

Near-Synonymy and Lexical Choice

Philip Edmonds*
Sharp Laboratories of Europe Limited

Graeme Hirst†
University of Toronto

We develop a new computational model for representing the fine-grained meanings of near-synonyms and the differences between them. We also develop a lexical-choice process that can decide which of several near-synonyms is most appropriate in a particular situation. This research has direct applications in machine translation and text generation.

We first identify the problems of representing near-synonyms in a computational lexicon and show that no previous model adequately accounts for near-synonymy. We then propose a preliminary theory to account for near-synonymy, relying crucially on the notion of granularity of representation, in which the meaning of a word arises out of a context-dependent combination of a context-independent core meaning and a set of explicit differences to its near-synonyms. That is, near-synonyms cluster together.

We then develop a clustered model of lexical knowledge, derived from the conventional ontological model. The model cuts off the ontology at a coarse grain, thus avoiding an awkward proliferation of language-dependent concepts in the ontology, yet maintaining the advantages of efficient computation and reasoning. The model groups near-synonyms into subconceptual clusters that are linked to the ontology. A cluster differentiates near-synonyms in terms of fine-grained aspects of denotation, implication, expressed attitude, and style. The model is general enough to account for other types of variation, for instance, in collocational behavior.

An efficient, robust, and flexible fine-grained lexical-choice process is a consequence of a clustered model of lexical knowledge. To make it work, we formalize criteria for lexical choice as preferences to express certain concepts with varying indirectness, to express attitudes, and to establish certain styles. The lexical-choice process itself works on two tiers: between clusters and between near-synonyms of clusters. We describe our prototype implementation of the system, called I-Saurus.

1. Introduction

A word can express a myriad of implications, connotations, and attitudes in addition to its basic “dictionary” meaning. And a word often has near-synonyms that differ from it solely in these nuances of meaning. So, in order to find the right word to use in any particular situation—the one that precisely conveys the desired meaning and yet avoids unwanted implications—one must carefully consider the differences between all of the options. Choosing the right word can be difficult for people, let alone present-day computer systems.

For example, how can a machine translation (MT) system determine the best English word for the French *bévue* when there are so many possible similar but slightly

* Sharp Laboratories of Europe Limited, Oxford Science Park, Edmund Halley Road, Oxford OX4 4GB, England. E-mail: phil@sharp.co.uk.

† Department of Computer Science, University of Toronto, Ontario, Canada M5S 3G4. E-mail: gh@cs.toronto.edu.

different translations? The system could choose *error*, *mistake*, *blunder*, *slip*, *lapse*, *boner*, *faux pas*, *boo-boo*, and so on, but the most appropriate choice is a function of how *bévue* is used (in context) and of the difference in meaning between *bévue* and each of the English possibilities. Not only must the system determine the nuances that *bévue* conveys in the particular context in which it has been used, but it must also find the English word (or words) that most closely convey the same nuances in the context of the other words that it is choosing concurrently. An exact translation is probably impossible, for *bévue* is in all likelihood as different from each of its possible translations as they are from each other. That is, in general, every translation possibility will omit some nuance or express some other possibly unwanted nuance. Thus, faithful translation requires a sophisticated lexical-choice process that can determine which of the near-synonyms provided by one language for a word in another language is the closest or most appropriate in any particular situation. More generally, a truly articulate natural language generation (NLG) system also requires a sophisticated lexical-choice process. The system must be able to reason about the potential effects of every available option.

Consider, too, the possibility of a new type of thesaurus for a word processor that, instead of merely presenting the writer with a list of similar words, actually assists the writer by ranking the options according to their appropriateness in context and in meeting general preferences set by the writer. Such an *intelligent thesaurus* would greatly benefit many writers and would be a definite improvement over the simplistic thesauri in current word processors.

What is needed is a comprehensive computational model of fine-grained lexical knowledge. Yet although **synonymy** is one of the fundamental linguistic phenomena that influence the structure of the lexicon, it has been given far less attention in linguistics, psychology, lexicography, semantics, and computational linguistics than the equally fundamental and much-studied **polysemy**. Whatever the reasons—philosophy, practicality, or expedience—synonymy has often been thought of as a “non-problem”: either there are synonyms, but they are completely identical in meaning and hence easy to deal with, or there are no synonyms, in which case each word can be handled like any other. But our investigation of **near-synonymy** shows that it is just as complex a phenomenon as polysemy and that it inherently affects the structure of lexical knowledge.

The goal of our research has been to develop a computational model of lexical knowledge that can adequately account for near-synonymy and to deploy such a model in a computational process that could “choose the right word” in any situation of language production. Upon surveying current machine translation and natural language generation systems, we found none that performed this kind of genuine lexical choice. Although major advances have been made in knowledge-based models of the lexicon, present systems are concerned more with structural paraphrasing and a level of semantics allied to syntactic structure. None captures the fine-grained meanings of, and differences between, near-synonyms, nor the myriad of criteria involved in lexical choice. Indeed, the theories of lexical semantics upon which present-day systems are based don’t even account for indirect, fuzzy, or context-dependent meanings, let alone near-synonymy. And frustratingly, no one yet knows how to implement the theories that do more accurately predict the nature of word meaning (for instance, those in cognitive linguistics) in a computational system (see Hirst [1995]).

In this article, we present a new model of lexical knowledge that explicitly accounts for near-synonymy in a computationally implementable manner. The **clustered model of lexical knowledge** clusters each set of near-synonyms under a common, coarse-

grained meaning and provides a mechanism for representing finer-grained aspects of denotation, attitude, style, and usage that differentiate the near-synonyms in a cluster. We also present a robust, efficient, and flexible lexical-choice algorithm based on the approximate matching of lexical representations to input representations. The model and algorithm are implemented in a sentence-planning system called I-Saurus, and we give some examples of its operation.

2. Near-Synonymy

2.1 Absolute and Near-Synonymy

Absolute synonymy, if it exists at all, is quite rare. Absolute synonyms would be able to be substituted one for the other in any context in which their common sense is denoted with no change to truth value, communicative effect, or “meaning” (however “meaning” is defined). Philosophers such as Quine (1951) and Goodman (1952) argue that true synonymy is impossible, because it is impossible to define, and so, perhaps unintentionally, dismiss all other forms of synonymy. Even if absolute synonymy were possible, pragmatic and empirical arguments show that it would be very rare. Cruse (1986, page 270) says that “natural languages abhor absolute synonyms just as nature abhors a vacuum,” because the meanings of words are constantly changing. More formally, Clark (1992) employs her principle of contrast, that “every two forms contrast in meaning,” to show that language works to eliminate absolute synonyms. Either an absolute synonym would fall into disuse or it would take on a new nuance of meaning. At best, absolute synonymy is limited mostly to dialectal variation and technical terms (*underwear* (AmE) : *pants* (BrE); *groundhog* : *woodchuck*; *distichous* : *two-ranked*; *plesionym* : *near-synonym*), but even these words would change the style of an utterance when intersubstituted.

Usually, words that are close in meaning are **near-synonyms** (or **plesionyms**)¹—almost synonyms, but not quite; very similar, but not identical, in meaning; not fully intersubstitutable, but instead varying in their shades of denotation, connotation, implicature, emphasis, or register (DiMarco, Hirst, and Stede 1993).² Section 4 gives a more formal definition.

Indeed, near-synonyms are pervasive in language; examples are easy to find. *Lie*, *falsehood*, *untruth*, *fib*, and *misrepresentation*, for instance, are near-synonyms of one another. All denote a statement that does not conform to the truth, but they differ from one another in fine aspects of their denotation. A lie is a deliberate attempt to deceive that is a flat contradiction of the truth, whereas a misrepresentation may be more indirect, as by misplacement of emphasis, an untruth might be told merely out of ignorance, and a fib is deliberate but relatively trivial, possibly told to save one’s own or another’s face (Gove 1984). The words also differ stylistically; *fib* is an informal, childish term, whereas *falsehood* is quite formal, and *untruth* can be used euphemistically to avoid some of the derogatory implications of some of the other terms (Gove [1984]; compare Coleman and Kay’s [1981] rather different analysis). We will give many more examples in the discussion below.

¹ In some of our earlier papers, we followed Cruse (1986) in using the term *plesionym* for *near-synonym*, the prefix *plesio-* meaning ‘near’. Here, we opt for the more-transparent terminology. See Section 4 for discussion of Cruse’s nomenclature.

² We will not add here to the endless debate on the normative differentiation of the near-synonyms *near-synonym* and *synonym* (Egan 1942; Sparck Jones 1986; Cruse 1986; Church et al. 1994). It is sufficient for our purposes at this point to simply say that we will be looking at sets of words that are intuitively very similar in meaning but cannot be intersubstituted in most contexts without changing some semantic or pragmatic aspect of the message.

Error implies a straying from a proper course and suggests guilt as may lie in failure to take proper advantage of a guide. . . . **Mistake** implies misconception, misunderstanding, a wrong but not always blameworthy judgment, or inadvertence; it expresses less severe criticism than *error*. **Blunder** is harsher than *mistake* or *error*; it commonly implies ignorance or stupidity, sometimes blameworthiness. **Slip** carries a stronger implication of inadvertence or accident than *mistake*, and often, in addition, connotes triviality. **Lapse**, though sometimes used interchangeably with *slip*, stresses forgetfulness, weakness, or inattention more than accident; thus, one says a *lapse* of memory or a *slip* of the pen, but not vice versa. **Faux pas** is most frequently applied to a mistake in etiquette. **Bull, howler, and boner** are rather informal terms applicable to blunders that typically have an amusing aspect.

Figure 1

An entry (abridged) from *Webster's New Dictionary of Synonyms* (Gove 1984).

2.2 Lexical Resources for Near-Synonymy

It can be difficult even for native speakers of a language to command the differences between near-synonyms well enough to use them with invariable precision, or to articulate those differences even when they are known. Moreover, choosing the wrong word can convey an unwanted implication. Consequently, lexicographers have compiled many reference books (often styled as “dictionaries of synonyms”) that explicitly discriminate between members of near-synonym groups. Two examples that we will cite frequently are *Webster's New Dictionary of Synonyms* (Gove 1984), which discriminates among approximately 9,000 words in 1,800 near-synonym groups, and *Choose the Right Word* (Hayakawa 1994), which covers approximately 6,000 words in 1,000 groups. The nuances of meaning that these books adduce in their entries are generally much more subtle and fine-grained than those of standard dictionary definitions. Figure 1 shows a typical entry from *Webster's New Dictionary of Synonyms*, which we will use as a running example. Similar reference works include Bailly (1970), Bénac (1956), Fernald (1947), Fujiwara, Isogai, and Muroyama (1985), Room (1985), and Urdang (1992), and usage notes in dictionaries often serve a similar purpose. Throughout this article, examples that we give of near-synonyms and their differences are taken from these references.

The concept of **difference** is central to any discussion of near-synonyms, for if two putative absolute synonyms aren't actually identical, then there must be something that makes them different. For Saussure (1916, page 114), difference is fundamental to the creation and demarcation of meaning:

In a given language, all the words which express neighboring ideas help define one another's meaning. Each of a set of synonyms like *redouter* ('to dread'), *craindre* ('to fear'), *avoir peur* ('to be afraid') has its particular value only because they stand in contrast with one another. . . . No word has a value that can be identified independently of what else is in its vicinity.

There is often remarkable complexity in the differences between near-synonyms.³ Consider again Figure 1. The near-synonyms in the entry differ not only in the expression of various concepts and ideas, such as misconception and blameworthiness, but also in the manner in which the concepts are conveyed (e.g., implied, suggested,

³ This contrasts with Markman and Gentner's work on similarity (Markman and Gentner 1993; Gentner and Markman 1994), which suggests that the more similar two items are, the easier it is to represent their differences.

Table 1
Examples of near-synonymic variation.

Type of variation	Example
Abstract dimension	<i>seep : drip</i>
Emphasis	<i>enemy : foe</i>
Denotational, indirect	<i>error : mistake</i>
Denotational, fuzzy	<i>woods : forest</i>
Stylistic, formality	<i>pissed : drunk : inebriated</i>
Stylistic, force	<i>ruin : annihilate</i>
Expressed attitude	<i>skinny : thin : slim, slender</i>
Emotive	<i>daddy : dad : father</i>
Collocational	<i>task : job</i>
Selectional	<i>pass away : die</i>
Subcategorization	<i>give : donate</i>

expressed, connoted, and stressed), in the frequency with which they are conveyed (e.g., commonly, sometimes, not always), and in the degree to which they are conveyed (e.g., in strength).

2.3 Dimensions of Variation

The previous example illustrates merely one broad type of variation, denotational variation. In general, near-synonyms can differ with respect to any aspect of their meaning (Cruse 1986):

- denotational variations, in a broad sense, including propositional, fuzzy, and other peripheral aspects
- stylistic variations, including dialect and register
- expressive variations, including emotive and attitudinal aspects
- structural variations, including collocational, selectional, and syntactic variations

Building on an earlier analysis by DiMarco, Hirst, and Stede (1993) of the types of differentiae used in synonym discrimination dictionaries, Edmonds (1999) classifies near-synonymic variation into 35 subcategories within the four broad categories above. Table 1 gives a number of examples, grouped into the four broad categories above, which we will now discuss.

2.3.1 Denotational Variations. Several kinds of variation involve denotation, taken in a broad sense.⁴ DiMarco, Hirst, and Stede (1993) found that whereas some differentiae are easily expressed in terms of clear-cut abstract (or symbolic) features such as

⁴ The classic opposition of *denotation* and *connotation* is not precise enough for our needs here. The denotation of a word is its literal, explicit, and context-independent meaning, whereas its connotation is any aspect that is *not* denotational, including ideas that color its meaning, emotions, expressed attitudes, implications, tone, and style. *Connotation* is simply too broad and ambiguous a term. It often seems to be used simply to refer to any aspect of word meaning that we don't yet understand well enough to formalize.

continuous/intermittent (Wine {seeped | dripped} from the barrel), many are not. In fact, denotational variation involves mostly differences that lie not in simple features but in full-fledged concepts or ideas—differences in concepts that relate roles and aspects of a situation. For example, in Figure 1, “severe criticism” is a complex concept that involves both a criticizer and a criticized, the one who made the error. Moreover, two words can differ in the manner in which they convey a concept. *Enemy* and *foe*, for instance, differ in the emphasis that they place on the concepts that compose them, the former stressing antagonism and the latter active warfare rather than emotional reaction (Gove 1984).

Other words convey meaning **indirectly** by mere suggestion or implication. There is a continuum of indirectness from suggestion to implication to denotation; thus *slip* “carries a stronger implication of inadvertence” than *mistake*. Such indirect meanings are usually peripheral to the main meaning conveyed by an expression, and it is usually difficult to ascertain definitively whether or not they were even intended to be conveyed by the speaker; thus *error* merely “suggests guilt” and a *mistake* is “not always blameworthy.” Differences in denotation can also be fuzzy, rather than clear-cut. The difference between *woods* and *forest* is a complex combination of size, primitiveness, proximity to civilization, and wildness.⁵

2.3.2 Stylistic Variations. Stylistic variation involves differences in a relatively small, finite set of dimensions on which all words can be compared. Many stylistic dimensions have been proposed by Hovy (1988), Nirenburg and Defrise (1992), Stede (1993), and others. Table 1 illustrates two of the most common dimensions: *inebriated* is formal whereas *pissed* is informal; *annihilate* is a more forceful way of saying *ruin*.

2.3.3 Expressive Variations. Many near-synonyms differ in their marking as to the speaker’s attitude to their denotation: good thing or bad thing. Thus the same person might be described as *skinny*, if the speaker wanted to be deprecating or pejorative, *slim* or *slender*, if he wanted to be more complimentary, or *thin* if he wished to be neutral. A hindrance might be described as an *obstacle* or a *challenge*, depending upon how depressed or inspired the speaker felt about the action that it necessitated.⁶ A word can also indirectly express the emotions of the speaker in a possibly finite set of emotive “fields”; *daddy* expresses a stronger feeling of intimacy than *dad* or *father*. Some words are explicitly marked as **slurs**; a slur is a word naming a group of people, the use of which implies hatred or contempt of the group and its members simply by virtue of its being marked as a slur.

2.3.4 Structural Variations. The last class of variations among near-synonyms involves restrictions upon deployment that come from other elements of the utterance and, reciprocally, restrictions that they place upon the deployment of other elements. In either case, the restrictions are independent of the meanings of the words themselves.⁷ The

⁵ “A ‘wood’ is smaller than a ‘forest’, is not so primitive, and is usually nearer to civilization. This means that a ‘forest’ is fairly extensive, is to some extent wild, and on the whole not near large towns or cities. In addition, a ‘forest’ often has game or wild animals in it, which a ‘wood’ does not, apart from the standard quota of regular rural denizens such as rabbits, foxes and birds of various kinds” (Room 1985, page 270).

⁶ Or, in popular psychology, the choice of word may determine the attitude: “[Always] substitute *challenge* or *opportunity* for *problem*. . . . Instead of saying *I’m afraid that’s going to be a problem*, say *That sounds like a challenging opportunity*” (Walther 1992, page 36).

⁷ It could be argued that words that differ *only* in these ways should count not merely as near-synonyms but as absolute synonyms.

restrictions may be either collocational, syntactic, or selectional—that is, dependent either upon other words or constituents in the utterance or upon other concepts denoted.

Collocational variation involves the words or concepts with which a word can be combined, possibly idiomatically. For example, *task* and *job* differ in their collocational patterns: one can *face a daunting task* but not **face a daunting job*. This is a lexical restriction, whereas in selectional restrictions (or preferences) the class of acceptable objects is defined semantically, not lexically. For example, unlike *die*, *pass away* may be used only of people (or anthropomorphized pets), not plants or animals: **Many cattle passed away in the drought*.

Variation in syntactic restrictions arises from differing syntactic subcategorization. It is implicit that if a set of words are synonyms or near-synonyms, then they are of the same syntactic category.⁸ Some of a set of near-synonyms, however, might be subcategorized differently from others. For example, the adjective *ajar* may be used predicatively, not attributively (*The door is ajar*; **the ajar door*), whereas the adjective *open* may be used in either position. Similarly, verb near-synonyms (and their nominalizations) may differ in their verb class and in the alternations that they they may undergo (Levin 1993). For example, *give* takes the dative alternation, whereas *donate* does not: *Nadia gave the Van Gogh to the museum*; *Nadia gave the museum the Van Gogh*; *Nadia donated the Van Gogh to the museum*; **Nadia donated the museum the Van Gogh*.

Unlike the other kinds of variation, collocational, syntactic, and selectional variations have often been treated in the literature on lexical choice, and so we will have little more to say about them here.

2.4 Cross-Linguistic Near-Synonymy

Near-synonymy rather than synonymy is the norm in lexical transfer in translation: the word in the target language that is closest to that in the source text might be a near-synonym rather than an exact synonym. For example, the German word *Wald* is similar in meaning to the English word *forest*, but *Wald* can denote a rather smaller and more urban area of trees than *forest* can; that is, *Wald* takes in some of the English word *woods* as well, and in some situations, *woods* will be a better translation of *Wald* than *forest*. Similarly, the German *Gehölz* takes in the English *copse* and the “smaller” part of *woods*. We can think of *Wald*, *Gehölz*, *forest*, *woods*, and *copse* as a cross-linguistic near-synonym group.

Hence, as with a group of near-synonyms from a single language, we can speak of the differences in a group of cross-linguistic near-synonyms. And just as there are reference books to advise on the near-synonym groups of a single language, there are also books to advise translators and advanced learners of a second language on cross-linguistic near-synonymy. As an example, we show in Figures 2 and 3 (abridgements of) the entries in Farrell (1977) and Batchelor and Offord (1993) that explicate, from the perspective of translation to and from English, the German and French near-synonym clusters that correspond to the English cluster for *error* that we showed in Figure 1.

2.5 Summary

We know that near-synonyms can often be intersubstituted with no apparent change of effect on a particular utterance, but, unfortunately, the context-dependent nature

⁸ A rigorous justification of this point would run to many pages, especially for near-synonyms. For example, it would have to be argued that the verb *sleep* and the adjective *asleep* are not merely near-synonyms that just happen to differ in their syntactic categories, even though the sentences *Emily sleeps* and *Emily is asleep* are synonymous or nearly so.

MISTAKE, ERROR. *Fehler* is a definite imperfection in a thing which ought not to be there. In this sense, it translates both *mistake* and *error*. *Irrtum* corresponds to *mistake* only in the sense of 'misunderstanding', 'misconception', 'mistaken judgment', i.e. which is confined to the mind, not embodied in something done or made. [footnote:] *Versehen* is a petty mistake, an oversight, a slip due to inadvertence. *Mißgriff* and *Fehlgriff* are mistakes in doing a thing as the result of an error in judgment.

Figure 2

An entry (abridged) from *Dictionary of German Synonyms* (Farrell 1977).

impair (3) blunder, error
bévue (3-2) blunder (due to carelessness or ignorance)
faux pas (3-2) mistake, error (which affects a person adversely socially or in his/her career, etc)
bavure (2) unfortunate error (often committed by the police)
bêtise (2) stupid error, stupid words
gaffe (2-1) boob, clanger

Figure 3

An entry (abridged) from *Using French Synonyms* (Batchelor and Offord 1993). The parenthesized numbers represent formality level from 3 (*most formal*) to 1 (*least formal*).

of lexical knowledge is not very well understood as yet. Lexicographers, for instance, whose job it is to categorize different uses of a word *depending* on context, resort to using mere "frequency" terms such as *sometimes* and *usually* (as in Figure 1). Thus, we cannot yet make any claims about the influence of context on near-synonymy.

In summary, to account for near-synonymy, a model of lexical knowledge will have to incorporate solutions to the following problems:

- The four main types of variation are qualitatively different, so each must be separately modeled.
- Near-synonyms differ in the manner in which they convey concepts, either with emphasis or indirectness (e.g., through mere suggestion rather than denotation).
- Meanings, and hence differences among them, can be fuzzy.
- Differences can be multidimensional. Only for clarity in our above explication of the dimensions of variation did we try to select examples that highlighted a single dimension. However, as Figure 1 shows, *blunder* and *mistake*, for example, actually differ on several denotational dimensions as well as on stylistic and attitudinal dimensions.
- Differences are not just between simple features but involve concepts that relate roles and aspects of the situation.
- Differences often depend on the context.

3. Near-Synonymy in Computational Models of the Lexicon

Clearly, near-synonymy raises questions about fine-grained lexical knowledge representation. But is near-synonymy a phenomenon in its own right warranting its own

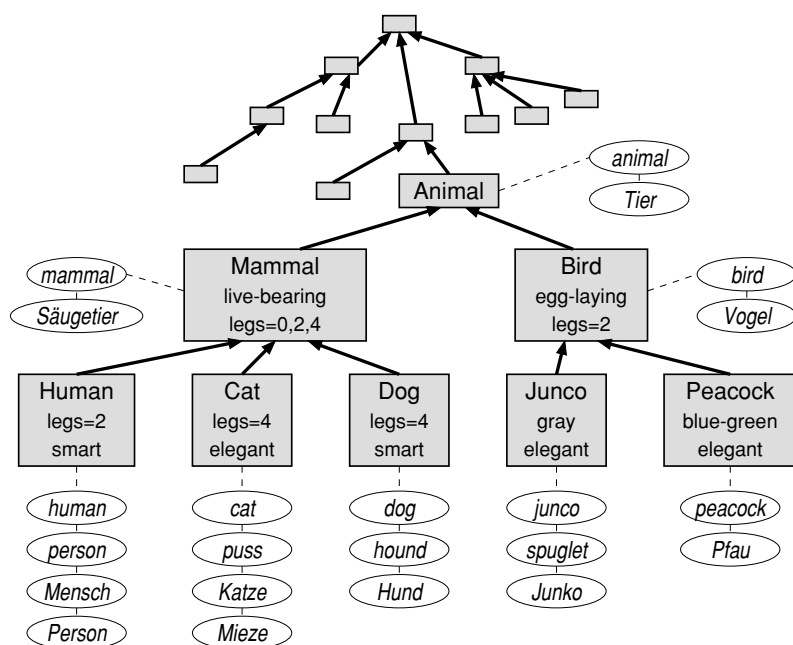


Figure 4

A simplistic hierarchy of conceptual schemata with connections to their lexical entries for English and German.

special account, or does it suffice to treat near-synonyms the same as widely differing words? We will argue now that near-synonymy is indeed a separately characterizable phenomenon of word meaning.

Current models of lexical knowledge used in computational systems, which are based on decompositional and relational theories of word meaning (Katz and Fodor 1963; Jackendoff 1990; Lyons 1977; Nirenburg and Defrise 1992; Lehrer and Kittay 1992; Evens 1988; Cruse 1986), cannot account for the properties of near-synonyms. In these models, the typical view of the relationship between words and concepts is that each element of the lexicon is represented as a conceptual schema or a structure of such schemata. Each word sense is linked to the schema or the conceptual structure that it lexicalizes. If two or more words denote the same schema or structure, all of them are connected to it; if a word is ambiguous, subentries for its different senses are connected to their respective schemata. In this view, then, to understand a word in a sentence is to find the schema or schemata to which it is attached, disambiguate if necessary, and add the result to the output structure that is being built to represent the sentence. Conversely, to choose a word when producing an utterance from a conceptual structure is to find a suitable set of words that “cover” the structure and assemble them into a sentence in accordance with the syntactic and pragmatic rules of the language (Nogier and Zock 1992; Stede 1999).

A conceptual schema in models of this type is generally assumed to contain a set of *attributes* or *attribute-value pairs* that represent the content of the concept and differentiate it from other concepts. An attribute is itself a concept, as is its value. The conceptual schemata are themselves organized into an inheritance hierarchy, taxonomy, or ontology; often, the ontology is language-independent, or at least language-neutral, so that it can be used in multilingual applications. Thus, the model might look

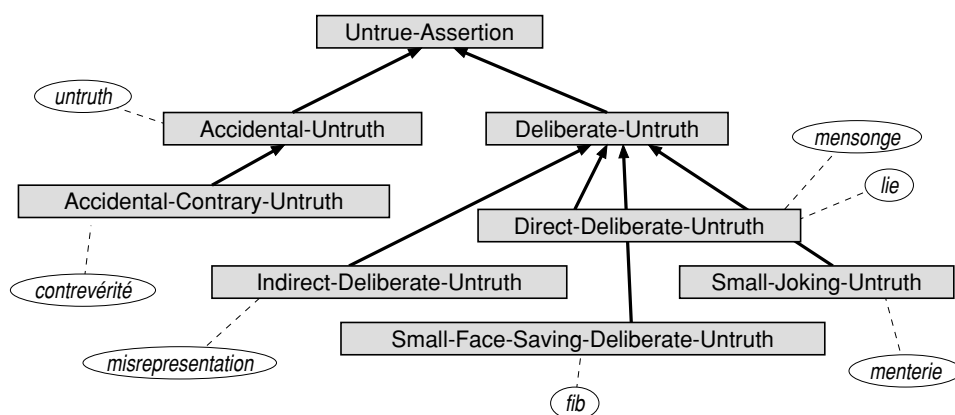


Figure 5
One possible hierarchy for the various English and French words for untrue assertions.
Adapted from Hirst (1995).

like the simplified fragment shown in Figure 4. In the figure, the rectangles represent concept schemata with attributes; the arrows between them represent inheritance. The ovals represent lexical entries in English and German; the dotted lines represent their connection to the concept schemata.⁹

Following Frege's (1892) or Tarski's (1944) truth-conditional semantics, the concept that a lexical item denotes in such models can be thought of as a set of features that are individually necessary and collectively sufficient to define the concept. Such a view greatly simplifies the word-concept link. In a text generation system, for instance, the features amount to the necessary applicability conditions of a word; that is, they have to be present in the input in order for the word to be chosen. Although such models have been successful in computational systems, they are rarely pushed to represent near-synonyms. (The work of Barnett, Mani, and Rich [1994] is a notable exception; they define a relation of semantic closeness for comparing the denotations of words and expressions; see Section 9.) They do not lend themselves well to the kind of fine-grained and often fuzzy differentiation that we showed earlier to be found in near-synonymy, because, in these models, except as required by homonymy and absolute synonymy, there is no actual distinction between a word and a concept: each member of a group of near-synonyms must be represented as a separate concept schema (or group of schemata) with distinct attributes or attribute values. For example, Figure 5 shows one particular classification of the *fib* group of near-synonyms in English and French.¹⁰ A similar proliferation of concepts would be required for various *error* clusters (as shown earlier in Figures 1, 2, and 3).

⁹ This outline is intended as a syncretism of many models found in the interdisciplinary literature and is not necessarily faithful to any particular one. For examples, see the papers in Evens (1988) (especially Sowa [1988]) and in Pustejovsky and Bergler (1992) (especially Nirenburg and Levin [1992], Sowa [1992], and Burkert and Forster [1992]); for a theory of lexico-semantic taxonomies, see Kay (1971). For a detailed construction of the fundamental ideas, see Barsalou (1992); although we use the term *schema* instead of *frame*, despite Barsalou's advice to the contrary, we tacitly accept most elements of his model. For bilingual aspects, see Kroll and de Groot (1997).

¹⁰ We do not claim that a bilingual speaker necessarily stores words and meanings from different languages together. In this model, if the concepts are taken to be language independent, then it does not matter if one overarching hierarchy or many distinct hierarchies are used. It is clear, however, that cross-linguistic near-synonyms do not have exactly the same meanings and so require distinct concepts in this model.

Although some systems have indeed taken this approach (Emele et al. 1992), this kind of fragmentation is neither easy nor natural nor parsimonious. Hirst (1995) shows that even simple cases lead to a multiplicity of nearly identical concepts, thereby defeating the purpose of a language-independent ontology. Such a taxonomy cannot efficiently represent the multidimensional nature of near-synonymic variation, nor can it account for fuzzy differences between near-synonyms. And since the model defines words in terms of only necessary and sufficient truth-conditions, it cannot account for indirect expressions of meaning and for context-dependent meanings, which are clearly not necessary features of a word's meaning.

Moreover, a taxonomic hierarchy emphasizes hyponymy, backgrounding all other relations, which appear to be more important in representing the multidimensional nature of fine-grained word meaning. It is not even clear that a group of synonyms *can* be structured by hyponymy, except trivially (and ineffectively) as hyponyms all of the same concept.

The model also cannot easily or tractably account for fuzzy differences or the full-fledged concepts required for representing denotational variation. First-order logic, rather than the description logic generally used in ontological models, would at least be required to represent such concepts, but reasoning about the concepts in lexical choice and other tasks would then become intractable as the model was scaled up to represent all near-synonyms.

In summary, present-day models of the lexicon have three kinds of problems with respect to near-synonymy and fine-grained lexical knowledge: the adequacy of coverage of phenomena related to near-synonymy; engineering, both in the design of an efficient and robust lexical choice process and in the design of lexical entries for near-synonyms; and the well-known issues of tractability of reasoning about concepts during natural language understanding and generation.

Nevertheless, at a coarse grain, the ontological model does have practical and theoretical advantages in efficient paraphrasing, lexical choice, and mechanisms for inference and reasoning. Hence, to build a new model of lexical knowledge that takes into account the fine-grainedness of near-synonymy, a logical way forward is to start with the computationally proven ontological model and to modify or extend it to account for near-synonymy. The new model that we will present below will rely on a much more coarsely grained ontology. Rather than proliferating conceptual schemata to account for differences between near-synonyms, we will propose that near-synonyms are connected to a single concept, despite their differences in meaning, and are differentiated at a subconceptual level. In other words, the connection of two or more words to the same schema will not imply synonymy but only near-synonymy. Differentiation between the near-synonyms—the fine tuning—will be done in the lexical entries themselves.

4. Near-Synonymy and Granularity of Representation

To introduce the notion of granularity to our discussion, we first return to the problem of defining near-synonymy.

Semanticists such as Ullmann (1962), Cruse (1986), and Lyons (1995) have attempted to define near-synonymy by focusing on “propositional” meaning. Cruse, for example, contrasts **cognitive synonyms** and **plesionyms**; the former are words that, when intersubstituted in a sentence, preserve its truth conditions but may change the expressive meaning, style, or register of the sentence or may involve different idiosyn-

cratic collocations (e.g., *violin* : *fiddle*),¹¹ whereas intersubstituting the latter changes the truth conditions but still yields semantically similar sentences (e.g., *misty* : *foggy*). Although these definitions are important for truth-conditional semantics, they are not very helpful for us, because plesionymy is left to handle all of the most interesting phenomena discussed in Section 2. Moreover, a rigorous definition of cognitive synonymy is difficult to come up with, because it relies on the notion of granularity, which we will discuss below.

Lexicographers, on the other hand, have always treated synonymy as near-synonymy. They define synonymy in terms of likeness of meaning, disagreeing only in how broad the definition ought to be. For instance, Roget followed the vague principle of “the grouping of words according to ideas” (Chapman 1992, page xiv). And in the hierarchical structure of *Roget’s Thesaurus*, word senses are ultimately grouped according to proximity of meaning: “the sequence of terms within a paragraph, far from being random, is determined by close, semantic relationships” (page xiii). The lexicographers of *Webster’s New Dictionary of Synonyms* define a synonym as “one of two or more words . . . which have the same or very nearly the same *essential* meaning. . . . Synonyms can be defined in the same terms up to a certain point” (Egan 1942, pages 24a–25a). *Webster’s Collegiate Thesaurus* uses a similar definition that involves the sharing of elementary meanings, which are “discrete objective denotations uncolored by . . . peripheral aspects such as connotations, implications, or quirks of idiomatic usage” (Kay 1988, page 9a). Clearly, the main point of these definitions is that near-synonyms must have the same essential meaning but may differ in peripheral or subordinate ideas. Cruse (1986, page 267) actually refines this idea and suggests that synonyms (of all types) are words that are identical in “central semantic traits” and differ, if at all, only in “peripheral traits.” But how can we specify formally just how much similarity of central traits and dissimilarity of peripheral traits is allowed? That is, just what counts as a central trait and what as a peripheral trait in defining a word?

To answer this question, we introduce the idea of **granularity of representation** of word meaning. By granularity we mean the level of detail used to describe or represent the meanings of a word. A fine-grained representation can encode subtle distinctions, whereas a coarse-grained representation is crude and glosses over variation. Granularity is distinct from **specificity**, which is a property of concepts rather than representations of concepts. For example, a rather general (unspecific) concept, say *Human*, could have, in a particular system, a very fine-grained representation, involving, say, a detailed description of the appearance of a human, references to related concepts such as *Eat* and *Procreate*, and information to distinguish the concept from other similar concepts such as *Animal*. Conversely, a very specific concept could have a very coarse-grained representation, using only very general concepts; we could represent a *Lexicographer* at such a coarse level of detail as to say no more than that it is a physical object.

Near-synonyms can occur at any level of specificity, but crucially it is the fine granularity of the representations of their meanings that enables one to distinguish one near-synonym from another. Thus, any definition of near-synonymy that does not take granularity into account is insufficient. For example, consider Cruse’s cognitive synonymy, discussed above. On the one hand, at an absurdly coarse grain of representation, any two words are cognitive synonyms (because every word denotes a “thing”). But on the other hand, no two words could ever be known to be cognitive synonyms, because, even at a fine grain, apparent cognitive synonyms might be fur-

11 What’s the difference between a violin and a fiddle? No one minds if you spill beer on a fiddle.

ther distinguishable by a still more fine-grained representation. Thus, granularity is essential to the concept of cognitive synonymy, as which pairs of words are cognitive synonyms depends on the granularity with which we represent their propositional meanings. The same is true of Cruse's plesionyms. So in the end, it should not be necessary to make a formal distinction between cognitive synonyms and plesionyms. Both kinds of near-synonyms should be representable in the same formalism.

By taking granularity into account, we can create a much more useful definition of near-synonymy, because we can now characterize the difference between **essential** and **peripheral** aspects of meaning. If we can set an appropriate level of granularity, the essential meaning of a word is the portion of its meaning that is representable only above that level of granularity, and peripheral meanings are those portions representable only below that level.

But what is the appropriate level of granularity, the dividing line between coarse-grained and fine-grained representations? We could simply use our intuition—or rather, the intuitions of lexicographers, which are filtered by some amount of objectivity and experience. Alternatively, from a concern for the representation of lexical knowledge in a multilingual application, we can view words as (language-specific) specializations of language-independent concepts. Given a hierarchical organization of coarse-grained language-independent concepts, a set of near-synonyms is simply a set of words that all link to the same language-independent concept (DiMarco, Hirst, and Stede 1993; Hirst 1995). So in this view, near-synonyms share the same propositional meaning just up to the point in granularity defined by language dependence. Thus we have an operational definition of near-synonymy: If the same concept has several reasonable lexicalizations in different languages, then it is a good candidate for being considered a language-independent concept, its various lexicalizations forming sets of near-synonyms in each language.¹²

Granularity also explains why it is more difficult to represent near-synonyms in a lexicon. Near-synonyms are so close in meaning, sharing all essential coarse-grained aspects, that they differ, by definition, in only aspects representable at a fine grain. And these fine-grained representations of differences tend to involve very specific concepts, typically requiring complex structures of more general concepts that are difficult to represent and to reason with. The matter is only made more complicated by there often being several interrelated near-synonyms with interrelated differences. On the other hand, words that are not near-synonyms—those that are merely similar in meaning (*dog : cat*) or not similar at all (*dog : hat*)—could presumably be differentiated by concepts at a coarse-grained, and less complex, level of representation.

5. A Model of Fine-Grained Lexical Knowledge

Our discussion of granularity leads us to a new model of lexical knowledge in which near-synonymy is handled on a separate level of representation from coarse-grained concepts.

5.1 Outline of the Model

Our model is based on the contention that the meaning of an open-class content word, however it manifests itself in text or speech, *arises* out of a context-dependent combination of a basic inherent context-independent denotation and a set of explicit differences

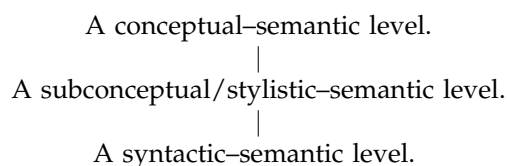
¹² EuroWordNet's Inter-Lingual-Index (Vossen 1998) links the synsets of different languages in such a manner, and Resnik and Yarowsky (1999) describe a related notion for defining word senses cross-lingually.

to its near-synonyms. (We don't rule out other elements in the combination, but these are the main two.) Thus, word meaning is not explicitly represented in the lexicon but is created (or generated, as in a generative model of the lexicon [Pustejovsky 1995]) when a word is used. This theory preserves some aspects of the classical theories—the basic denotation can be modeled by an ontology—but the rest of a word's meaning relies on other nearby words and the context of use (cf. Saussure). In particular, each word and its near synonyms form a **cluster**.¹³

The theory is built on the following three ideas, which follow from our observations about near-synonymy. First, the meaning of any word, at some level of granularity, must indeed have some inherent context-independent denotational aspect to it—otherwise, it would not be possible to define or “understand” a word in isolation of context, as one in fact can (as in dictionaries). Second, nuances of meaning, although difficult or impossible to represent in positive, absolute, and context-independent terms, can be represented as differences, in Saussure's sense, between near-synonyms. That is, every nuance of meaning that a word might have can be thought of as a relation between the word and one or more of its near-synonyms. And third, differences must be represented not by simple features or truth conditions, but by structures that encode relations to the context, fuzziness, and degrees of necessity.

For example, the word *forest* denotes a geographical tract of trees at a coarse grain, but it is only in relation to *woods*, *copse*, and other near-synonyms that one can fully understand the significance of *forest* (i.e., that it is larger, wilder, etc.). The word *mistake* denotes any sort of action that deviates from what is correct and also involves some notion of criticism, but it is only in relation to *error* and *blunder* that one sees that the word can be used to criticize less severely than these alternatives allow. None of these differences could be represented in absolute terms, because that would require defining some absolute notion of size, wildness, or severity, which seems implausible. So, at a fine grain, and only at a fine grain, we make explicit use of Saussure's notion of contrast in demarcating the meanings of near-synonyms. Hence, the theory holds that near-synonyms are explicitly related to each other not at a conceptual level but at a **subconceptual** level—outside of the (coarser-grained) ontology. In this way, a cluster of near-synonyms is not a mere list of synonyms; it has an internal structure that encodes fine-grained meaning as differences between lexical entries, and it is situated between a conceptual model (i.e., the ontology) and a linguistic model.

Thus the model has three levels of representation. Current computational theories suggest that at least two levels of representation, a **conceptual–semantic** level and a **syntactic–semantic** level, are necessary to account for various lexico-semantic phenomena in computational systems, including compositional phenomena such as paraphrasing (see, for instance, Stede's [1999] model). To account for fine-grained meanings and near-synonymy, we postulate a third, intermediate level (or a splitting of the conceptual–semantic level). Thus the three levels are the following:



¹³ It is very probable that many near-synonym clusters of a language could be discovered automatically by applying statistical techniques, such as cluster analysis, on large text corpora. For instance, Church et al. (1994) give some results in this area.

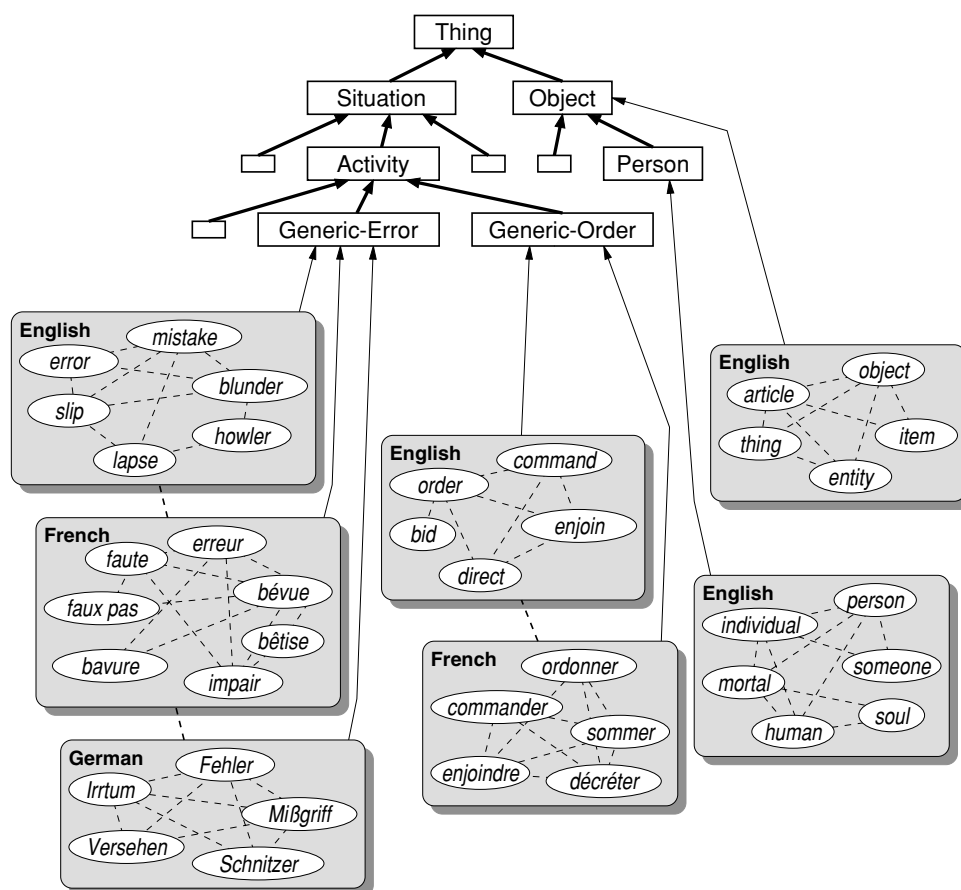


Figure 6
A clustered model of lexical knowledge

So, taking the conventional ontological model as a starting point, we cut off the ontology at a coarse grain and cluster near-synonyms under their shared concepts rather than linking each word to a separate concept. The resulting model is a **clustered model of lexical knowledge**. On the conceptual–semantic level, a cluster has a **core denotation** that represents the essential shared denotational meaning of its near-synonyms. On the subconceptual/stylistic–semantic level, we represent the fine-grained differences between the near-synonyms of a cluster in denotation, style, and expression. At the syntactic–semantic level, syntactic frames and collocational relations represent how words can be combined with others to form sentences.

Figure 6 depicts a fragment of the clustered model. It shows how the clusters of the near-synonyms of *error*, *order*, *person*, and *object* in several languages could be represented in this model. In the figure, each set of near-synonyms forms a cluster linked to a coarse-grained concept defined in the ontology: *Generic-Error*, *Generic-Order*, *Person*, and *Object*, respectively. Thus, the core denotation of each cluster is the concept to which it points. Within each cluster, the near-synonyms are differentiated at the subconceptual/stylistic level of semantics, as indicated by dashed lines between the words in the cluster. (The actual differences are not shown in the figure.) The dashed lines between the clusters for each language indicate similar cross-linguistic differenti-

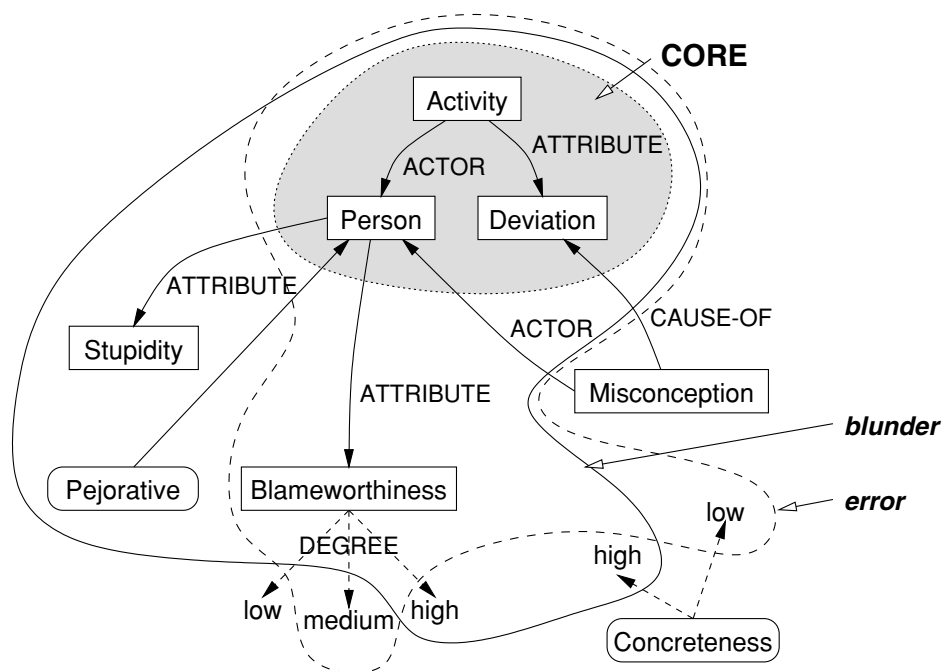


Figure 7
 The core denotation and some of the peripheral concepts of the cluster of *error* nouns. The two large regions, bounded by the solid line and the dashed line, show the concepts (and attitudes and styles) that can be conveyed by the words *error* and *blunder* in relation to each other.

ation between some or all of the words of each cluster. Not all words in a cluster need be differentiated, and each cluster in each language could have its own “vocabulary” for differentiating its near-synonyms, though in practice one would expect an overlap in vocabulary. The figure does not show the representation at the syntactic–semantic level. We can now describe the internal structure of a cluster in more detail, starting with two examples.

Figure 7 depicts part of the representation of the cluster of *error* nouns (*error*, *mistake*, *blunder*, ...); it is explicitly based on the entry from *Webster’s New Dictionary of Synonyms* shown in Figure 1. The core denotation, the shaded region, represents an activity by a person (the actor) that is a deviation from a proper course.¹⁴ In the model, **peripheral concepts** are used to represent the denotational distinctions of near-synonyms. The figure shows three peripheral concepts linked to the core concept: Stupidity, Blameworthiness, and Misconception. The peripheral concepts represent that a word in the cluster can potentially express, in relation to its near-synonyms, the stupidity of the actor of the error, the blameworthiness of the actor (of different degrees: low, medium, or high), and misconception as cause of the error. The representation also contains an expressed attitude, Pejorative, and the stylistic dimension of Concreteness. (Concepts are depicted as regular rectangles, whereas stylistic dimensions and attitudes are depicted as rounded rectangles.) The core denotation and peripheral concepts together form a directed graph of concepts linked by relations;

¹⁴ Specifying the details of an actual cluster should be left to trained knowledge representation experts, who have a job not unlike a lexicographer’s. Our model is intended to encode such knowledge once it is elucidated.

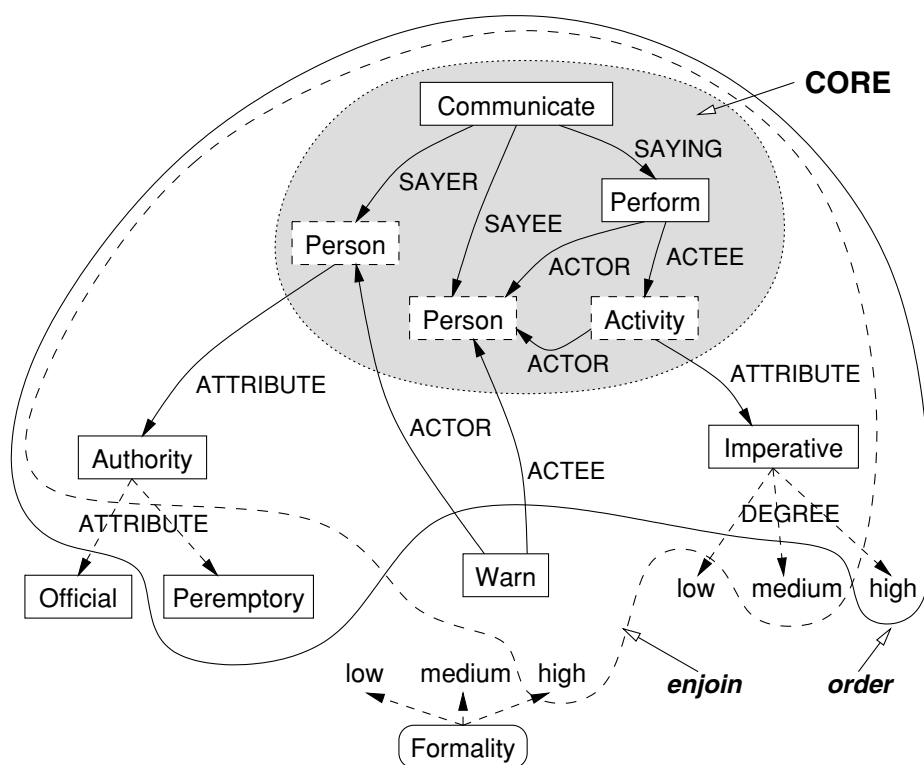


Figure 8

The core denotation and peripheral concepts of the cluster of *order* verbs. The two large regions, bounded by the solid line and the dashed line, show the concepts that can be conveyed by the words *order* and *enjoin* in relation to each other.

the individual concepts and relations are defined in the ontology. But although all of the near-synonyms in the cluster will convey the concepts in the core denotation, the peripheral concepts that will be conveyed depend on each near-synonym. This is depicted by the two large regions in the figure (bounded by the solid line and the dashed line), which each contain the concepts, styles, and attitudes conveyed by their associated near-synonyms, *blunder* and *error*, respectively. Thus, *error* conveys a degree of Blameworthiness compared to the higher degree that *blunder* conveys; *error* does not convey Stupidity whereas *blunder* does; *blunder* can also express a Pejorative attitude toward the actor, but *error* does not express any attitude; and *error* and *blunder* differ stylistically in their degree of Concreteness. Notice that the attitude connects to the concept Person, because all attitudes must be directed toward some entity in the situation. Stylistic dimensions such as Concreteness, on the other hand, are completely separate from the graph of concepts. Also, the indirectness of expression of each of the peripheral concepts by each of the near-synonyms is not shown in this diagram (but see below). The Appendix gives the complete representation of this cluster in the formalism of our model.

Similarly, Figure 8 depicts the cluster of *order* verbs (*order*, *enjoin*, *command*, ...), including three of its peripheral concepts and one stylistic dimension. In this cluster, the core represents a communication by a person (the sayer) to another person (the sayee) of an activity that the sayee must perform. The core includes several concepts that are not actually lexicalized by any of the words in the cluster (e.g., the sayer of the

order) but that nevertheless have to be represented because the peripheral concepts refer to them. (Such concepts are indicated by dashed rectangles.) The peripheral concepts represent the idea that a near-synonym can express the authority of the sayer (with possible values of *Official* or *Peremptory*), a warning to the sayee, and the imperativeness of the activity (with possible values of *low*, *medium*, or *high*). The figure shows the difference between *order* (the region bounded by the solid line) and *enjoin* (the region bounded by the dashed line).

5.2 Core Denotation

The core denotation of a cluster is the inherent context-independent (and in this formulation of the model, language-neutral) denotation shared by all of its near-synonyms. The core denotation must be specified at a level of granularity sufficient to form a useful cluster of near-synonyms (i.e., at the right level of granularity so that, for instance, *human* and *person* fall into the same cluster, but *dwarf* and *giant* do not; see Section 4).

A core denotation is represented as a directed graph of concepts linked by relations. The graph can be of arbitrary size, from a single concept (such as *Generic-Error*) up to any number of interrelated concepts (as shown in Figures 7 and 8). It must be specified in enough detail, however, for the peripheral concepts to also be specified. For instance, in the *error* cluster, it was not possible to use the simple concept *Generic-Error*, because the peripheral concepts of the cluster refer to finer-grained aspects of the concept (the actor and the deviation); hence we used a finer-grained representation of the concept.

5.3 Peripheral Concepts

Peripheral concepts form the basic vocabulary of fine-grained denotational distinctions. They are used to represent non-necessary and indirect aspects of word meaning. That is, they are concepts that might be implied, suggested, emphasized, or otherwise when a word is used, but not always. For instance, in differentiating the *error* words, a lexicographer would first decide that the basic peripheral concepts required might be ‘stupidity’, ‘blameworthiness’, ‘criticism’, ‘misconception’, ‘accidentalness’, and ‘inattention’. Then the lexicographer would proceed to distinguish the near-synonyms in terms of these concepts, for instance, by specifying that *blunder* involves a higher degree of blameworthiness than *error*.

More formally, peripheral concepts are structures of concepts defined in the same ontology that core denotations are defined in. In fact, every peripheral concept in a cluster must “extend” the core denotation in some way, because, after all, peripheral concepts represent ideas related to the core meaning of a cluster of near-synonyms. But peripheral concepts are represented separately from the core denotation.

Moreover, since peripheral concepts are defined in the ontology, they can be reasoned about, which, in principle, makes the formalism robust to variation in representation. That is, if a lexicographer used, say, ‘responsibility’ to define *mistake* and ‘blameworthiness’ to define *blunder*, the words could still be compared, because inference would find a connection between ‘responsibility’ and ‘blameworthiness’. See Section 6.1 below for more discussion on this point.

5.4 Distinctions between Near-Synonyms

Following Hirst (1995), we would like to represent differences explicitly as first-class objects (so that we can reason about them during processing). While we don’t adopt an explicit formalism, for reasons of practicality of representation, our implicit formalism provides a method for computing explicit differences as needed (as we’ll see in Section 6.1). Thus we associate with each near-synonym in a cluster a set of

Table 2
Examples of distinctions of words.

Denotational distinctions:		
Binary:	<i>blunder</i> :	(usually medium implication Stupidity)
Continuous:	<i>blunder</i> :	(always medium implication (Blameworthiness (DEGREE high)))
Discrete:	<i>order</i> :	(always medium implication (Authority (ATTRIBUTE (Peremptory))))
Expressive distinctions:		
	<i>blunder</i> :	(always medium pejorative V1)
Stylistic distinctions:		
	<i>blunder</i> :	(high concreteness)
	<i>error</i> :	(low concreteness)

Note: See the Appendix for the details. In the expressive distinction, V1 is a variable that refers to the actor of the error as specified in the core denotation, and in the denotational distinction high is a fuzzy set of values in the range [0, 1].

distinctions that are taken to be relative within the cluster; the cluster establishes the local frame of reference for comparing them. So a word's set of distinctions implicitly differentiates the word from its near-synonyms. In other words, if one considers the peripheral concepts, attitudes, styles, and so on, to be dimensions, then the set of distinctions situates a word in a multidimensional space relative to its near-synonyms. We define three types of distinction below: denotational, expressive, and stylistic.

5.4.1 Denotational Distinctions. In our formalism, each denotational distinction refers to a particular peripheral concept and specifies a value on that dimension, which can be binary (i.e., is or isn't expressed), continuous (i.e., takes a possibly fuzzy value in the range [0, 1]), or discrete (i.e., takes a conceptual structure as a value).

Now, in Section 2.3.1 we observed that indirectness forms a continuum (*suggestion, implication, denotation*), and, following the method used by lexicographers in near-synonym guides, points on the continuum are modulated up or down by a **strength**, which can take the values *weak, medium, or strong*. To also account for context dependence at least as well as lexicographers do, we include a measure of the frequency with which the peripheral concept is conveyed by the word. This can take one of five values (*never, seldom, sometimes, often, always*). When the problem of context dependence is better understood, this part of the formalism will need to be changed.

Thus, a denotational distinction of a word w is a quadruple of components as follows:

$$w: (\text{frequency strength indirectness concept})$$

The first part of Table 2 gives some examples for the distinctions of Figures 7 and 8.

5.4.2 Expressive Distinctions. Since a word can express a speaker's attitude toward potentially any entity in a situation, an expressive distinction must include a reference to the entity. As for the attitude itself, we take a conservative approach, for now, and

define only three possible attitudes: *favorable*, *neutral*, and *pejorative*. Thus, an expressive distinction has the following form:

w: (frequency strength attitude entity)

Frequency and strength have the same role as above. The entity is actually a reference (i.e., a variable) to one of the concepts specified in the core denotation of peripheral concepts. The second part of Table 2 gives an example.

5.4.3 Stylistic Distinctions. Although we take a rather basic approach to representing stylistic distinctions, that does not imply that style is easy to capture. Style is one of the most difficult of lexical phenomena to account for, since it affects the text at a pragmatic level and is highly influenced by context. Since there is as yet no comprehensive theory of style, our approach is similar to past approaches, such as those of DiMarco and Hirst (1993), Stede (1993), and Hovy (1988).

Unlike the denotational distinctions discussed above, stylistic features have a global or absolute quality to them. We can compare all words, whether or not they are near-synonyms, on various stylistic dimensions, such as formality and concreteness. Because style is a global aspect of text, a certain style can be (and should be) achieved by more than just lexical choice; structural choices are just as important (DiMarco and Hirst 1993). Hence, in defining a set of stylistic dimensions, we must look for global stylistic features that can be carried not only by words but also by syntactic and larger text structures. Our stylistic dimensions include, but are not limited to, **formality**, **force**, **concreteness**, **floridity**, and **familiarity**.

Stylistic variation also differs from the other types of variation in being related solely to the lexeme itself and not to its denotation or conceptual meaning (though in a deeper sense style is certainly related to meaning). So in representing stylistic distinctions we don't have to make any reference to entities or other aspects of the core denotation or peripheral concepts in a cluster. Thus, we represent a stylistic distinction as follows:

w: (degree dimension)

where *degree* can take a value of *low*, *medium*, or *high* (though more values could easily be added to increase the precision). The third part of Table 2 gives two examples.

6. Lexical Similarity

It is not sufficient merely to represent differences between near-synonyms; we must also be able to use these representations effectively. For lexical choice, among other tasks, we need to be able to compare the similarities of pairs of near-synonyms. For example, in a transfer-based MT system, in order to translate the French word *bavure* into English, we need to compare the similarities of at least the three pairs *bavure* : *error*, *bavure* : *mistake*, and *bavure* : *blunder* and choose the English word whose meaning is closest to *bavure*, subject to any constraints arising from the context. And in text generation or interlingual MT, we need to be able to compare the similarities of each of several near-synonyms to a particular semantic representation or conceptual structure in order to choose the one that is closest to it in meaning.

Now, the general problem of measuring the **semantic distance** between words or concepts has received much attention. This century, Wittgenstein (1953) formulated the notion of **family resemblance**—that several things can be related because

they overlap with respect to a set of properties, no property being common to all of the words—which Rosch (1978) then used as the basis for the prototype theory of meaning. Recent research in computational linguistics has focused more on developing methods to compute the *degree* of semantic similarity between any two words, or, more precisely, between the simple or primitive concepts¹⁵ denoted by any two words.

There are many different similarity measures, which variously use taxonomic lexical hierarchies or lexical-semantic networks, large text corpora, word definitions in machine-readable dictionaries or other semantic formalisms, or a combination of these (Dagan, Marcus, and Markovitch 1993; Kozima and Furugori 1993; Pereira, Tishby, and Lee 1993; Church et al. 1994; Grefenstette 1994; Resnik 1995; McMahon and Smith 1996; Jiang and Conrath 1997; Schütze 1998; Lin 1998; Resnik and Diab 2000; Budanitsky 1999; Budanitsky and Hirst 2001, 2002). Unfortunately, these methods are generally unhelpful in computing the similarity of near-synonyms because the measures lack the required precision. First, taxonomic hierarchies and semantic networks inherently treat near-synonyms as absolute synonyms in grouping near-synonyms into single nodes (e.g., in WordNet). In any case, as we argued in Section 3, taxonomies are inappropriate for modeling near-synonyms. Second, as we noted in Section 2.2, standard dictionary definitions are not usually fine-grained enough (they define the core meaning but not all the nuances of a word) and can even be circular, defining each of several near-synonyms in terms of the other near-synonyms. And third, although corpus-based methods (e.g., Lin’s [1998]) do compute different similarity values for different pairs of near-synonyms of the same cluster, Church et al. (1994) and Edmonds (1997) show that such methods are not yet capable of uncovering the more subtle differences in the use of near-synonyms for lexical choice.

But one benefit of the clustered model of lexical knowledge is that it naturally lends itself to the computation of explicit differences or degrees of similarity between near-synonyms. Although a fully effective similarity measure for near-synonyms still eludes us, in this section we will characterize the problem and give a solution to one part of it: computing the similarity of individual lexical distinctions.

6.1 Computing the Similarity of Near-Synonyms

In the clustered model of lexical knowledge, a difference between two near-synonyms is encoded implicitly in two sets of relative distinctions. From two such sets of distinctions, one can compute, or build, an explicit representation of the difference between two near-synonyms. Thus, the difference between, or similarity of, two near-synonyms depends on the semantic content of their representations on the subconceptual/stylistic level (cf. Resnik and Diab [2000], in which similarity is computed according to the structure, rather than content, of lexical conceptual structure representations of verbs; see Jackendoff [1983] and Dorr [1993]).

Comparing two sets of distinctions is not straightforward, however, because, near-synonyms often differ on seemingly incommensurate dimensions. That is, the distinctions of one near-synonym will often not match those of another near-synonym, leaving no basis for comparison. For instance, in Figure 9, *bavure* and *mistake* align on only two of five denotational dimensions (Blameworthiness and Criticism), and this assumes that each of the near-synonyms was represented using the exact same pe-

¹⁵ By primitive concepts, we mean *named* concepts, or concepts that can be lexicalized by a single word, even though they may be defined in terms of other concepts in an ontology.

```

Diff ("bavure" / "mistake") =
  ( ( [usually / unknown] [medium / unknown] [implication / unknown]
      (Stupidity (ATTRIBUTE-OF V1)) )
    ( [always / sometimes] medium implication
      (Blameworthiness (ATTRIBUTE-OF V1) (DEGREE [more / ])) )
    ( always medium implication\
      (Criticism (ACTEE V1) (ATTRIBUTE (Severity (DEGREE [more / ])))) )
    ( [unknown / always] [unknown / medium] [unknown / implication]
      (Misconception (CAUSE-OF V2) (ACTOR V1)) )
    ( [unknown / always] [unknown / weak] [unknown / implication]
      (Accident (CAUSE-OF V2) (ACTOR V1)) )
    ( [always / unknown] [medium / unknown] [implication / unknown]
      (Unfortunate (ATTRIBUTE-OF ROOT)) )
    ( [usually / always] medium [pejorative / neutral] V1 )
    ( [more / ] concreteness ) )

```

Figure 9

A structure that explicitly represents the difference between *bavure* and *mistake*. The separate structures were merged, and where they differed, the two values are shown within square brackets separated by a /.

ripheral concepts to begin with (i.e., both with Blameworthiness rather than, say, one with Blameworthiness and the other with a closely related concept such as Responsibility). Can one even compare an error that is caused by a misconception to an error that is stupid? (See Figure 3 for *bavure*.)

When several dimensions are commensurate, how should one compute similarity? Consider the near-synonyms of *forest*: Is it possible to decide whether a “large and wild” tract of trees is closer to a “small wild” one or to a “medium-sized non-wild” one? In other words, how much of a change in the size of a forest will compensate for an opposite change in its wildness?

Part of the solution lies in the fact that the dimensions of any cluster are never actually completely incommensurate; usually they will have interrelationships that can be both modeled in the ontology and exploited when comparing the representations of words. For instance, in the cluster of near-synonyms of *forest*, the wildness of a tract of trees is related to its size and distance from civilization (which one can infer from one’s knowledge about forests and wildlife; e.g., most wildlife tries to avoid people); so there may be a way of comparing a “wild” tract of trees to a “large” tract of trees. And in the *error* cluster, the dimensions are related in similar ways because of their semantics and pragmatics (e.g., responsibility leads to blameworthiness, which often leads to criticism, and stupidity often leads to a pejorative attitude). Certainly these interrelationships influence both what can be coherently represented in a cluster and how similar near-synonyms are. And such relationships can be represented in the knowledge base, and hence reasoned about; a complete model, however, is out of the scope of this article.

The interaction of the dimensions within a cluster is not yet very well studied, so for a partial solution, we make the simplifying assumptions that the dimensions of a cluster are independent and that each can be reduced to a true numeric dimension.¹⁶

¹⁶ Certainly, numeric values are necessary at some level of representation. As we’ve seen, nuances of meaning and style are not always clear-cut but can be vague, fuzzy, and continuously variable. Using a numerical method would seem to be the most intuitive way of computing similarity, which we have to do to compare and choose appropriate lexical items.

Thus, two distinctions d_1 and d_2 are **commensurate** if the following two conditions hold:

- d_1 and d_2 are of the same type (i.e., stylistic, expressive, or denotational).
- If d_1 and d_2 are stylistic, then they involve the same stylistic dimension; if they are expressive, then they refer to the same entity; and if they are denotational, then they involve the same peripheral concept.

6.2 Computing the Similarity of Distinctions

Given our simplifications from above, a word's set of distinctions situates it in a *numeric* multidimensional space. Consider a function $Sim: D \times D \rightarrow [0, 1]$, for computing the similarity of two commensurate lexical distinctions taken from the set D of all possible distinctions that can be represented in a particular cluster. A value of 0 means that the distinctions are completely different (or can't even be compared), and a value of 1 means that they are equivalent (though not necessarily identical, as two equivalent distinctions might be structurally different).

Hence, each type of distinction requires its own similarity function:

$$Sim(d_1, d_2) = \begin{cases} 0.0 & \text{if } d_1 \text{ and } d_2 \text{ are not commensurate} \\ Sim_{denotational}(d_1, d_2) & \text{if } d_1 \text{ and } d_2 \text{ are denotational} \\ Sim_{expressive}(d_1, d_2) & \text{if } d_1 \text{ and } d_2 \text{ are expressive} \\ Sim_{stylistic}(d_1, d_2) & \text{if } d_1 \text{ and } d_2 \text{ are stylistic} \end{cases} \quad (1)$$

Each of the similarity functions must compare the values that the pair of distinctions has on each of their components (see Section 5.4). To arrive at a final numerical value, we must reduce each component to a real-valued dimension and assign each symbolic value for that component to a numeric position on the line. Edmonds (1999) gives complete details of the formulas we developed.

There is, however, a remaining interesting problem: How does one compute the degree of similarity of two conceptual structures? Denotational distinctions sometimes involve complex structures of concepts, and these structures must be somehow compared to determine their numeric degree of similarity. For instance, we might need to decide how similar a high degree of blameworthiness is to a moderate degree of blameworthiness, or to blameworthiness. Or, we might need to decide how similar official authority is to peremptory authority, or how similar arbitrary power is to peremptory authority (where arbitrariness is a kind of peremptoriness and authority is a kind of power). Computing this type of similarity is clearly different from, but related to, the problem of computing the similarity of primitive concepts (or words). We have to consider not only the content but also the structure of the representations.

We are not aware of any research on the general problem of computing the similarity of arbitrary conceptual structures, though some related work has been done in the area of description logics. Cohen, Borgida, and Hirsh (1992), for example, formalize a "least common subsumer" operation that returns the largest set of commonalities between two descriptions. And Resnik and Diab (2000) use a technique, attributed to Lin, of decomposing a structure into feature sets. Edmonds (1999) describes a technique for simultaneously traversing a pair of conceptual structures under the assumption that the structures will be "similar" because they are commensurate. Still, a good solution to this problem remains an open issue.

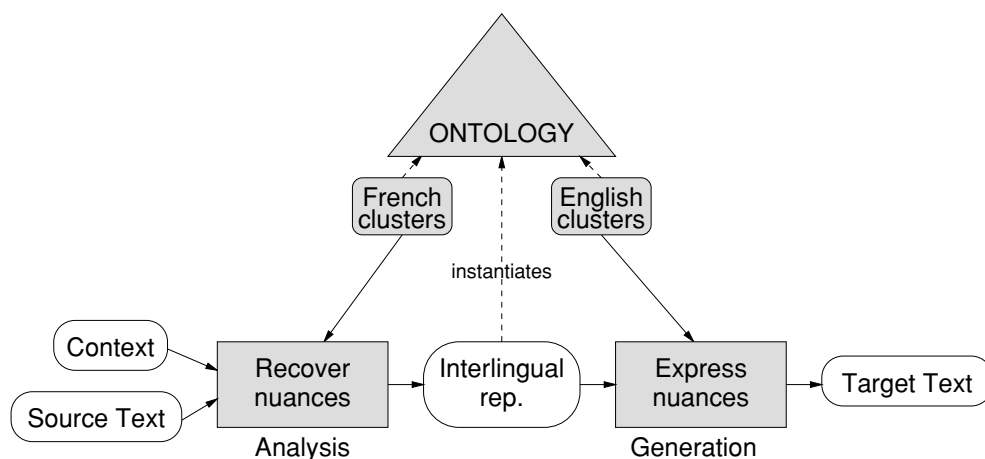


Figure 10
Lexical analysis and choice in machine translation.

7. Lexical Choice

7.1 Architectures for Lexical Choice

The clustered model of lexical knowledge is applicable to both the lexical-analysis and lexical-choice phases of a machine translation system. Figure 10 shows that during analysis, fine-grained lexical knowledge of the source language is accessed, in conjunction with the context, to determine **possibilities** of what is expressed in the source language text. Then, depending on the type of MT system (i.e., transfer or interlingual), the appropriate target language words can be chosen: The possibilities become **preferences** for choice. Recovering nuances of expression from source text is currently an open problem, which we do not explore further here (but see Edmonds [1998] for some preliminary work). In this section we concentrate on the second phase of MT and show that robust, efficient, flexible, and accurate fine-grained lexical choice is a natural consequence of a clustered model.

Lexical choice, as we see it, is more than a problem of mapping from concepts to words, as the previous section might have implied; it is a problem of selecting words so as to meet or satisfy a large set of possibly conflicting preferences to express certain nuances in certain ways, to establish the desired style, and to respect collocational and syntactic constraints. So lexical choice—genuine lexical choice—is making choices between options rather than merely finding the words for concepts, as was the case in many early text generation systems (for instance, BABEL [Goldman 1975], MUMBLE [McDonald 1983], and TEXT [McKeown 1985]). This kind of lexical choice is now thought to be the central task in text generation (or, at least, sentence generation), because it interacts with almost every other task involved. Indeed, many recent text generation systems, including MOOSE (Stede 1999), ADVISOR II (Elhadad, McKeown, and Robin 1997), and Hunter-Gatherer (Beale et al. 1998), among others (see Reiter and Dale's [1997] survey), adopt this view, yet their lexical-choice components do not account for near-synonymy. Without loss of generality, we will look at fine-grained lexical choice in the context of one of these systems: Stede's MOOSE (1999).

The input to MOOSE is a "SitSpec," that is, a specification of a situation represented on the conceptual-semantic level as a graph of instances of concepts linked

by relations. MOOSE outputs a complete well-formed “SemSpec,” or semantic specification on the syntactic–semantic level, from which the Penman sentence realization system can generate language.¹⁷ MOOSE processes the input in two stages. It first gathers all of the lexical items (as options for choice) whose conceptual–semantic representation covers any part of the SitSpec. Then it chooses a set of lexical items that satisfy Stede’s three criteria for sentence planning: the input SitSpec is completely covered (and so is completely lexicalized without redundancy); a well-formed SemSpec can be built out of the partial SemSpecs associated with each of the chosen lexical items; and as many of the preferences are satisfied as possible. MOOSE supports preferences, but only those that require structural decisions, such as choosing a causative over inchoative verb alternation. The main goal of Stede’s work was to account for structural paraphrase in sentence generation, not near-synonymy.

In the general case of sentence planning, given a set of input constraints and preferences, a sentence planner will make a large number of decisions of different types—lexical, syntactic, and structural—each of which has the potential to satisfy any, some, or all of the input preferences (while trying to satisfy the constraints, of course). It is unlikely that any particular set of preferences can all be satisfied simultaneously, so some kind of conflict resolution strategy is required in order to manage the decision-making task. It is not within the scope of this paper to develop solutions to this general problem (but see Nirenburg, Lesser, and Nyberg [1989], Wanner and Hovy [1996], Elhadad, McKeown, and Robin [1997], and Stede [1999] for a variety of solutions). Instead, we will discuss the following two new issues that arise in managing the interactions between lexical choices that a clustered model brings out:

- We will argue for a unified model for representing any type of preference for lexical choice.
- We describe a two-tiered model of lexical choice that is the consequence of a clustered model of lexical knowledge.

Then, we will end the section with a brief description of our software implementation of the model, called I-Saurus.

7.2 Constraints and Preferences

Simple systems for lexical choice need only to make sure that the denotations of the words chosen in response to a particular input exactly match the input. But when we use fine-grained aspects of meaning, the lexical-choice process, and so, in turn, its input, will ultimately be more complex. But with so many possibilities and options, choosing from among them will necessarily involve not only degrees of satisfying various criteria, but also trade-offs among different criteria. Some of the criteria will be hard **constraints** (i.e., a SitSpec), ensuring that the basic desired meaning is accurately conveyed, and others will be **preferences**.

The main difference between a constraint and a preference is that a preference is allowed to be satisfied to different degrees, or even not at all, depending on the decisions that are made during sentence planning. A preference can be satisfied by

¹⁷ A SemSpec is a fully lexicalized sentence plan in Penman’s Sentence Plan Language (SPL). SPL is defined in terms of the Penman Upper Model, a model of meaning at the syntactic–semantic level, which ensures that the SemSpec is well-formed linguistically. Penman can thus turn any SemSpec into a well-formed sentence without having to make any open-class lexical decisions (Penman Natural Language Group 1989; Stede 1999)

a single decision or collectively by a group of decisions.¹⁸ And because conflicts and trade-offs might arise in the satisfaction of several preferences at once, each preference must have an externally assigned importance factor.

Many types of preference pertain to lexical choice, including emphasizing an aspect of an entity in a situation, using normal words or a certain dialect, using words with a particular phonology (e.g., words that rhyme), using different near-synonyms for variety or the same word as before for consistency, and so on. All should be formalizable in a unified model of preference, but we have started with three types corresponding to the subconceptual level of the clustered model: denotational (or semantic), expressive, and stylistic preferences.

Denotational preferences are distinct from denotational constraints, but there is no theoretical difference in the nature of a “preferred” meaning to a “constrained” meaning. Hence, we can represent both in the same SitSpec formalism. Thus, a denotational preference is a tuple consisting of a partial SitSpec and a preferred method of expression, which takes a value on the continuum of indirectness (see Section 5.4). An **expressive preference** requests the expression of a certain attitude toward a certain entity that is part of the situation. Thus, an expressive preference is a tuple consisting of a reference to the entity and the stance that the system should take: favor, remain neutral, or disfavor. A **stylistic preference**, for now, is simply a value (of low, medium, or high) on one of the stylistic dimensions. We will see some examples in Section 8.

7.2.1 Satisfying Preferences by Lexical Choice. In the best case, it will be possible to simultaneously satisfy all of the input preferences by choosing appropriate near-synonyms from appropriate clusters. But if none of the available options will satisfy, to any degree, a particular preference, then that preference is trivially impossible to satisfy (by lexical choice). But even when there are options available that satisfy a particular preference, various types of conflicts can arise in trying to satisfy several preferences at once, making it impossible to use any of those options. At the level of clusters, for instance, in choosing a particular cluster in order to satisfy one preference, we might be therefore unable to satisfy another preference that can be satisfied only by a different, competing cluster: We might choose the cluster of the *err* verbs (*to err*, *to blunder*) because of the simplicity or directness of its syntax: *John erred*; but we would not be able simultaneously to satisfy a preference for implying a misconception by choosing, say, *mistake* from the cluster of *error* nouns: *John made a mistake*.

Similar trade-offs occur when choosing among the near-synonyms of the same cluster. Such lexical gaps, where no single word can satisfy all of the input preferences that a cluster can potentially satisfy, are common. For instance, in English, it’s hard to talk about a mistake without at least some overtones of criticism; in Japanese one can: with *ayamari* instead of *machigai* (Fujiwara, Isogai, and Muroyama 1985). There is also no near-synonym of *error* in English that satisfies preferences to imply both stupidity and misconception; *blunder* satisfies the former but not the latter, and *mistake* vice versa. Similarly, there is no formal word for an untrue statement (i.e., a *lie*) that also expresses that the lie is insignificant; *fib* is an option, but it is not a formal word. And there is no word for a tract of trees that is both large and not wild; *forest* has the former property, *woods* the latter.

18 A preference is like a **floating constraint** (Elhadad, McKeown, and Robin 1997) in that it can be satisfied by different types of decision in sentence planning but differs in that it may be satisfied to different degrees.

Two separate simultaneous choices might also conflict in their satisfaction of a single preference. That is, the preference might be satisfied by one choice and negatively satisfied by another choice. For instance, it might happen that one word is chosen in order to express a favorable attitude toward a participant in a particular situation and another word is chosen that inadvertently expresses a pejorative attitude toward the same person if that second word is chosen in order to satisfy some other preference. And of course, stylistic decisions can often conflict (e.g., if one has to choose both formal and informal words).

Our solution to resolving such lexical gaps and conflicting preferences is to use an approximate matching algorithm that attempts to satisfy collectively as many of the preferences as possible (each to the highest degree possible) by choosing, on two tiers, the right words from the right clusters.¹⁹ We will describe this model in Section 7.3.

7.2.2 Compatibility of Preferences. But what happens when it is impossible to simultaneously satisfy two preferences under any circumstances? We have assumed up to now that the set of preferences in the input is consistent or well-formed. This is often a reasonable assumption. In the context of MT, for instance, we can assume that a “good” analysis stage would output only well-formed expressions free of incompatibilities.

But two preferences may be **incompatible**, and we would like our system to be able to detect such situations. For instance, preferences for both low and high severity are incompatible; not only is it impossible for a word to simultaneously express both ideas, but if the system were to attempt to satisfy both, it might output a dissonant expression²⁰ such as “I (gently) chided Bill for his (careless) blunder” (the preference to harshly criticize Bill is satisfied by *blunder*, and the preference to gently criticize Bill is satisfied by *chide*). (Of course, a dissonant expression is not always undesirable; it might be used for special effect.) This kind of incompatibility is easy to detect in our formalism, because peripheral concepts are explicitly modeled as dimensions. There are, of course, other types of incompatibility, such as denotational and contextual incompatibilities, but we won’t discuss them further here (see Edmonds [1999]).

7.3 Two-Tiered Lexical Choice

Assume now that all of the options for choice have been identified by the system. In our system, these options are the clusters whose core denotations match part of the input SitSpec. Ignoring the coverage and well-formedness constraints for now, two different, mutually constraining types of decision must be made:

- Choosing from several competing cluster options.
- Choosing a near-synonym from a cluster option.

We believe that it is important to separate the processes for making these two types of decision—even though they must interact—because of their different choice criteria and effects. The former type involves choosing between options of differing coarse-grained semantic content and resulting syntactic structure (i.e., paraphrases): clusters

¹⁹ A complementary approach is to paraphrase the input and hence explicitly express a preferred implication or mitigate against an unwanted implication (for instance, by generating *insignificant lie* when *fib* is too informal). A sentence planner, like MOOSE, is designed to generate such structural paraphrases, so we have concentrated on the lexical issues here.

²⁰ Dissonance is one form of semantic anomaly that Cruse (1986) defines by example: “Arthur is a married bachelor.”

have different core denotations, after all. Here, issues of syntactic and semantic style are involved, as one can choose how the semantic content is to be incorporated. On the other hand, the latter type of decision involves options that might have subtle semantic and stylistic differences but result in the same syntactic structure (though collocational and subcategorization structure can vary).

In other words, lexical choice is a **two-tiered** process that must find both the appropriate set of cluster options and the appropriate set of lexical items (one from each chosen cluster option) whose contributing SemSpec fragments can be unified into a complete well-formed SemSpec. Of course, many possible SemSpecs can usually be generated, but the real problem is to find the combination of cluster options and lexical items that *globally* satisfy as many of the input preferences as possible.

For instance, Figure 11 depicts the state of processing the SitSpec for the utterance by John of an untrue statement just before lexical choice occurs. There are four cluster options (denoted by the suffix *_C*): *say_C* and *tell-a-lie_C* match subgraphs of the SitSpec rooted at *say1*, *untruth_C* matches the graph rooted at *lie1*, and *John_C* matches *john1*. Now, the system could choose the *tell-a-lie_C* cluster and the *John_C* cluster, which fully cover the SitSpec, and then choose the words *John* and *lie* to come up with *John lies*, or the system could choose *John* and *prevaricate* for *John prevaricates*. The system could also choose the *say_C*, *untruth_C* and *John_C* clusters, and then the words *tell*, *fib*, and *John*, to end up with *John tells a fib*. These alternatives—there are many others—are different in structure, meaning, and style. Which best satisfies the input preferences, whatever they may be?

We can formally define fine-grained lexical choice (within sentence planning) as follows. Given an input SitSpec S and a set of compatible preferences P , the goal is to find a set C of i cluster options and a word w_i from each $c_i \in C$ such that

- every node of S is covered by exactly one c_i
- the partial SemSpecs of all the words w_i can be combined into a well-formed SemSpec SP
- $Satisfaction(P, SP)$ is maximized over all possible SemSpecs

The first criterion ensures complete coverage without redundancy of the input SitSpec, so the desired meaning, at a coarse grain, is expressed; the second ensures that a SemSpec can be constructed that will lead to a grammatical sentence; and the third ensures that the preferences are collectively satisfied as much as is possible by any sentence plan. The third criterion concerns us here; the first two are dealt with in MOOSE.

As we said earlier, a thorough understanding of the interacting decisions in lexical choice is still an open problem, because it is context dependent. Our present solution is simply to assume that no complex interaction between decisions takes place. So, assuming that each option has an associated numeric score (the degree to which it satisfies all of the preferences), we can simply choose the set of options that maximizes the sum of the scores, subject to the other constraints of building a proper sentence plan. Thus, we do not provide a solution to how the context affects the combination of the scores. So, given a sentence plan SP and a set of preferences P , we have

$$Satisfaction(P, SP) = \sum_{w \in SP} WSat(P, w). \quad (2)$$

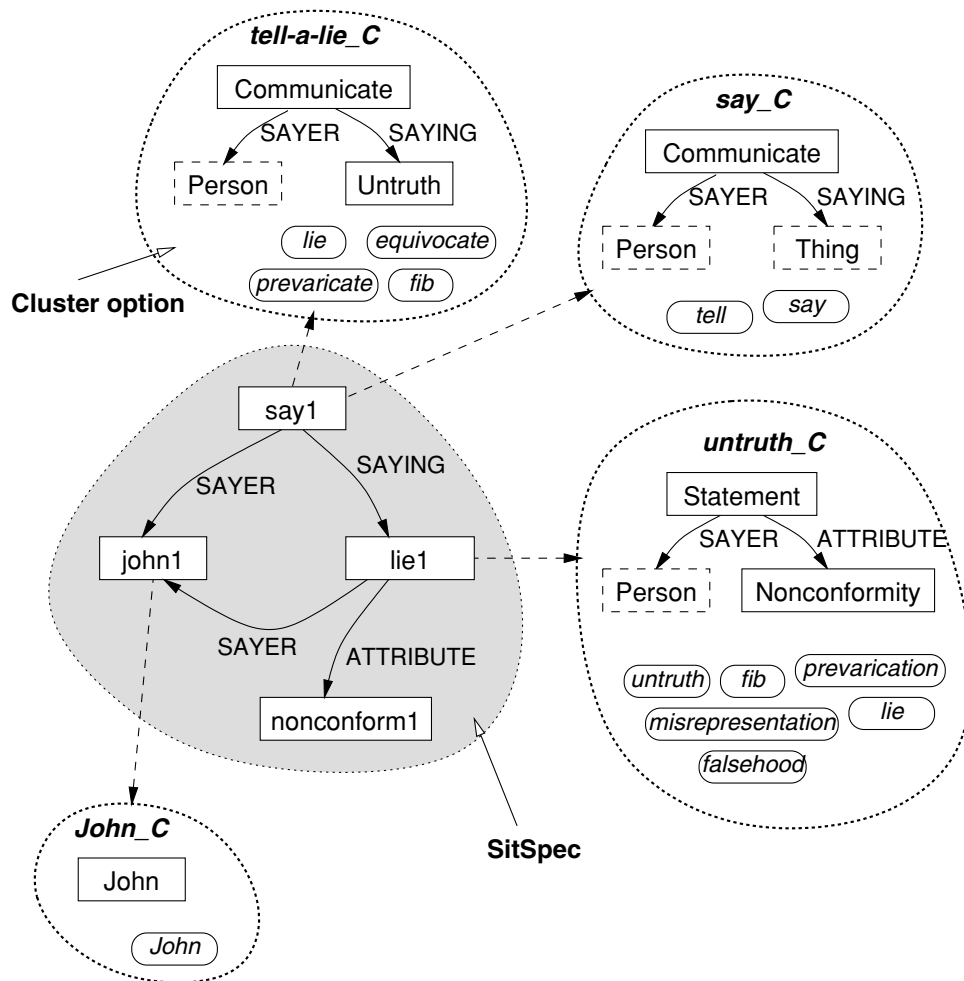


Figure 11

The state of processing just before lexical choice on the input for *John tells a lie*. Four clusters have become options; each is shown with its core denotation and near-synonyms. Solid arrows in the SitSpec indicate relations between instances of concepts. Solid arrows in the cluster options relate concepts in the core denotations. Dashed arrows link SitSpec nodes to the cluster options that cover subgraphs rooted at the nodes.

where $WSat$ is the degree to which w satisfies the preferences P (see Equation (3)). The most preferred sentence plan SP' is thus the one whose set of word choices maximizes $Satisfaction(P, SP')$. This function accounts for trade-offs in the satisfaction of preferences, because it finds the set of words that collectively satisfy as many of the preferences as possible, each to the highest degree possible.

Each word in SP has to be chosen from a distinct cluster. Thus, given

- a particular cluster c in the set of all cluster options
- a list W of candidate near-synonyms of c , ordered according to a prespecified criterion (some candidates of the cluster might have already been ruled out because of collocational constraints)

- a set $P_c \subset P$ of compatible preferences, each of which can potentially be satisfied by a word in c , with associated importances: $Imp: P \rightarrow [0, 1]$

find the first candidate $w' \in W$ such that $WSat(P, w')$ is maximized.

We use an approximate-matching algorithm to compute $WSat(P, w)$. Under the simplification that its value depends on the degree to which w individually satisfies each of the preferences in P_c , the algorithm computes $WSat(P, w)$ by combining the set of scores $Sat(p, w)$ for all $p \in P_c$. Various combination functions are plausible, including simple functions, such as a weighted average or a distance metric, and more complex functions that could, for instance, take into account dependencies between preferences.²¹ Deciding on this function is a subject for future research that will empirically evaluate the efficacy of various possibilities. For now, we define $WSat$ as a weighted average of the individual scores, taking into account the importance factors:

$$WSat(P, w) = WSat(P_c, w) = \sum_{p \in P_c} \frac{Imp(p)}{|P_c|} Sat(p, w) \quad (3)$$

For a given preference $p \in P_c$, the degree to which p is satisfied by w , $Sat(p, w)$, is reducible to the problem of computing similarity between lexical distinctions, for which we already have a solution (see Equation (1)). Thus,

$$Sat(p, w) = Sim(d(p), d(w)) \quad (4)$$

where $d(p)$ is a kind of **pseudo-distinction** generated from p to have the same form as a lexical distinction, putting it on equal footing to $d(w)$, and $d(w)$ is the distinction of w that is commensurate with $d(p)$, if one exists.

7.4 Implementation: I-Saurus

I-Saurus, a sentence planner that splits hairs, extends Stede's MOOSE (1999) with the modifications formalized above for fine-grained lexical choice. It takes a SitSpec and a set of preferences as input, and outputs a sentence plan in Penman's SPL, which Penman generates as a sentence in English. (Section 8 provides an example.)

Now, finding the best set of options could involve a lengthy search process. An exhaustive search through all possible sentence plans to find the one that maximizes $Satisfaction(P, SP)$ can be very time-inefficient: In the relatively small example given in Section 8, there are 960 different sentence plans to go through. To avoid an exhaustive search, we use the following heuristic, adopted from Stede (1999): In order to find the *globally* preferred sentence plan, make the most preferred *local* choices. That is, whenever a (local) decision is made between several options, choose the option with the highest score. Thus, we postulate that the most preferred sentence plan will be one of the first few sentence plans generated, though we offer no proof beyond our intuition that complex global effects are relatively rare, which is also a justification for the simplifications we made above.

Figure 12 gives an algorithm for two-tiered lexical choice embedded in MOOSE's sentence planner. The main additions are the procedures *Next-Best-Cluster-Option* and

²¹ For instance, we might want to consider a particular preference only after some other preference has been satisfied (or not) or only to resolve conflicts when several words satisfy another preference to the same degree.

```

Build-Sentence-Plan(node, P)
  (1)  $c \leftarrow \text{Next-Best-Cluster-Option}(\textit{node}, P)$ 
  if we've tried all the options then return "fail"
  (2)  $w \leftarrow \text{Next-Best-Near-Synonym}(c, P)$ 
  if we've tried all the near-synonyms in  $c$  then backtrack to (1)
   $p \leftarrow$  partial SemSpec of  $w$ 
  if  $p$  has external variables then
    for each external variable  $v$  in  $p$ 
       $s \leftarrow$  Build-Sentence-Plan(node bound to  $v$ ,  $P$ )
      if  $s =$  "fail" then
        backtrack to (2)
      else
        attach  $s$  to  $p$  at  $v$ 
  return  $p$ 

```

Figure 12

The sentence-planning algorithm. This algorithm outputs the most preferred complete well-formed SemSpec for a subgraph rooted at given node in the SitSpec.

Next-Best-Near-Synonym. (Note, however, that this version of the algorithm does not show how complete coverage or well-formedness is ensured.) *Next-Best-Cluster-Option* moves through the cluster options that cover part of the SitSpec rooted at *node* in order of preference. As we said above, structural decisions on this tier of lexical choice are outside the scope of this article, but we can assume that an algorithm will in due course be devised for ranking the cluster options according to criteria supplied in the input. (In fact, MOOSE can rank options according to preferences to foreground or background participants, in order to make them more or less salient, but this is only a start.) *Next-Best-Near-Synonym* steps through the near-synonyms for each cluster in order of preference as computed by $WSat(P, w)$.

7.5 Summary

The two-tiered lexical-choice algorithm (and sentence-planning algorithm) developed in this section is as efficient as any algorithm developed to date for a conventional model of lexical knowledge (without near-synonyms), because it can find the appropriate cluster or clusters just as easily as the latter can find a word; A cluster in our model corresponds to an individual word in the conventional model. And choosing a near-synonym from a cluster is efficient because there are normally only a few of them per cluster. The system does not have to search the entire lexicon. The full complexity of representing and using fine-grained lexical differences is partitioned into small clusters. The process is also robust, ensuring that the right meaning (at a coarse grain) is lexicalized even if a "poor" near-synonym is chosen in the end. And when the right preferences are specified in the input, the algorithm is accurate in its choice, attempting to meet as many preferences as possible while also satisfying the constraints.

8. Example

A formal evaluation of I-Saurus would require both a substantial lexicon of clusters and a large test suite of input data correlated with the desired output sentences. Building such a suite would be a substantial undertaking in itself. Barring this, we could

Table 3Four simultaneous preferences and the six candidates of the *untruth_C* cluster.

Preferences:

- 1 (imply (significance1 (ATTRIBUTE-OF lie1) (DEGREE low)))
- 2 (imply (intend1 (ACTOR john1) (ACTEE mislead1)))
- 3 (disfavor john1)
- 4 (low formality)

Preference	Candidate					
	<i>fib</i>	<i>lie</i>	<i>misrepresentation</i>	<i>untruth</i>	<i>prevarication</i>	<i>falsehood</i>
1 Insignificance	1.00	0.00	0.00	0.00	0.00	0.00
2 Deliberateness	0.50	1.00	0.75	0.25	0.75	0.00
3 Disfavor	0.50	0.63	0.50	0.50	0.50	0.50
4 Low formality	1.00	0.50	0.50	0.50	0.00	0.50
Total Score	3.00	2.13	1.75	1.25	1.25	1.00

Note: For each candidate, we show the satisfaction scores (*Sat*) for each individual preference and the total satisfaction scores (*WSat*): *fib* scores highest.

evaluate I-Saurus as an MT system, in terms of coverage and of quality (intelligibility, fidelity, and fluency). Unfortunately, I-Saurus is but a prototype with a small experimental lexicon, so we can only show by a few examples that it chooses the most appropriate words given a variety of input preferences.

Returning again the situation of John and his lie (Figure 11), consider the set of four simultaneous preferences shown in the top part of Table 3. The bottom part shows the scores for each candidate in the *untruth_C* cluster. If this cluster option were chosen, then I-Saurus would choose the noun *fib*, because it simultaneously and maximally satisfies all of the preferences as shown by the score of $WSat(\{1, 2, 3, 4\}, fib) = 3.00$. But note that if *fib* were not available, then the second-place *lie* would be chosen, leaving unsatisfied the preference to express insignificance.

Now, for a whole sentence, consider the SitSpec shown in Table 4. For this, I-Saurus can generate 960 different sentence plans, including plans that realize the sentences *John commands an alcoholic to lie* and *John orders a drunkard to tell a fib*. I-Saurus can be so prolific because of the many possible combinations of the near-synonyms of the six clusters involved: *John_C* (one near-synonym), *alcoholic_C* (ten near-synonyms), *order_C* (six near-synonyms), *say_C* (two near-synonyms), *untruth_C* (six near-synonyms), and *tell-a-lie_C* (four near-synonyms).

The bottom part of Table 4 shows the variety of output that is possible when each individual preference and various combinations of simultaneous preferences (cases i–x) are input to the system. (The numbered preferences are listed at the top of the table.) So for example, if we input preference 3 (high formality), the system outputs *John enjoins an inebriate to prevaricate*. The output appears stilted in some cases because no other parameters, such as desired verb tense, were given to Penman and because the system has no knowledge of collocational constraints. Of course, we could have defined many other preferences (and combinations of preferences), but we chose these particular ones in order to show some of the interesting interactions that occur among the cluster options during processing; they are not meant to be representative of what a user would normally ask of the system.

Consider, for instance, case iv. Here, one cluster can satisfy preference 6 (pejorative attitude), and another cluster can satisfy preference 10 (misconception), but neither

Table 4

A sample of output sentences of I-Saurus given an input SitSpec and various preferences and combinations of preferences (cases i–x).

SitSpec:

```
(order1 (SAYER john1)
  (SAYEE alcoholic1)
  (SAYING (perform1 (ACTOR alcoholic1)
    (ACTEE (tell1 (SAYER alcoholic1)
      (SAYING (lie1
        (ATTRIBUTE nonconform1))))))))))
```

Preferences:

```
1 (low formality)
2 (medium formality)
3 (high formality)
4 (high concreteness)
5 (favor alcoholic1)
6 (disfavor alcoholic1)
7 (imply (authority1 (ATTRIBUTE-OF john1) (ATTRIBUTE official1)))
8 (imply (authority1 (ATTRIBUTE-OF john1) (ATTRIBUTE peremptory1)))
9 (imply (significance1 (ATTRIBUTE-OF lie1) (DEGREE low)))
10 (imply (misconceive1 (ACTOR alcoholic1) (CAUSE-OF lie1)))
11 (imply (contradict2 (ACTOR lie1) (ATTRIBUTE categorical2)))
```

Case	Input preferences	Output
	None	John commands an alcoholic to lie.
	1	John commands a drunk to fib.
	2	John commands an alcoholic to lie.
	3	John enjoins an inebriate to prevaricate.
	4	John directs a drunkard to tell a lie.
	5	John commands a tippler to fib.
	6	John commands a drunk to lie.
	7	John commands an alcoholic to lie.
	8	John orders an alcoholic to lie.
	9	John commands an alcoholic to fib.
	10	John commands an alcoholic to tell an untruth.
	11	John commands an alcoholic to lie.
i	2, 4	John directs a drunkard to tell a lie.
ii	1, 9	John commands a drunk to fib.
iii	3, 6	John enjoins a drunkard to prevaricate.
iv	6, 10	John commands a drunkard to tell an untruth.
v	3, 9	John enjoins an inebriate to fib.
vi	3, 7, 9	John commands an inebriate to fib.
vii	3, 8, 9	John orders an inebriate to fib.
viii	3, 6, 8, 9	John orders a drunkard to fib.
ix	3, 5	John enjoins a tippler to tell a prevarication.
x	3, 5, 11	John enjoins a tippler to tell a prevarication.

cluster can satisfy both preferences on its own. So the system chooses *drunkard*, because it is pejorative, and *untruth*, because it implies a misconception. No other combination of choices from the two clusters could have simultaneously satisfied both preferences.

And finally, consider case v, which illustrates a clash in the satisfaction of one of the preferences. *Fib* is chosen despite the fact that it is informal, because it is the only word that implies an insignificant lie. But the system compensates by choosing two

other formal words: *enjoin* and *inebriate*. If we add a preference to this case to imply that John has official authority (case vi), then I-Saurus system chooses *command* instead of *enjoin*, further sacrificing high formality.

9. Related Work

Most computational work on near-synonymy has been motivated by *lexical mismatches* in machine translation (Kameyama et al. 1991). In interlingual MT, an intermediate representational scheme, such as an ontology in knowledge-based machine translation (KBMT) (Nirenburg et al. 1992), or lexical-conceptual structures in UNITRAN (Dorr 1993) is used in encoding lexical meaning (and all other meaning). But as we showed in Section 3, such methods don't work at the fine grain necessary for near-synonymy, despite their effectiveness at a coarse grain. To overcome these problems but retain the interlingual framework, Barnett, Mani, and Rich (1994) describe a method of generating natural-sounding text that is maximally close in meaning to the input interlingual representation. Like us, they define the notion of semantic closeness, but whereas they rely purely on denotational representations and (approximate) logical inference in addition to lexical features for relative naturalness, we explicitly represent fine-grained aspects on a subconceptual level and use constraints and preferences, which gives flexibility and robustness to the lexical-choice process. Viegas (1998), on the other hand, describes a preliminary solution that accounts for semantic vagueness and underspecification in a generative framework. Although her model is intended to account for near-synonymy, she does not explicitly discuss it.

Transfer-based MT systems use a bilingual lexicon to map words and expressions from one language to another. Lists, sometimes huge, of handcrafted language-pair-specific rules encode the knowledge to use the mapping (e.g., in SYSTRAN [Gerber and Yang 1997]). EuroWordNet (Vossen 1998) could be used in such a system. Its Inter-Lingual-Index provides a language-independent link between synsets in different languages and has an explicit relation, EQ_NEAR_SYNONYM, for relating synsets that are not directly equivalent across languages. But, as in individual WordNets, there is no provision for representing differences between near-synonyms.

In statistical MT, there would seem to be some promise for handling near-synonymy. In principle, a system could choose the near-synonym that is most probable given the source sentence and the target-language model. Near-synonymy seems to have been of little concern, however, in statistical MT research: The seminal researchers, Brown et al. (1990), viewed such variations as a matter of taste; in evaluating their system, two different translations of the same source that convey roughly the same meaning (perhaps with different words) are considered satisfactory translations. More recently, though, Foster, Isabelle, and Plamondon (1997) show how such a model can be used in interactive MT, and Langkilde and Knight (1998) in text generation. Such methods are unfortunately limited in practice, because it is too computationally expensive to go beyond a trigram model (only two words of context). Even if a statistical approach could account for near-synonymy, Edmonds (1997) showed that its strength is not in choosing the right word, but rather in determining which near-synonym is most typical or natural in a given context. So such an approach would not be so useful in goal-directed applications such as text generation, or even in sophisticated MT.

10. Conclusion

Every natural language processing system needs some sort of lexicon, and for many systems, the lexicon is the most important component. Yet, real natural language pro-

cessing systems today rely on a relatively shallow coverage of lexical phenomena, which unavoidably restricts their capabilities and thus the quality of their output. (Of course, shallow lexical semantics is a necessary starting point for a practical system, because it allows for broad coverage.) The research reported here pushes the lexical coverage of natural language systems to a deeper level.

The key to the clustered model of lexical knowledge is its subconceptual/stylistic level of semantic representation. By introducing this level between the traditional conceptual and syntactic levels, we have developed a new model of lexical knowledge that keeps the advantages of the conventional model—efficient paraphrasing, lexical choice (at a coarse grain), and mechanisms for reasoning—but overcomes its shortcomings concerning near-synonymy. The subconceptual/stylistic level is more expressive than the top level, yet it allows for tractable and efficient processing because it “partitions,” or isolates, the expressiveness (i.e., the non-truth-conditional semantics and fuzzy representations) in small clusters. The model reconciles fine-grained lexical knowledge with coarse-grained ontologies using the notion of granularity of representation.

The next stage in this work is to build a more extensive lexicon of near-synonym clusters than the few handwritten clusters that were built for the simple implementation described in this article. To this end, Inkpen and Hirst (2001a, 2001b) are developing a method to automatically build a clustered lexicon of 6,000 near-synonyms (1,000 clusters) from the machine-readable text of Hayakawa’s *Choose the Right Word* (1994).

Besides MT and NLG, we envision other applications of the model presented in this article. For instance, an interactive dictionary—an *intelligent thesaurus*—would actively help a person to find and choose the right word in any context. Rather than merely list possibilities, it would rank them according to the context and to parameters supplied by the user and would also explain potential effects of any choice, which would be especially useful in computer-assisted second-language instruction. Or the model could be applied in the automatic (post)editing of text in order to make the text conform to a certain stylistic standard or to make a text more readable or natural to a given audience.

We leave a number of open problems for another day, including recovering nuances from text (see Edmonds [1998] for a preliminary discussion); evaluating the effectiveness of the similarity measures; determining the similarity of conceptual structures; understanding the complex interaction of lexical and structural decisions during lexical choice; exploring the requirements for logical inference in the model; modeling other aspects of fine-grained meaning, such as emphasis; and understanding the context-dependent nature of lexical differences and lexical knowledge.

Appendix: An Example Representation: The *Error* Cluster

The following is the representation of the cluster of *error* nouns in our formalism. Tokens ending in `_1` represent lexical items. In upper case are either variables (for cross-reference) or relations; it should be clear from the context which is which. Capitalized tokens are concepts. In lower case are values of various features (such as “indirectness” and “strength”) defined in the model. We have not discussed many of the implementation details in this article, including `p-link` and `covers` (see Edmonds [1999]).

```
(defcluster error_C
  ;; from Gove (1984)
  :syns (error_1 mistake_1 blunder_1 slip_1 lapse_1 howler_1)
```

```

:core (ROOT Generic-Error)
:p-link ((V1 (:and (Person V1) (ACTOR ROOT V1)))
         (V2 (:and (Deviation V2) (ATTRIBUTE ROOT V2))))
:covers (ROOT)

:periph((P1 Stupidity (ATTRIBUTE-OF V1))
        (P2 Blameworthiness (ATTRIBUTE-OF V1))
        (P3 Criticism (ACTEE V1) (ATTRIBUTE (P31 Severity)))
        (P4 Misconception (CAUSE-OF V2) (ACTOR V1))
        (P5 Accident (CAUSE-OF V2) (ACTOR V1))
        (P6 Inattention (CAUSE-OF V2) (ACTOR V1)))

:distinctions (
;; Blunder commonly implies stupidity.
(blunder_l usually medium implication P1)

;; Mistake does not always imply blameworthiness, blunder sometimes.
(mistake_l sometimes medium implication (P2 (DEGREE 'medium)))
(error_l always medium implication (P2 (DEGREE 'medium)))
(blunder_l sometimes medium implication (P2 (DEGREE 'high)))

;; Mistake implies less severe criticism than error.
;; Blunder is harsher than mistake or error.
(mistake_l always medium implication (P31 (DEGREE 'low)))
(error_l always medium implication (P31 (DEGREE 'medium)))
(blunder_l always medium implication (P31 (DEGREE 'high)))

;; Mistake implies misconception.
(mistake_l always medium implication P4)

;; Slip carries a stronger implication of accident than mistake.
;; Lapse implies inattention more than accident.
(slip_l always medium implication P5)
(mistake_l always weak implication P5)
(lapse_l always weak implication P5)
(lapse_l always medium implication P6)

;; Blunder expresses a pejorative attitude towards the person.
(blunder_l always medium pejorative V1)

;; Blunder is a concrete word, error and mistake are abstract.
(blunder_l high concreteness)
(error_l low concreteness)
(mistake_l low concreteness)

;; Howler is an informal term
(howler_l low formality))
)

```

Acknowledgments

Our work is financially supported by the Natural Sciences and Engineering Research Council of Canada, the Ontario Graduate Scholarship program, and the University of Toronto. For discussions, suggestions, and comments on this work, we are grateful to Jack Chambers, Mark Chignell, Robert Dale, Chrysanne DiMarco, Paul Deane, Steve Green, Eduard Hovy, Brian Merrilees, John

Mylopoulos, Kazuko Nakajima, Sergei Nirenburg, Geoffrey Nunberg, Henry Schoght, Manfred Stede and the anonymous reviewers of *Computational Linguistics*.

References

Bailly, René. 1970. *Dictionnaire des synonymes de la langue française*. Librairie Larousse, Paris.

- Barnett, James, Inderjeet Mani, and Elaine Rich. 1994. Reversible machine translation: What to do when the languages don't match up. In Tomek Strzalkowski, editor, *Reversible Grammar in Natural Language Processing*. Kluwer Academic, pages 321–364.
- Barsalou, Lawrence W. 1992. Frames, concepts, and conceptual fields. In Adrienne Lehrer and Eva Fedder Kittay, editors, *Frames, Fields, and Contrasts: New Essays in Semantic and Lexical Organization*, pages 21–74. Lawrence Erlbaum.
- Batchelor, Ronald E. and Malcolm H. Offord. 1993. *Using French Synonyms*. Cambridge University Press.
- Beale, Stephen, Sergei Nirenburg, Evelyne Viegas, and Leo Wanner. 1998. De-constraining text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 48–57.
- Bénac, Henri. 1956. *Dictionnaire des synonymes*. Librairie Hachette, Paris.
- Brown, Peter F., John Cooke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.
- Budanitsky, Alexander. 1999. Measuring semantic relatedness and its applications. Master's thesis, technical report CSRG-390, Department of Computer Science, University of Toronto, Toronto, Canada. Available at <http://www.cs.toronto.edu/compling/Publications/Abstracts/Theses/Budanitsky-thabs.html>.
- Budanitsky, Alexander and Graeme Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources: Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 29–34, Pittsburgh.
- Budanitsky, Alexander and Graeme Hirst. 2002. Lexical semantic relatedness. Manuscript in preparation.
- Burkert, Gerrit and Peter Forster. 1992. Representation of semantic knowledge with term subsumption languages. In James Pustejovsky and Sabine Bergler, editor, *Lexical Semantics and Knowledge Representation: First SIGLEX Workshop*. Lecture Notes in Artificial Intelligence 627. Springer-Verlag, pages 75–85.
- Chapman, Robert L., editor. 1992. *Roget's International Thesaurus*. 5th edition. HarperCollins Publishers.
- Church, Kenneth Ward, William Gale, Patrick Hanks, Donald Hindle, and Rosamund Moon. 1994. Lexical substitutability. In B. T. S. Atkins and A. Zampolli, editors, *Computational Approaches to the Lexicon*. Oxford University Press, pages 153–177.
- Clark, Eve V. 1992. Conventionality and contrast: Pragmatic principles with lexical consequences. In Adrienne Lehrer and Eva Fedder Kittay, editors, *Frames, Fields, and Contrasts: New Essays in Semantic and Lexical Organization*. Lawrence Erlbaum, pages 171–188.
- Cohen, William W., Alex Borgida, and Haym Hirsh. 1992. Computing least common subsumers in description logic. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 754–760.
- Coleman, Linda and Paul Kay. 1981. Prototype semantics: The English word *lie*. *Language*, 57(1):26–44.
- Cruse, D. Alan. 1986. *Lexical Semantics*. Cambridge University Press.
- Dagan, Ido, Shaul Marcus, and Shaul Markovitch. 1993. Contextual word similarity and estimation from sparse data. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 164–171.
- DiMarco, Chrysanne and Graeme Hirst. 1993. A computational theory of goal-directed style in syntax. *Computational Linguistics*, 19(3):451–500.
- DiMarco, Chrysanne, Graeme Hirst, and Manfred Stede. 1993. The semantic and stylistic differentiation of synonyms and near-synonyms. In *AAAI Spring Symposium on Building Lexicons for Machine Translation*, pages 114–121, Stanford, CA, March.
- Dorr, Bonnie J. 1993. *Machine Translation: A View from the Lexicon*. MIT Press.
- Edmonds, Philip. 1997. Choosing the word most typical in context using a lexical co-occurrence network. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 507–509, Madrid, Spain.
- Edmonds, Philip. 1998. Translating near-synonyms: Possibilities and preferences in the interlingua. In *Proceedings of the AMTA/SIG-IL Second Workshop on Interlinguas*, pages 23–30, Langhorne, PA. (Proceedings published as technical report MCCS-98-316, Computing Research Laboratory, New Mexico State University.)

- Edmonds, Philip. 1999. *Semantic Representations of Near-Synonyms for Automatic Lexical Choice*. Ph.D. thesis, Department of Computer Science, University of Toronto. Available at <http://www.cs.toronto.edu/compling/Publications/Abstracts/Theses/EdmondsPhD-thabs.html>.
- Egan, Rose F. 1942. "Survey of the history of English synonymy" and "Synonym: Analysis and definition." Reprinted in Philip B. Gove, editor, *Webster's New Dictionary of Synonyms*. Merriam-Webster, Springfield, MA, pp. 5a–31a.
- Elhadad, Michael, Kathleen McKeown, and Jacques Robin. 1997. Floating constraints in lexical choice. *Computational Linguistics*, 23(2):195–240.
- Emele, Martin, Ulrich Heid, Stefan Momma, and Rémi Zajac. 1992. Interactions between linguistic constraints: Procedural vs. declarative approaches. *Machine Translation*, 7(1–2):61–98.
- Evens, Martha, editor. 1988. *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*. Cambridge University Press.
- Farrell, Ralph Barstow. 1977. *Dictionary of German Synonyms*. 3rd edition. Cambridge University Press.
- Fernald, James C., editor. 1947. *Funk & Wagnall's Standard Handbook of Synonyms, Antonyms, and Prepositions*. Funk & Wagnall's, New York.
- Foster, George, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation*, 12:175–194.
- Frege, Gottlob. 1892. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und Philosophie Kritik*, 100:25–50. English translation: On sense and reference. In P. Geach and M. Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*. Blackwell, 1960.
- Fujiwara, Yoichi, Hideo Isogai, and Toshiaki Muroyama. 1985. *Hyogen ruigo jiten*. Tokyodo Shuppan, Tokyo.
- Gentner, Dedre and Arthur B. Markman. 1994. Structural alignment in comparison: No difference without similarity. *Psychological Science*, 5(3):152–158.
- Gerber, Laurie, and Jin Yang. 1997. SYSTRAN MT dictionary development. In *Machine Translation: Past, Present, and Future: Proceedings of Machine Translation Summit VI*, pages 211–218, San Diego, CA.
- Goldman, Neil M. 1975. Conceptual generation. In Roger C. Schank, editor, *Conceptual Information Processing*. North-Holland, Amsterdam, pages 289–371.
- Goodman, Nelson. 1952. On likeness of meaning. In L. Linsky, editor, *Semantics and the Philosophy of Language*. University of Illinois Press, pages 67–74.
- Gove, Philip B., editor. 1984. *Webster's New Dictionary of Synonyms*. Merriam-Webster, Springfield, MA.
- Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.
- Hayakawa, S. I., editor. 1994. *Choose the Right Word: A Contemporary Guide to Selecting the Precise Word for Every Situation*. 2nd edition, revised by Eugene Ehrlich. HarperCollins Publishers, New York.
- Hirst, Graeme. 1995. Near-synonymy and the structure of lexical knowledge. In *AAAI Symposium on Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, pages 51–56, Stanford, CA, March.
- Hovy, Eduard. 1988. *Generating Natural Language Under Pragmatic Constraints*. Lawrence Erlbaum Associates.
- Inkpen, Diana Zaiu and Graeme Hirst. 2001a. Experiments on extracting knowledge from a machine-readable dictionary of synonym differences. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing (Proceedings, Second Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February 2001)*, Lecture Notes in Computer Science 2004. Springer-Verlag, pages 264–278.
- Inkpen, Diana Zaiu and Graeme Hirst. 2001b. Building a lexical knowledge-base of near-synonym differences. *Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 47–52, Pittsburgh.
- Jackendoff, Ray. 1983. *Semantic and Cognition*. MIT Press.
- Jackendoff, Ray. 1990. *Semantic Structures*. MIT Press.
- Jiang, Jay J. and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference for Research on Computational Linguistics (ROCLING X)*, Taiwan.
- Kameyama, Megumi, Ryo Ochitani, Stanley Peters, and Hidetoshi Sirai. 1991. Resolving translation mismatches with information flow. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 193–200.
- Katz, Jerrold J. and Jerry A. Fodor. 1963. The structure of a semantic theory.

- Language*, 39:170–210.
- Kay, Mairé Weir, editor. 1988. *Webster's Collegiate Thesaurus*. Merriam-Webster, Springfield, MA.
- Kay, Paul. 1971. Taxonomy and semantic contrast. *Language*, 47(4):866–887.
- Kozima, Hideki and Teiji Furugori. 1993. Similarity between words computed by spreading activation on an English dictionary. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 232–239, Utrecht, Netherlands.
- Kroll, Judith F. and Annette M.B. de Groot. 1997. Lexical and conceptual memory in the bilingual: Mapping form to meaning in two languages. In Annette M.B. de Groot and Judith F. Kroll, editors, *Tutorials in Bilingualism: Psycholinguistic Perspectives*. Lawrence Erlbaum, pages 169–199.
- Langkilde, Irene and Kevin Knight. 1998. The practical value of n-grams in generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 248–255, Niagara-on-the-Lake, Canada.
- Lehrer, Adrienne and Eva Feder Kittay. 1992. Introduction. In Adrienne Lehrer and Eva Feder Kittay, editors, *Frames, Fields, and Contrasts: New Essays in Semantic and Lexical Organization*. Lawrence Erlbaum, pages 1–20.
- Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.
- Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL-98)*, pages 768–774, Montreal.
- Lyons, John. 1977. *Semantics*. Cambridge University Press.
- Lyons, John. 1995. *Linguistic Semantics: An Introduction*. Cambridge University Press.
- Markman, Arthur B. and Dedre Gentner. 1993. Splitting the differences: A structural alignment view of similarity. *Journal of Memory and Language*, 32:517–535.
- McDonald, David D. 1983. Description directed control: Its implications for natural language generation. In Nick Cercone, editor, *Computational Linguistics*, International Series in Modern Applied Mathematics and Computer Science 5. Plenum Press, New York, pages 111–129. Reprinted in B. J. Grosz, K. Sparck Jones, and B. L. Webber, editors, *Readings in Natural Language Processing*. Morgan Kaufmann, 1986, pages 519–537.
- McKeown, Kathleen R. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.
- McMahon, John G. and Francis J. Smith. 1996. Improving statistical language model performance with automatically generated word hierarchies. *Computational Linguistics*, 22(2):217–248.
- Nirenburg, Sergei, Jaime Carbonell, Masaru Tomita, and Kenneth Goodman. 1992. *Machine Translation: A Knowledge-Based Approach*. Morgan Kaufmann.
- Nirenburg, Sergei and Christine Defrise. 1992. Application-oriented computational semantics. In Michael Rosner and Roderick Johnson, editors, *Computational Linguistics and Formal Semantics*. Cambridge University Press, pages 223–256.
- Nirenburg, Sergei, Victor Lesser, and Eric Nyberg. 1989. Controlling a language generation planner. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1524–1530.
- Nirenburg, Sergei and Lori Levin. 1992. Syntax-driven and ontology-driven lexical semantics. In James Pustejovsky and Sabine Bergler, editors, *Lexical Semantics and Knowledge Representation: First SIGLEX Workshop*. Lecture Notes in Artificial Intelligence 627. Springer-Verlag, pages 5–20.
- Nogier, Jean-François and Michael Zock. 1992. Lexical choice as pattern matching. *Knowledge-Based Systems*, 5:200–212.
- Penman Natural Language Group. 1989. *The Penman reference manual*. Technical report, Information Sciences Institute of the University of Southern California.
- Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190.
- Pustejovsky, James. 1995. *The Generative Lexicon*. MIT Press.
- Pustejovsky, James and Sabine Bergler, editors. 1992. *Lexical Semantics and Knowledge Representation: First SIGLEX Workshop*. Lecture Notes in Artificial Intelligence 627. Springer-Verlag.
- Quine, W. V. O. 1951. Two dogmas of empiricism. *Philosophical Review*, 60:20–43.
- Reiter, Ehud and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–88.
- Resnik, Philip. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th*

- International Joint Conference on Artificial Intelligence*, pages 448–453, Montreal.
- Resnik, Philip and Mona Diab. 2000. Measuring verb similarity. In *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society (COGSCI 2000)*.
- Resnik, Philip, and David Yarowsky. 1999. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2):135–146.
- Room, Adrian. 1985. *Dictionary of Confusing Words and Meanings*. Dorset, New York.
- Rosch, Eleanor. 1978. Principles of categorization. In Eleanor Rosch and Barbara B. Lloyd, editors, *Cognition and categorization*. Lawrence Erlbaum Associates, pages 27–48.
- Saussure, Ferdinand de. 1916. *Cours de linguistique générale*. Translated by Roy Harris as *Course in General Linguistics*, London: G. Duckworth, 1983.
- Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.
- Sowa, John F. 1988. Using a lexicon of canonical graphs in a semantic interpreter. In Martha Evens, editor, *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*. Cambridge University Press, pages 113–137.
- Sowa, John F. 1992. Logical structures in the lexicon. In James Pustejovsky and Sabine Bergler, editors, *Lexical Semantics and Knowledge Representation: First SIGLEX Workshop*. Lecture Notes in Artificial Intelligence 627. Springer-Verlag, pages 39–60.
- Sparck Jones, Karen. 1986. *Synonymy and Semantic Classification*. Edinburgh University Press.
- Stede, Manfred. 1993. Lexical choice criteria in language generation. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, pages 454–459, Utrecht, Netherlands.
- Stede, Manfred. 1999. *Lexical Semantics and Knowledge Representation in Multilingual Text Generation*. Kluwer Academic.
- Tarski, Alfred. 1944. The semantic conception of truth. *Philosophy and Phenomenological Research*, 4:341–375.
- Ullmann, Stephen. 1962. *Semantics: An Introduction to the Science of Meaning*. Blackwell.
- Urdang, Laurence. 1992. *Dictionary of Differences*. Bloomsbury, London.
- Viegas, Evelyne. 1998. Multilingual computational semantic lexicons in action: The WYSINWYG approach to NLG. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL-98)*, pages 1321–1327, Montreal, Canada.
- Vossen, Piek. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic.
- Walther, George. 1992. *Power Talking: 50 Ways to Say What You Mean and Get What You Want*. Berkley, New York.
- Wanner, Leo and Eduard Hovy. 1996. The HealthDoc sentence planner. In *Proceedings of the Eighth International Workshop on Natural Language Generation*, pages 1–10.
- Wittgenstein, Ludwig. 1953. *Philosophical Investigations*. Blackwell.