEVEMENT

Figure 2. Sample Noun Leaf Nodes in the Ontology

In KT, associated with each term are generic descriptions which are bundles of features. The generic descriptions throughout the KT system are based as much as possible on published psycholinguistic data and raw data kindly supplied by several authors (Ashcraft [10]), Rosch et al [6]), Dahlgren [7]), Fehr and Russell [11]), Rifkin [12])). A typical study of this kind would ask subjects to write down in some fixed length of time (1.5 minutes) everything characteristic of a presented category (say elephant). Subjects are surprisingly consistent. The theoretical claim is that if a particular feature is listed by a significant number of subjects, then that feature forms at least part of the representation for the term in the subculture's language (Dahlgren [13]). In KT generic descriptions are encoded in a generic lexicon in two lists for each lexical item. The representation has the form (Sense1,Sense2...) where each sense representation is a list of the form {sense index,ontological node,{typical features list }, {inherent features list}}, as in the first sense of *office* below.

office({1,INSTITUTION,
{hierarchy({management,staff}),
roles({management(manager),
staff(secretary),staff(clerk)})},
{function(business)}},

Experimental psycholinguistics shows that features in generic descriptions fall into predictable patterns of types (Dahlgren [7], Ashcraft [10], Smith and Medin [14]). For example, the feature types for social institutions include hierarchy, roles and functions, while those for physical places include physical descriptions and physical structure. Such patterns are the basis of kind types, which are classes of kinds which share predictable patterns of feature types. Examples of kind types are ANIMAL, ARTIFACT, SOCIAL ROLE, INSTITUTION and PLACE. Terms attached at a node in the ontology inherit all feature types of higher nodes.

Paraphrase versions of the readings for two ambiguous nouns in the study *office* and *hand* are given in Table I. These will be used as examples in Section 4.

Table I. Senses of Two Nouns

office({1,INSTITUTION, business},
{2,PLACE, place of business},
{3,ROLEINITSELF, elective position},
{4,TEMPORAL, service},
{5,INSTITUTION,organization of an elected official}).

hand({1,HUMAN,part of human body},
{2,PRIMATE, part of non-human animal's body},
{3,DIRECTION, orientation to left or right side of body},
{4,INSTRUMENT, the hand used as instrument of production},
{5,SOCIAL, control},
{6,DISCOURSE, plan or influence},
{7,ABSTRACT, symbol of promise},
{8,MANNER, manner of doing something},
{9,ABSTRACT, handwriting},
{10,TEMPORAL, applause},
{11,TEMPORAL, assistance},
{12,ROLE, laborer},

{13,ROLE, skillful person},
{14,SENTIENT, source of information},
{15,ARTIFACT, indicator, as on a clock},
{16,ABSTRACT, available}.

4. Noun Disambiguation

The present section will describe the environments in which the algorithm applies, and the effects it has. Fixed and frequent phrases, syntactic tests and commonsense reasoning will be considered in turn. In each description, the name of the rule will be listed in boldface. This may be used as a guide to the Section 9, which describes the encoding of the rules.

4.1. Fixed and Frequent Phrases

Using known computational methods, it is possible to identify sequences of words which are always used with the same sense of an ambiguous noun (Black [2]). Once encoded, they can be compared with an incoming text using fast methods. In order to be worth encoding, such phrases should be frequent. Fixed phrases vary in length from long (*on the one hand, on the other hand, in good company, in good hands*) to short (*iron hand, office space, office hours, by hand, to hand, in hand, in office*). Many of the fixed phrases and verb-noun combinations sought in these tests are idiomatic expressions, that is, phrases in which the meaning is not compositional.[2] The reason they are distinguished in the algorithm is that they employ different computational methods.

4.2. Syntactic Tests

Syntactic tests inspect the a simplified version of the parse to disambiguate. In "Fred ruled the famous detective agency with a free hand", *rule* disambiguates *hand* in oblique position, and the simplified parse indicates that the two words have the required syntactic relationship. There is a fixed set of types of tests, to be described here, and for each ambiguous noun, the test consists of a PROLOG rule (predicate name in boldface) associated with the noun which says, in effect, if a particular syntactic context exists, select a certain sense or sublist of senses.

The nature of the determiner in a noun phrase is powerful in disambiguating. The presence or absence of a definite article, the choice of definite article, and the presence or absence of personal pronouns are all clues to disambiguation which may partially or fully reduce the ambiguity of a noun. This relates to the fact that the generic use of a noun is often the clue which disambiguates it. No semantic generalizations have emerged here, simply the fact that these syntactic structures select senses. In some cases the absence of a determiner disambiguates (**nodet**). The phrase *aspire to high office* must use the electoral sense (3) of *office.* When the determiner is present, but is not a definite article, ambiguity can be reduced or eliminated (**nodef**). *hands* with an indefinite article has only seven senses (1,2,5,8,11,12,13). The definite article can disambiguate (**defart**). *people* with the definite article has three of six senses. The indefinite article disambiguates (**indefart**). *office* with the indefinite article reduces ambiguity from five senses to three (senses 1,2 or 3). An even more detailed rule is the one which distinguishes *the office of the Senator* from *the office of Senator.* The former is ambiguous between the INSTITUTION and PLACE senses, while the latter can only have the electoral sense (3). The difference is the definite article used with the ROLE term. This difference was frequent in the concordances (n=203). When the determiner is a personal pronoun, certain senses are selected (**perspron**). The phrase "her hand", has five readings for *hand* (senses 1,2,6,9,12). Add the preposition *in* and the ambiguity is reduced again so that the only possibilities are the bodily part sense (1,2) or the power sense (5). Similarly, "with her hands" can only have the bodily part sense, the INSTRUMENT sense (4) or the ROLE sense (12). Quantifiers can disambiguate (**quant**). When *idea* occurs with the quantifiers *some, any* and *no*, the sense which means "vague notion" must be selected.

When nouns take complements, the presence of the complement can disambiguate (**comp**). *idea* with a complement can have two of four senses. Much of the commonsense reasoning concerns prepositions and verbs. However, word-specific tests for specific prepositions or verbs are also frequent. Being the object of certain prepositions is enough to disambiguate some ambiguous nouns (**prep**). The preposition *by* selects the INSTRUMENT sense (4) of *hand* as in "Fred made the chair by hand". The preposition *with* in combination with a personal pronoun reduces the ambiguity of *hand* from sixteen to four senses (1,2,4,12). Ambiguity can be resolved when the noun is in subject, object or oblique position relative to a particular verb (**verbsubj verbobj verbobliq**). With *hand*, the verb *tremble* selects the bodily part sense (1) in subject position. The verb *shake* selects its bodily part sense in object position.

If some specific noun disambiguates an ambiguous noun in a frequent pair, this fact will reflected in the list of fixed phrases for the ambiguous noun. If a noun disambiguates another through more complex commonsense reasoning, this fact will be discovered by the commonsense rules. Therefore, no syntactic tests for noun-noun disambiguation exist.

Figure 3 summarizes the effect of the syntactic rules with the noun *office*. After each rule has been tried, if the rule succeeds it either reduces the senselist to one member, resulting in an exit from the algorithm, or it reduces it to more than one member and the algorithm continues. If the rule fails, the senselist remains unchanged, and the algorithm continues. At the end, if the senselist has more than one member, the commonsense rules are tried. The rules in PROLOG form are listed in Section 9.

4.3. Commonsense Knowledge

The power of the commonsense knowledge representations in the KT system can be used to resolve difficult cases of ambiguity by determining ontological similarity or generic relationships. Of 2193 concordances, 1068 (49%) were disambiguated using commonsense knowledge. Ontological similarity is defined as the attachment or inheritance of the same ontological node by two nouns. For example, in *the senator or his office* the INSTITUTION sense (1) of *office* is selected because senator is SENTIENT and so is INSTITUTION. (This is not to claim that ontologically dissimilar nouns cannot be conjoined, only that in cases of ambiguity, ontological similarity is used to disambiguate.) Where both nouns are ambiguous, the sense of each which matches the other ontologically is chosen. If ontological tests fail, the deepest level of reasoning must be invoked, in which information about an ambiguous noun is found in the generic representation of another verb or noun. In "John patted the dog with his hand", the personal pronoun in combination with the preposition *with* reduces the ambiguity of *hand* from sixteen senses to two, the bodily part (1) and the laborer

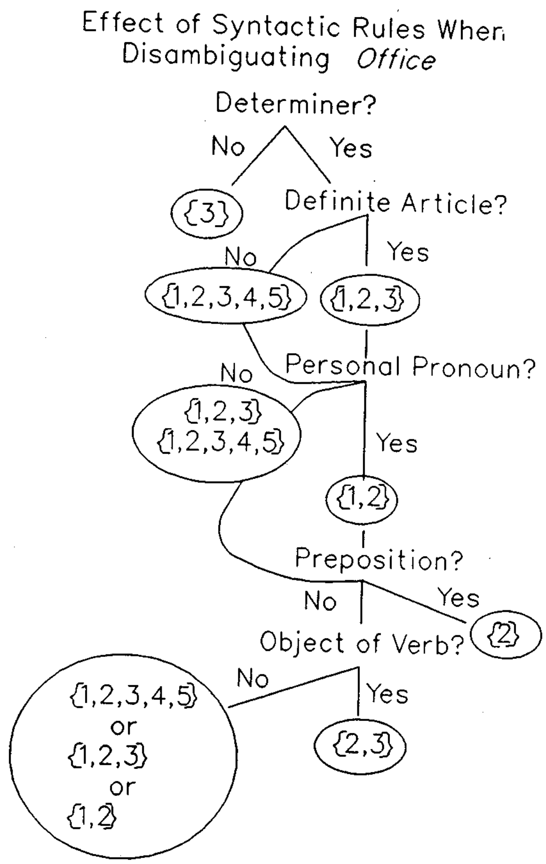

Figure 3

sense (12), as in "farm hands". In order to select the bodily part sense, generic knowledge of the typical instruments of the verb *pat* must be accessed.

When the ambiguous noun is in a conjunction with a another noun, ontological similarity disambiguates (**conj**), as in the above example of *the senator or his office*. In contrast, in *the office and furniture* only the PLACE sense (2) of *office* is possible because *furniture* is PHYSICAL, and PLACE is under PHYSICAL in the ontology. Where both nouns are ambiguous, the sense of each which matches the other ontologically is chosen, as in *the company and the school* which selects the INSTITUTION sense of both.

Frequently a noun immediately preceding an ambiguous noun disambiguates because the pair names a subordinate of the head noun (**prevnoun**). For example, *oil company*, *coal company* and *manufacturing company* are all ontologically subordinate to *company*. In another type, the ambiguity can be resolved by matching ontological attachments, as in *parent company* where both nouns are attached to SOCIAL. When a noun immediately follows the ambiguous noun, ontological matching is usually sufficient (**subseqnoun**), as in *company official* and *office worker*. The pairs share a SOCIAL attachment, so the INSTITUTIONAL senses (1) of *office* and *company* are chosen. In *store window* and *office door*, the pairs share a the PHYSICAL node, so the PLACE senses (2) of *store* and *office* are chosen.

An ambiguous noun can be disambiguated by a prepositional phrase which modifies it (**ppmod**). In the phrase "take a hand in government" the indefinite article reduces the ambiguity of *hand* from sixteen to six senses, and the head of the prepositional phrase modifier selects the power sense (5). In *the office in New York*, the fact that New York is a PLACE selects the PLACE sense (2) of *office*. When the ambiguous noun is inside a prepositional phrase which modifies another noun, which preposition is used is irrelevant (**nounpp**). Commonsense reasoning concerning the relationship between the modified noun and the ambiguous noun is enough to disambiguate. This was tested on the concordances with 100% success. An example is *a secretary in the office*, where matching SOCIAL only partially disambiguates from five senses to three, the INSTITUTION (1), PLACE (2) and ELECTORAL (3) senses. The rest of the disambiguation frequently results from commonsense reasoning about the verb, as in "The secretary in the office decided to buy more paper", "The secretary in the office sits near the door", and "The secretary in the office of President would be unusual". When the noun modified by the PP containing the ambiguous noun is a nominalized verb, the selectional restrictions of the underlying verb are used to disambiguate. For example, the verb *establish* selects SOCIAL objects, so that the phrase *establishment of the office* can only have the INSTITUTIONAL (1) or ELECTORAL (3) senses for *office*.

Commonsense reasoning with verbs is very important, and accounts for 24% of the commonsense disambiguations. The reasoning depends upon the relationship between the ambiguous noun and the verb. Disambiguations were evenly distributed across subject, object or oblique argument positions for the ambiguous noun (**verbsubj__cs, verbobj__cs, verbobliq__cs**). Verb selectional restrictions are very important in determining which sense of a noun is used in the verb's argument positions.

When the ambiguous noun is in subject position, and the verb selects for a certain ontological class in that argument, the verb disambiguates. In subject position, frequently the SENTIENT sense of nouns is chosen corresponding to the agentive role SENTIENTs play in actions. For example, in "The office solved the problem", because *solve* selects for a SENTIENT in subject position, the INSTITUTIONAL sense (1) of *office* is selected. In contrast, with "The office looks nice", the verb *look* with no object selects a PHYSICAL OBJECT in subject position, so the PLACE sense (2) of *office* is selected. In object position, selectional restrictions of the main verb disambiguate as well. MOTION verbs select for PLACE in object position, so in the phrases *move the office* and *locate the office* the PLACE sense (2) of *office* is chosen. Verbs select for certain ontological classes in oblique argument position as well. Examples are *send to the office* where *send* selects for PHYSICAL in the oblique argument, as opposed to *speak to the office* where speak selects for a SENTIENT in the oblique argument. In the first case the PLACE sense (2) of *office* is selected, and in the second case the INSTITUTION sense (1). Similarly, in *build the ship with his hands, build* selects for an INSTRUMENT in oblique argument (sense 4). MOTION verbs select for PLACE in oblique position, so *go across the office* and *arrive at the office* select the PLACE sense (2) of *office*.

Adjectives, like verbs, select for certain types of arguments (adj). The phrase "big office" selects the PLACE sense (2) for office because *big* selects physical objects as arguments. The adjectives *regional* and *local* are ambiguous, and can select either the INSTITUTION sense (1) of *office* selecting a SOCIAL argument, or the PLACE sense (2) which locates physical objects. These choices are reflected in the difference between "The district office is in charge of immigrants" and "The district office has four cubicles." When the adjective is derived from a verb as in "elective office" and "appointed office", the relevant verb test is used.

5. Verb Sense Disambiguation

The disambiguation of verbs can be accomplished using the same local combined algorithm as for nouns. To test the usefulness of the algorithm for verbs, four highly ambiguous verbs were studied---*move, use, support* and *work*. Generic representations and ontological attachments for each were encoded, and these were used to test the disambiguation algorithm against a total of 1789 (1225 transitive, 549 intransitive) instances in the Hansard corpus supplied by Ezra Black. With the exception of 99 instances, none of which contained full sentences, the algorithm succeeded in 99% of the cases.

As the disambiguation rules for verbs look very similar to those for nouns, this section will only describe and illustrate the types of information used in the rules. Intransitive *work* and transitive *move* will be used as detailed examples. Their senses are summarized in Table II by synonymous phrases rather than the full formal implicational representation which is actually used in KT. Alongside these are selectional restrictions included in the generic representations of the verbs.

Table II. Representation of Two Verbs Summarized

Verb	#	Paraphrase of Sense	Selection Restrictions
work	1	to labor	human(subject)
	2	to be employed	human(subject)
	3	to perform its function	social(subject)
	4	to operate effectively	machine(subject)
	5	to influence	human(subject)
	6	to be kneaded	dough(subject)
	7	to move with difficulty	selfmoving(subject)
	8	to change into a specified condition	-
move	1	change the position of	selfmoving(subject), physical(object)
	2	set in motion	selfmoving(subject), physical(object)
	3	persuade to do something	sentient(subject), sentient(object)
	4	stir the emotions of	sentient(subject), human(object)
	5	formally propose	sentient(subject), propositional(object)

Although there is some overlap, the content of the syntactic rules for verbs differs from the rules for nouns. The syntactic rules for verbs inspect the verb phrase specifier (particularly the auxiliary), adverbs and complement clauses. Commonsense rules for verbs and nouns are very similar, but they apply in opposite directions. They appeal to ontological and generic knowledge. As applied to verbs, the rules use the selectional restrictions of verbs, as shown in Table II, to inspect the ontological attachments of the verb's arguments. For example, the sense of *move* (5) which has to do with proposing motions in legislative bodies requires an object which is PROPOSITIONAL. The ontological attachments of surrounding nouns disambiguate verbs, rather than the ontological attachments of verbs themselves. In general, different senses of a verb all have the same or very similar ontological attachments, as can be observed in the example of *work* in Table II. What differs between senses are the selectional restrictions. The arguments of the verb which are most important in disambiguating are those which occur inside the verb phrase. For transitive verbs, the object of the verb disambiguates in 60% of the cases (n=737). A number of other elements contribute as well, such as prepositional phrases, adverbs and complements. For transitive verbs, the most effective disambiguators are adverbs (31%, n=171) and prepositional phrases (29%, n=161). Surprisingly, for neither transitive or intransitive verbs, was the subject particularly important (2% and 17% respectively).

5.1. Frequent Phrases in Verb Disambiguation

Fixed and frequent phrases which fully disambiguate are found in the data. For example, particles with the verb select senses, as in *work on, work against, move ahead.*

5.2. Syntactic Tests in Verb Disambiguation

There are fewer syntactic tests for verb senses than there are for nouns. The most clear cut one is the presence of a reflexive object (**reflex**), as in *move himself* which selects the first sense of transitive *move*. Elements of the specifier of the VP disambiguate, such as aspectuals (**aspectual, modal**). For intransitive *work* the auxiliaries *get to, start, get down to* select the first sense, the modals *can, want, must* select the second sense, and the modals *might, could, would, should* select the fourth sense. Particular adverbs select senses (**adv**). For intransitive *work* the adverbs *closely, satisfactorily, diligently* select the first sense, *part time, full time* select the second sense, *automatically* selects the third sense, and *effectively, well, successfully* select the fourth sense. The presence of a complement of the verb can disambiguate (**comp**). With intransitive *work* the presence of a complement clause narrows the ambiguity to senses one or two only, as in *work to expand the economy* (sense one) and *work to support one's family* (sense two). Particular prepositions select senses (**vprep**). *work together* can only have the first sense of transitive *work*.

5.3. Commonsense in Verb Disambiguation

Commonsense reasoning for verb disambiguation deals most often with selectional restrictions, although generic information is used as well. Most striking are selectional restrictions for objects of the verb. Transitive *move* has five senses, and for each sense, a particular ontological attachment of its object is selected, as indicated in Table II. Thus a PROPOSITIONAL object decisively disambiguates, as in "John moved the amendment". A SENTIENT or PHYSICAL object reduces the number of senses from five to two, as in "John moved the girl to tears" or "John moved his friend to take a stand". If *move* has a quantifier as object, sense one is selected, as in "We moved four today". Another important selectional restriction concerns for oblique arguments. For example, with intransitive *work*, sense two selects for a relational phrase with the prepositions *for, in, at* as in *work for the company*, where the oblique argument is an INSTITUTION. The arguments *to the advantage, to the disadvantage* select sense three. Subject selectional restrictions are less important, but do disambiguate. For example, if the subject of intransitive *work* is a SOCIAL ROLE such as *employee, secretary, mother*, sense two is selected, if ARTIFACT, sense three is selected, and if INSTITUTION, such as *party, system, democracy* sense four is selected. Similarly, a PHYSICAL subject selects the physical sense of *support*, as in "The sandbags support the bank".

Just as with nouns, verb disambiguation rules are applied to the most deeply embedded S first and upwards through the parse. This is how phrases such as *the will to work* and *ready to work* are disambiguated. On the embedded level in the these cases, there is no reduction of ambiguity. On the level of the next higher S, the noun or adjective disambiguates the embedded verb.

6. Interaction of Ambiguous Verb and Noun

How will this algorithm fare with sentences containing more than one ambiguous element, such as a verb and object which are both ambiguous? Clearly there will be cases it cannot handle. On the other hand, it will be able to handle many such cases, as will be illustrated here. Consider the pair of sentences below:

The sandbags supported the bank.
The investors supported the bank.

Assume that *bank* has two readings, one, "edge of a river" which is a NATURAL PLACE and two, "financial house" which is an INSTITUTION. As we saw above, the verb *support* in the sense which means "hold up physically" requires a PHYSICAL subject, and the other senses require sentients, so the verb is disambiguated by the subject in "The sandbags supported the bank". In the second case, there are seven possible senses of *support* which have SENTIENT subjects, so the subject only reduces the ambiguity from nine to seven. Others require certain object types, for example, the senses which mean "support an idea or cause" and "propose to a legislature" both require propositional objects. Eliminating senses through object selectional restrictions, the only remaining possibilities are senses which mean "supply funds for" and "assist or further" (as "support the party"). Both can have INSTITUTIONs as objects. In order to further disambiguate, the object rule which inspects generic information in the representation must be used. It will discover that both the sense of *support* which means "supply funds for" and the sense of *bank* which means "financial house" have to do with money, and in this way disambiguate both words. What about the case where the sentence subject can attach to the ontology in two places? Consider

John supported the bank.

John can attach either as a HUMAN, in which case he is PHYSICAL and can be viewed as holding up the bank bodily, or he can attach as a PERSON, in which case he is SENTIENT and can be viewed as supporting the bank financially or furthering the cause of the bank. In the latter case, after considering all selectional restrictions, the sentence remains ambiguous. In the former case, the question of choosing between

PHYSICAL and SENTIENT John must be resolved by the wider context of preceding or following sentences or by knowing that a person is not strong enough to physically hold up a bank.

In summary, the disambiguation of nouns and verbs uses information from several sources in the KT system:

1. A lexicon of fixed phrases
2. A set of word-specific rules for each word which relate syntactic structure to word senses
3. The output of the parse of a sentence[3]
4. The ontology
5. Generic information about nouns and verbs, such as selectional restrictions

7. Syntactic and Lexical Ambiguity

The KT system has rules for attaching post-verbal prepositional phrase modifiers to the sentence, the verb phrase or the object noun phrase (Dahlgren and McDowell [5]). These rules apply before the disambiguation rules. The algorithm uses syntactic, ontological and generic information. Consider the sentence "Fred spoke of that in his office today". No global prepositional phrase attachment rules apply, and so the *in*-rule applies. It checks whether the object of the preposition is a PLACE. The rule scans all the senses of *office* to see whether any is a PLACE ontologically. For *office* there is such a sense, and so the rule disambiguates *office* and selects the PLACE sense (2). It also attaches the prepositional phrase to the sentence. Thus the disambiguation rules never apply to this instance of *office*. It comes into those rules with the sense already selected.

8. Intersentential Reasoning

Obviously not all disambiguation can take place within a single sentence. Sometimes the ambiguity cannot be resolved within a single sentence, though it could in 99% of the test concordances. "John argues with his hands" is ambiguous between a reading An example of a sentential context which fails to disambiguate is in which hands are John's body parts, and hands are workers who work for John. A following sentence will usually disambiguate as in the following texts:

Jones argues with his hands. He gestures constantly.
Jones argues with his hands. He listens to their complaints.

Three other senses of *hand* are the promise sense (7), the applause sense (10) and the assistance sense (11), as illustrated in the following sentences:

She gave him her hand in marriage.
The audience gave him a hand.
She gave him her hand to help him into the boat.

The disambiguation algorithm can use other elements of the above sentences to disambiguate, specifically the PP *in marriage*, the subject noun *audience* and the embedded sentence "to help him". There are cases, however, when the indefinite article is used and other elements do not help, leaving all three senses as possibilities.

She gave him a hand.
    She promised to repay the loan promptly.
She gave him a hand.
    His singing was wonderful.
She gave him a hand.
    He was drowning.

Here we need to look at other sentences to disambiguate. For example, in the first one, the promise sense has a typical function of promising in the generic information. The occurrence of *promise* in the second sentence could be used to reduce the senselist from three to one. In the next case, generic information associated with the applause sense indicates that it is typically located at performances, and the generic information for singing says that it typically takes place at performances, and thus the noun *hand* can be disambiguated.

9. Disambiguation Rules

The algorithm is programming language-independent. Its implementation in VM/PROLOG [15] is described here in terms of a subset of the rule types outlined in Section 4 and 5. The top-level disambiguation rule gets the senselist (Senselistin) for the ambiguous noun (Word) and finds the index of the appropriate sense (Senselistout) by invoking fixed-phrases, syntactic and commonsense tests. A parse in the form of a labelled bracketing of the entire sentence the ambiguous noun occurs in (Parse) is passed to this rule.

```
disambig(Word,String,Parse,Index) <-
   senselist(Word,Senselistin)
   & fixedphrase(Word,String,Senselistin,Senselist1).
   & get__struc(Word,Parse,Struc,Senselist1)
   & syntactic__test(Word,Struc,Senselist1,Senselistout)
   & commonsense(Word,Struc,Senselistout,Index).
```

The fixed phrases logic attempts to match the ambiguous word and surrounding words to a list of disambiguated or partially disambiguated fixed phrases involving the ambiguous word. For example, the following ground clause says that *Printing Office* selects sense 5 of *office*.

```
fixedphrase(office,{Printing,Office},{1,2,3,4,5},5).
```

Since the disambiguation algorithm is local, only one S-node is considered at a time (no higher or lower S-nodes). This makes possible a simplification of the parse. The procedure **get__struc** takes Parse and simplifies it into a fixed-position list of the verb and its arguments in the sentence, followed by a fixed-position sublist. The sublist differs for ambiguous nouns and verbs. For ambiguous nouns, the sublist has elements of the NP in which the ambiguous noun occurs, such as determiner, quantifier, and adjective, that is, the specifier of the NP. For verbs the sublist has the VP specifier and structures such as reflexive and complement. The elements returned by **get__struc** in the variable Struc are employed by syntactic and CK tests. Both their presence and their content are significant information in the algorithm. In what follows, Struc is not given in detail. Irrelevant positions are indicated by (...).

Turning to the syntactic tests, there is a fixed set of types of them. The logic of the syntactic test algorithm assumes that the rules for an a given ambiguous word draw upon this fixed set of types. Each type of rule is attempted for each ambiguous word. If the rule type is irrelevant to that word, it trivially succeeds. If the senselist has already been reduced to one member (a solution), the test trivially succeeds. Otherwise, the test is relevant, and the rule type is attempted. Because PROLOG can represent rules as data, the content of the rule can differ for each noun. The test is carried out by interpreting PROLOG clauses listed in the lexical entry for the word, and applying them in the same way regardless of the word. This makes the programming very simple. The order of the types is fixed, and for any given word a variable number of the types of rules is actually applied.

The top-level syntactic test rules for nouns follow. Notice that the names of the predicates are the same as those listed next to the rule type descriptions in Section 4. The first rule is for the case where fixed phrases fully disambiguate, and the disambiguated senselist is simply passed through the rule.

```
syntactic_test(Noun,Struc,Senselistin,Senselistin) <-
          atomic(Senselistin).
syntactic_test(Noun,Struc,Senselistin,Senselistout) <-
        nodet(Noun,Struc,Senselistin,Senselist1)
     & nodefsing(Noun,Struc,Senselist1,Senselist2)
     & nodefplu(Noun,Struc,Senselist2,Senselist3)
     & defart(Noun,Struc,Senselist3,Senselist4)
     & indefart(Noun,Struc,Senselist4,Senselist5)
     & perspron(Noun,Struc,Senselist5,Senselist6)
     & quant(Noun,Struc,Senselist6,Senselist7)
     & propappos(Noun,Struc,Senselist7,Senselist8)
     & prep(Noun,Struc,Senselist8,Senselist9)
     & verbsubj(Noun,Struc,Senselist9,Senselist10)
     & verbobj(Noun,Struc,Senselist10,Senselist11)
     & verbobliq(Noun,Struc,Senselist11,Senselist12)
     & bepp(Noun,Struc,Senselist12,Senselist13)
     & comp(Noun,Struc,Senselist13,Senselistout).
```

A subset of the word-particular syntactic rules for *office* appears below. The **nodet** rule succeeds if there is no determiner and returns sense 3, the electoral sense. **defart** succeeds if there is a definite article and reduces the senselist to three senses. **perspron** reduces it to two. The **prep** rules look for specific prepositions, and the *verb* rules for specific verbs and argument positions. Notice that some rules have bodies, and others are ground clauses.

```
nodet(office,Struc,Senselist,3) <- nodet(Struc).
defart(office,Struc,Senselist3,{1,2,3}) <- defart(office,Struc).
indefart(office,Struc,Senselist3,{1,2,3}) <- indefart(office,Struc).
perspron(office,Struc,Senselist,{1,2}) <- perspron(office,Struc).
prep(office,{...at...},Senselist,2).
prep(office,{...against...},Senselist,1).
prep(office,{...through...},Senselist,{1,2}).
prep(office,{...to...},Senselist,{1,2}).
verbobj(office ,{...occupy...},Senselist,{2,3}).
```

```
verbobliq(office ,{...elect...},Senselist,3).
verbobliq(office ,{...appoint...},Senselist,3).
```

The top-level syntactic test rules for verbs are listed below. The names of the predicates are the same as those used in the descriptions in Section 5.

```
syntactic__test(Verb,Struc,Senselistin,Senselistin) <-
        atomic(Senselistin).
syntactic__test(Verb,Struc,Senselistin,Senselistout) <-
      reflex(Verb,Struc,Senselistin,Senselist1)
    & aspectual(Verb,Struc,Senselist1,Senselist2)
    & modal(Verb,Struc,Senselist2,Senselist3)
    & adv(Verb,Struc,Senselist3,Senselist4)
    & comp(Verb,Struc,Senselist4,Senselist5)
    & vprep(Verb,Struc,Senselist5,Senselistout).
```

Some of the word-specific rules for intransitive *work* are given below. As with nouns, some word-specific verb rules are ground clauses, and others have bodies. The **aspectual** rule succeeds if the auxiliary is *get* or *start* In either case the senselist is returned as 1. The **modal** rule succeeds if there is a modal, and it is *can* or *might*. The cases select different senses. The other rules work similarly. Irrelevant elements of Struc are elided with (...)

```
aspectual(work,{...get...},Senselist,1).
aspectual(work,{...start...},Senselist,1).
modal(work,{...can...},Senselist,2).
modal(work,{..might...},Senselist,4).
adv(work,{...diligently...",Senselist,1).
adv(work,{...part,time...},Senselist,2).
adv(work,{...automatically...},Senselist,3).
prep(work,{..to...},Senselist,{1,2}).
```

The commonsense rules operate similarly to the syntactic rules in that a fixed set of types of rules is attempted for every ambiguous noun, if the senselist has not yet been reduced to one member when the commonsense test is invoked. Below is the top-level commonsense rule, which applies to both nouns and verbs. Those tests irrelevant to a particular word trivially succeed.

```
commonsense(Word,Struc,Senselistout,Senselistout) <-
   atomic(Senselistout).
commonsense(Word,Struc,Senselistin,Senselistout) <-
     conj(Word,Struc,Senselistin,Senselist1)
   & prevnoun(Word,Struc,Senselist2,Senselist3)
   & subseqnoun(Word,Struc,Senselist3,Senselist4)
   & ppmod(Word,Struc,Senselist4,Senselist5)
   & nounpp(Word,Struc,Senselist5,Senselist6)
   & verbsubj__cs(Word,Struc,Senselist6,Senselist7)
   & verbobj__cs(Word,Struc,Senselist7,Senselist8)
   & verbobliq(Word,Struc,Senselist8,Senselist9)
   & subj__cs(Word,Struc,Senselist9,Senselist10)
   & obj__cs(Word,Struc,Senselist10,Senselistout).
```

The commonsense rules are more general than are the syntactic rules, in that rather than accessing lexical rules specific to each word, they compare ontological and generic information in ways which apply to any noun or verb. For example, in the conjunction rule **conj__cs** the ontological attachment of an ambiguous word (Word) is compared with the ontological attachment of a word it is conjoined with, and if they match for some sense of the ambiguous word, that sense is chosen. This reflects the fact that sentences like "The office and the office door solved the problem", which would conjoin the INSTITUTION and PLACE senses of *office* are semantically anomalous. Rules which look at generic information, such as the second **ppmod** rule, access elements of the commonsense knowledge representations of typical and inherent features of objects named by nouns and actions named by verbs. **adjselect** and **verbselect** extract selectional restrictions for adjectives and verbs in the form of nodes in the ontology.

```
conj__cs(Word,{...Word...Conjunct...},S1,S2) <-
        ontattach(Conjunct,Node)
        & ontattach(Word,Node,S1,S2).
prevnoun(Word,{...Prevnoun...Word...},S1,S2) <-
        isa(Prevnoun,Word).
prevnoun(Word,{...Prevnoun...Word...},S1,S2) <-
        ontattach(Prevnoun,Node)
        & ontattach(Word,Node,S1,S2).
subseqnoun(Word,{...Word...Subseqnoun...},S1,S2) <-
        ontattach(Subseqnoun,Node)
        & ontattach(Word,Node,S1,S2).
adj(Word,{...Adj,Word...},S1,S2) <-
        adjselect(Adj,Node)
        & ontattach(Word,Node,S1,S2).
ppmod(Word,{...Word...Prep,Noun...},S1,S2) <-
        ontattach(Noun,Node)
        & ontattach(Word,Node,S1,S2).
ppmod(Word,{...Word...Prep,Noun...},S1,S2) <-
        generic__relation(Noun,Word).
nounpp(Word,{...Prep,Noun...Word...},S1,S2) <-
        ontattach(Noun,Node)
        & ontattach(Word,Node,S1,S2).
nounpp(Word,{...Prep,Noun...Word...},S1,S2) <-
        generic__relation(Noun,Word).
verbsubj__cs(Word,{...Word,Verb...},S1,S2) <-
        verbselect(Verb,subj,Node)
        & ontattach(Word,Node,S1,S2).
verbobj__cs(Word,{...Verb,Word...},S1,S2) <-
        verbselect(Verb,obj,Node)
        & ontattach(Word,Node,S1,S2).
verbobliq__cs(Word,{...Verb...Word...},S1,S2) <-
        verbselect(Verb,obliq,Node)
        & ontattach(Word,Node,S1,S2).
```

10. Efficiency and Timing

The efficiency of the algorithm depends upon a number of factors. Fixed phrases account for 70% of lexical ambiguities, and finding them is a simple matter of a table lookup. If these fail, then for each term the 14 syntactic rules, and the 10 commonsense rules are tried in turn until they succeed. A maximum of 22 rules is applied for any ambiguous word. Rules try to unify with a word-specific rule, but not all words have all rule types. For example, *office* has only eleven rules word-specific syntactic rules.

The fixed phrases portion of the algorithm is linear because it matches the input string to a database of fixed phrases. The length of time it takes is O(n) where n is the number of fixed phrases. The syntactic tests are also linear. Syntactic complexity of the sentence is simplified by applying to only one S-node at a time, and by converting the parse into a fixed-position list in **get__struc**. The logic then employs PROLOG unification in applying syntactic tests. PROLOG unification is at best linear (Paterson and Wegman [17]). The commonsense reasoning has two levels of efficiency. Generically-based tests search a database of generic information which consists of PROLOG ground clauses, which again can be accomplished in linear time relative to the number of entries in the generic database. Ontologically-based tests are non-linear. The KT ontology is an acyclic, directed, connected graph which can be searched in O(log n) time. Another source of non-linearity would be long-distance dependencies. Since the algorithm applies locally, to only one S-node at a time, the above description of efficiency does not include the case where an embedded sentence contains a trace, and cannot be disambiguated until the NP which belongs in the trace has been located in a higher S. Such long-distance dependencies can result in non-linear processing time.

The relative timing of the elements of the algorithm have been tested in relation to the noun *office*. The algorithm was coded in VM/PROLOG [15] using the output of the parser [16] as input. Ten sample sentences were chosen to test timing, as shown in Table III. For purposes of the test, it is assumed that *office* is the only ambiguous word in the sentences, and that no sentence is syntactically ambiguous. There were 11 fixed phrases in the VM/PROLOG workspace. The first sentence is disambiguated by fixed phrases, the second through fifth by syntactic tests, and the six through tenth by commonsense reasoning. In order to factor out the effect trying fixed phrases first, then all syntactic tests, then commonsense reasoning, the timing for each aspect of the algorithm for each sentence was measured, as shown in Table III. The program was run on an IBM 3081 with no other users. The algorithm is very fast, and takes an insignificant amount of time in comparison with parsing. As predicted, commonsense reasoning takes the most time. The differences among sentences disambiguated by commonsense reasoning is due to the fact that it is faster to look up generic knowledge than it is to search the ontology (which is a lattice). This test was based upon simple sentences, because the algorithm works to disambiguate each sentence as it is parsed, including embedded sentences.

The timing test shows that speed could be improved by an implementation using parallel processing. Since fixed phrases, syntactic tests and generic commonsense reasoning are linear, all three processes could be started at once. The reduced sense lists from the three disambiguation methods could then be intersected. Parallel processing is not indicated for ontologically-based commonsense reasoning, as in Sentence 10, because as described above it is nonlinear. Therefore a version of the algorithm using parallel processing should start up only the generically-based commonsense rules in parallel with the fixed phrases and syntactic tests, and if these do not fully disambiguate, then try ontologically-based commonsense reasoning. Such an implementation would bear no relation to cognitive modelling.

Table III. Timing of Test Sentences in Milliseconds

Test Sentence	*Fixed Phrases*	*Syntax*	*CK*	*Total Disambiguation*
1. John called the patent office.	2.1	-	-	2.2
2. John walked through his office.	2.5	2.7	-	2.7
3. John ran for office.	2.5	2.5	-	2.7
4. John arranged for it through his office.	2.8	3.0	-	3.0
5. John ran the office.	2.0	2.1	-	2.2
6.The office desk is pretty.	2.4	2.5	3.4	3.4
7. John saw the desk in the office.	2.7	2.8	4.0	4.3
8. John moved his office to Chicago.	2.6	2.9	5.6	6.0
9. John called his office.	2.1	2.3	4.4	4.7
10. The office clerk called John.	2.5	2.4	6.1	6.5

11. Problems for the Method

The disambiguation algorithm is a preference strategy which sometimes fails, as in the text

Jones reached the office he had been
looking for. He served with distinction.

The present disambiguation rules will use the verb *look for* in the embedded sentence to guess the PLACE sense for *office*, which is incorrect. The way to solve this problem would be to keep a trace of the reasoning rules used to disambiguate for possible later revision. The verb *serve* in the second sentence selects the INSTITUTION sense of *office* in subject position, and the electoral sense in object position. This fact could be used to overturn the original guess of the PLACE sense and replace it with the electoral sense.

Certain nouns are ambiguous within the same ontological attachment and the disambiguation occurs situationally. The commonsense reasoning involved may go way beyond lexical level representations. An example is the noun *mat* which has four senses, one for floor mats, under vase mats, exercise mats and matted objects (such as a mat of vines):

```
mat({1,artifact,{size(2x3), shape(rectangular), material(woven hemp)},
{function(protect a floor)}},
{2,artifact,{size(5" diameter), shape(circular), material(lace)}, {function(protect a
table)}},
{3,artifact, {size(3' by 8'), shape(rectangular), material(foam rubber)},
{function(cushion a person)}},
{4,natural, {form(matted)},{}}).
```

The first three are all artifacts, and all quite similar, except that generic information differs in values. Some disambiguation rules apply, but frequently the senses cannot be distinguished. The only solution would be more sophisticated commonsense reasoning, which would employ naive physics to reason that the agent in a paragraph is exercising, and needs the protection of the mat. Such reasoning is possible, and would connect this work with other work in commonsense reasoning (Hobbs et al [8]).

12. Conclusion

A word sense disambiguation algorithm was developed which combines information from fixed phrases, syntactic context and commonsense knowledge to disambiguate locally. It was successfully tested on a corpus of text. Almost all cases could be disambiguated using intrasentential information. The next step is to expand the method to combinations of highly ambiguous nouns and verbs.

Acknowledgements

The author gratefully acknowledges the useful comments of two anonymous reviewers, Bijan Arbab, Ezra Black, Carol Lord, Susan Mordechay, Joyce McDowell, and Eric Wehrli.

Notes

1. This paper uses italics for lexical items, capitals for ontological nodes. Long phrases and sentences are quoted rather than italicized.

2. I am indebted to Eric Wehrli for pointing this out.

3. The parser is NL/PROLOG by Stabler and Tarnawsky [16], [17]. It produces an X-bar parse tree and enforces bounding nodes.

References

[1] Milne, R. 1986. "Resolving Lexical Ambiguity in a Deterministic Parser" *Computational Linguistics* 12:1-12.

[2] Black, Ezra W. 1986. "Towards Computational Discrimination of English Word Senses". CUNY dissertation.

[3] Gross, M. 1985. "Projecting the lexicon-grammar on Texts". Paper presented at the Roman Jakobson Conference, NYU, 1985.

[4] Dahlgren, K. and J.P. McDowell. [4]. "Kind Types in Knowledge Representation". *Proceedings COLING86* 216-221.

[5] Dahlgren, K. and J.P. McDowell. [5]. "Using Commonsense Knowledge to Disambiguate Prepositional Phrase Modifiers". *Proceedings AAAI86*

[6] Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M. and P. Boyes-Braem. "Basic Objects in Natural Categories". *Cognitive Psychology* 8:382-439.

[7] Dahlgren, K. [3]. "The Cognitive Structure of Social Categories" *Cognitive Science* 9:379-398.

[8] Hobbs, J.R., Croft, W., Davies, T., Edwards, D. and K. Laws. 1986. "Commonsense Metaphysics and Lexical Semantics" *Proceedings ACL86* 231-240.

[9] Chomsky, Noam. 1965. *Aspects of the Theory of Syntax.*

[10] Ashcraft, Mark H. 1976. "Property Norms for Typical and Atypical Items from 17 Categories: A Description and Discussion" *Memory and Cognition* 6:227-232.

[11] Fehr, Beverly and James A. Russell. 1984. "Concept of Emotion Viewed From a Prototype Perspective". *Journal of Experimental Psychology: General* 113.3:464-486.

[12] Rifkin, Anthony. 1985. "Evidence for a Basic Level in Event Taxonomies" *Memory and Cognition* 13:538-556.

[13] Dahlgren, K. 1987. Naive Semantics for Natural Language Understanding. To appear.

[14] Smith, E.E. and D.L. Medin. 1981. *Categories and Concepts* Cambridge, Mass: Harvard University Press.

[15] *VM*Programming in Logic (VM/PROLOG)/, Program Description and Operations Manual, IBM Corp.

[16] Stabler, E.P. Jr. and Tarnawsky, G.O. 1987. "NL/Prolog--A Prolog-Based atural Language Facility". Forthcoming, IBM Systems Journal.

[17] Paterson, M.S. and M.N. Wegman. 1978. "Linear Unification" *J. Computer and System Sciences* 16:158-167.