

CHAPTER

C

WordNet: Word Relations, Senses, and Disambiguation

In this chapter we introduce computation with a thesaurus: a structured list of words organized by meaning. The most popular thesaurus for computational purposes is **WordNet**, a large online resource with versions in many languages. One use of WordNet is to represent **word senses**, the many different meanings that a single lemma can have (Chapter 6) Thus the lemma *bank* can refer to a financial institution or to the sloping side of a river. WordNet also represents relations between senses, like the **IS-A** relation between *dog* and *mammal* or the part-whole relationship between *car* and *engine*. Finally, WordNet includes **glosses**, a definition for senses in the form of a text string.

We'll see how to use each of these aspects of WordNet to address the task of computing **word similarity**; the similarity in meaning of two different words, an alternative to the embedding-based methods we introduced in Chapter 6. And we'll introduce **word sense disambiguation**, the task of determining which sense of a word is being used in a particular context, a task with a long history in computational linguistics and applications from machine translation to question answering. We give a number of algorithms for using features from the context for deciding which sense was intended in a particular context.

C.1 Word Senses

Consider the two uses of the lemma *bank* mentioned above, meaning something like “financial institution” and “sloping mound”, respectively:

- (C.1) Instead, a *bank* can hold the investments in a custodial account in the client's name.
- (C.2) But as agriculture burgeons on the east *bank*, the river will shrink even more.

word sense

We represent this variation in usage by saying that the lemma *bank* has two **senses**.¹ A **sense** (or **word sense**) is a discrete representation of one aspect of the meaning of a word. Loosely following lexicographic tradition, we represent each sense by placing a superscript on the lemma as in **bank¹** and **bank²**.

Homonym
Homonymy

homographs

The senses of a word might not have any particular relation between them; it may be almost coincidental that they share an orthographic form. For example, the *financial institution* and *sloping mound* senses of *bank* seem relatively unrelated. In such cases we say that the two senses are **homonyms**, and the relation between the senses is one of **homonymy**. Thus **bank¹** (“financial institution”) and **bank²** (“sloping mound”) are homonyms, as are the sense of *bat* meaning ‘club for hitting a ball’ and the one meaning ‘nocturnal flying animal’. We say that these two uses of *bank* are **homographs**, as are the two uses of *bat*, because they are written the

¹ Confusingly, the word “lemma” is itself ambiguous; it is also sometimes used to mean these separate senses, rather than the citation form of the word. You should be prepared to see both uses in the literature.

2 APPENDIX C • WORDNET: WORD RELATIONS, SENSES, AND DISAMBIGUATION

same. Two words can be homonyms in a different way if they are spelled differently but pronounced the same, like *write* and *right*, or *piece* and *peace*. We call these **homophones**; they are one cause of real-word spelling errors.

homophones Homonymy causes problems in other areas of language processing as well. In question answering or information retrieval, we better help a user who typed “bat care” if we know whether they are vampires or just want to play baseball. And they will also have different translations; in Spanish the animal bat is a *murciélagos* while the baseball bat is a *bate*. **Homographs** that are pronounced differently cause problems for speech synthesis (Chapter 28) such as these homographs of the word *bass*, the fish pronounced *b ae s* and the instrument pronounced *b ey s*.

- (C.3) The expert angler from Dora, Mo., was fly-casting for **bass** rather than the traditional trout.
- (C.4) The curtain rises to the sound of angry dogs baying and ominous **bass** chords sounding.

Sometimes there is also some semantic connection between the senses of a word. Consider the following example:

- (C.5) While some banks furnish blood only to hospitals, others are less restrictive.

Although this is clearly not a use of the “sloping mound” meaning of *bank*, it just as clearly is not a reference to a charitable giveaway by a financial institution. Rather, *bank* has a whole range of uses related to repositories for various biological entities, as in *blood bank*, *egg bank*, and *sperm bank*. So we could call this “biological repository” sense **bank³**. Now this new sense **bank³** has some sort of relation to **bank¹**; both **bank¹** and **bank³** are repositories for entities that can be deposited and taken out; in **bank¹** the entity is monetary, whereas in **bank³** the entity is biological.

polysemy When two senses are related semantically, we call the relationship between them **polysemy** rather than homonymy. In many cases of polysemy, the semantic relation between the senses is systematic and structured. For example, consider yet another sense of *bank*, exemplified in the following sentence:

- (C.6) The bank is on the corner of Nassau and Witherspoon.

This sense, which we can call **bank⁴**, means something like “the building belonging to a financial institution”. It turns out that these two kinds of senses (an organization and the building associated with an organization) occur together for many other words as well (*school*, *university*, *hospital*, etc.). Thus, there is a systematic relationship between senses that we might represent as

BUILDING ↔ ORGANIZATION

metonymy This particular subtype of polysemy relation is often called **metonymy**. Metonymy is the use of one aspect of a concept or entity to refer to other aspects of the entity or to the entity itself. Thus, we are performing metonymy when we use the phrase *the White House* to refer to the administration whose office is in the White House. Other common examples of metonymy include the relation between the following pairings of senses:

Author (<i>Jane Austen wrote Emma</i>) ↔ Works of Author (<i>I really love Jane Austen</i>)
Tree (<i>Plums have beautiful blossoms</i>) ↔ Fruit (<i>I ate a preserved plum yesterday</i>)

While it can be useful to distinguish polysemy from unrelated homonymy, there is no hard threshold for how related two senses must be to be considered polysemous. Thus, the difference is really one of degree. This fact can make it very difficult to decide how many senses a word has, that is, whether to make separate senses for

closely related usages. There are various criteria for deciding that the differing uses of a word should be represented with discrete senses. We might consider two senses discrete if they have independent truth conditions, different syntactic behavior, and independent sense relations, or if they exhibit antagonistic meanings.

Consider the following uses of the verb *serve* from the WSJ corpus:

- (C.7) They rarely *serve* red meat, preferring to prepare seafood.
- (C.8) He *served* as U.S. ambassador to Norway in 1976 and 1977.
- (C.9) He might have *served* his time, come out and led an upstanding life.

The *serve* of *serving red meat* and that of *serving time* clearly have different truth conditions and presuppositions; the *serve* of *serve as ambassador* has the distinct subcategorization structure *serve as NP*. These heuristics suggest that these are probably three distinct senses of *serve*. One practical technique for determining if two senses are distinct is to conjoin two uses of a word in a single sentence; this kind of conjunction of antagonistic readings is called **zeugma**. Consider the following ATIS examples:

- (C.10) Which of those flights serve breakfast?
- (C.11) Does Midwest Express serve Philadelphia?
- (C.12) ?Does Midwest Express serve breakfast and Philadelphia?

We use (?) to mark those examples that are semantically ill-formed. The oddness of the invented third example (a case of zeugma) indicates there is no sensible way to make a single sense of *serve* work for both breakfast and Philadelphia. We can use this as evidence that *serve* has two different senses in this case.

Dictionaries tend to use many fine-grained senses so as to capture subtle meaning differences, a reasonable approach given that the traditional role of dictionaries is aiding word learners. For computational purposes, we often don't need these fine distinctions, so we may want to group or cluster the senses; we have already done this for some of the examples in this chapter.

How can we define the meaning of a word sense? We introduced in Chapter 6 the standard computational approach of representing a word as an **embedding**, a point in semantic space. The intuition was that words were defined by their co-occurrences, the counts of words that often occur nearby.

Thesauri offer an alternative way of defining words. But we can't just look at the definition itself. Consider the following fragments from the definitions of *right*, *left*, *red*, and *blood* from the *American Heritage Dictionary* (Morris, 1985).

<i>right</i>	<i>adj.</i> located nearer the right hand esp. being on the right when facing the same direction as the observer.
<i>left</i>	<i>adj.</i> located nearer to this side of the body than the right.
<i>red</i>	<i>n.</i> the color of blood or a ruby.
<i>blood</i>	<i>n.</i> the red liquid that circulates in the heart, arteries and veins of animals.

Note the circularity in these definitions. The definition of *right* makes two direct references to itself, and the entry for *left* contains an implicit self-reference in the phrase *this side of the body*, which presumably means the *left* side. The entries for *red* and *blood* reference each other in their definitions. Such circularity is inherent in all dictionary definitions. For humans, such entries are still useful since the user of the dictionary has sufficient grasp of these other terms.

For computational purposes, one approach to defining a sense is—like the dictionary definitions—defining a sense through its relationship with other senses. For

zeugma

4 APPENDIX C • WORDNET: WORD RELATIONS, SENSES, AND DISAMBIGUATION

example, the above definitions make it clear that *right* and *left* are similar kinds of lemmas that stand in some kind of alternation, or opposition, to one another. Similarly, we can glean that *red* is a color, that it can be applied to both *blood* and *rubies*, and that *blood* is a *liquid*. **Sense relations** of this sort are embodied in on-line databases like **WordNet**. Given a sufficiently large database of such relations, many applications are quite capable of performing sophisticated semantic tasks (even if they do not *really* know their right from their left).

C.1.1 Relations Between Senses

This section explores some of the relations that hold among word senses, focusing on a few that have received significant computational investigation: **synonymy**, **antonymy**, and **hyponymy**, as well as a brief mention of other relations like **meronymy**.

Synonymy We introduced in Chapter 6 the idea that when two senses of two different words (lemmas) are identical, or nearly identical, we say the two senses are **synonyms**. Synonyms include such pairs as

couch/sofa vomit/throw up filbert/hazelnut car/automobile

And we mentioned that in practice, the word *synonym* is commonly used to describe a relationship of approximate or rough synonymy. But furthermore, synonymy is actually a relationship between senses rather than words. Considering the words *big* and *large*. These may seem to be synonyms in the following ATIS sentences, since we could swap *big* and *large* in either sentence and retain the same meaning:

- (C.13) How big is that plane?
(C.14) Would I be flying on a large or small plane?

But note the following WSJ sentence in which we cannot substitute *large* for *big*:

- (C.15) Miss Nelson, for instance, became a kind of big sister to Benjamin.
(C.16) ?Miss Nelson, for instance, became a kind of large sister to Benjamin.

This is because the word *big* has a sense that means being older or grown up, while *large* lacks this sense. Thus, we say that some senses of *big* and *large* are (nearly) synonymous while other ones are not.

hyponym **Hyponymy** One sense is a **hyponym** of another sense if the first sense is more specific, a subclass. For example, *car* is a hyponym of *vehicle*; *dog* is a hyponym of *animal*, and *mango* is a hyponym of *fruit*. Conversely, *vehicle* is a **hypernym** of *car*, and *animal* is a hypernym of *dog*. It is unfortunate that the two words hypernym and hyponym are very similar and hence easily confused; for this reason, the word **superordinate** is often used instead of **hypernym**.

Superordinate	vehicle	fruit	furniture	mammal
Hyponym	car	mango	chair	dog

meronymy **part-whole** **meronym** **holonym** **Meronymy** Another common relation is **meronymy**, the **part-whole** relation. A *leg* is part of a *chair*; a *wheel* is part of a *car*. We say that *wheel* is a **meronym** of *car*, and *car* is a **holonym** of *wheel*.

C.2 WordNet: A Database of Lexical Relations

WordNet The most commonly used resource for English sense relations is the **WordNet** lexical database (Fellbaum, 1998). WordNet consists of three separate databases, one each for nouns and verbs and a third for adjectives and adverbs; closed class words are not included. Each database contains a set of lemmas, each one annotated with a set of senses. The WordNet 3.0 release has 117,798 nouns, 11,529 verbs, 22,479 adjectives, and 4,481 adverbs. The average noun has 1.23 senses, and the average verb has 2.16 senses. WordNet can be accessed on the Web or downloaded and accessed locally. Figure C.1 shows the lemma entry for the noun and adjective *bass*.

<p>The noun “bass” has 8 senses in WordNet.</p> <ol style="list-style-type: none"> 1. bass¹ - (the lowest part of the musical range) 2. bass², bass part¹ - (the lowest part in polyphonic music) 3. bass³, basso¹ - (an adult male singer with the lowest voice) 4. sea bass¹, bass⁴ - (the lean flesh of a saltwater fish of the family Serranidae) 5. freshwater bass¹, bass⁵ - (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus)) 6. bass⁶, bass voice¹, basso² - (the lowest adult male singing voice) 7. bass⁷ - (the member with the lowest range of a family of musical instruments) 8. bass⁸ - (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes) <p>The adjective “bass” has 1 sense in WordNet.</p> <ol style="list-style-type: none"> 1. bass¹, deep⁶ - (having or denoting a low vocal or instrumental range) “<i>a deep voice</i>”; “<i>a bass voice is lower than a baritone voice</i>”; “<i>a bass clarinet</i>”

Figure C.1 A portion of the WordNet 3.0 entry for the noun *bass*.

gloss Note that there are eight senses for the noun and one for the adjective, each of which has a **gloss** (a dictionary-style definition), a list of synonyms for the sense, and sometimes also usage examples (shown for the adjective sense). Unlike dictionaries, WordNet doesn’t represent pronunciation, so doesn’t distinguish the pronunciation [b ae s] in **bass⁴**, **bass⁵**, and **bass⁸** from the other senses pronounced [b ey s].

synset The set of near-synonyms for a WordNet sense is called a **synset** (for **synonym set**); synsets are an important primitive in WordNet. The entry for *bass* includes synsets like {*bass¹*, *deep⁶*}, or {*bass⁶*, *bass voice¹*, *basso²*}. We can think of a synset as representing a concept of the type we discussed in Chapter 14. Thus, instead of representing concepts in logical terms, WordNet represents them as lists of the word senses that can be used to express the concept. Here’s another synset example:

{chump¹, fool², gull¹, mark⁹, patsy¹, fall guy¹, sucker¹, soft touch¹, mug²}

The gloss of this synset describes it as *a person who is gullible and easy to take advantage of*. Each of the lexical entries included in the synset can, therefore, be used to express this concept. Synsets like this one actually constitute the senses associated with WordNet entries, and hence it is synsets, not wordforms, lemmas, or individual senses, that participate in most of the lexical sense relations in WordNet.

WordNet represents all the kinds of sense relations discussed in the previous section, as illustrated in Fig. C.2 and Fig. C.3. WordNet hyponymy relations correspond

6 APPENDIX C • WORDNET: WORD RELATIONS, SENSES, AND DISAMBIGUATION

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	<i>breakfast</i> ¹ → <i>meal</i> ¹
Hyponym	Subordinate	From concepts to subtypes	<i>meal</i> ¹ → <i>lunch</i> ¹
Instance Hypernym	Instance	From instances to their concepts	<i>Austen</i> ¹ → <i>author</i> ¹
Instance Hyponym	Has-Instance	From concepts to concept instances	<i>composer</i> ¹ → <i>Bach</i> ¹
Member Meronym	Has-Member	From groups to their members	<i>faculty</i> ² → <i>professor</i> ¹
Member Holonym	Member-Of	From members to their groups	<i>copilot</i> ¹ → <i>crew</i> ¹
Part Meronym	Has-Part	From wholes to parts	<i>table</i> ² → <i>leg</i> ³
Part Holonym	Part-Of	From parts to wholes	<i>course</i> ⁷ → <i>meal</i> ¹
Substance Meronym		From substances to their subparts	<i>water</i> ¹ → <i>oxygen</i> ¹
Substance Holonym		From parts of substances to wholes	<i>gin</i> ¹ → <i>martini</i> ¹
Antonym		Semantic opposition between lemmas	<i>leader</i> ¹ ⇔ <i>follower</i> ¹
Derivationally Related Form		Lemmas w/same morphological root	<i>destruction</i> ¹ ⇔ <i>destroy</i> ¹

Figure C.2 Noun relations in WordNet.

Relation	Definition	Example
Hypernym	From events to superordinate events	<i>fly</i> ⁹ → <i>travel</i> ⁵
Troponym	From events to subordinate event (often via specific manner)	<i>walk</i> ¹ → <i>stroll</i> ¹
Entails	From verbs (events) to the verbs (events) they entail	<i>snore</i> ¹ → <i>sleep</i> ¹
Antonym	Semantic opposition between lemmas	<i>increase</i> ¹ ⇔ <i>decrease</i> ¹
Derivationally Related Form	Lemmas with same morphological root	<i>destroy</i> ¹ ⇔ <i>destruction</i> ¹

Figure C.3 Verb relations in WordNet.

to the notion of immediate hyponymy discussed on page 4. Each synset is related to its immediately more general and more specific synsets through direct hypernym and hyponym relations. These relations can be followed to produce longer chains of more general or more specific synsets. Figure C.4 shows hypernym chains for **bass**³ and **bass**⁷.

In this depiction of hyponymy, successively more general synsets are shown on successive indented lines. The first chain starts from the concept of a human bass singer. Its immediate superordinate is a synset corresponding to the generic concept of a singer. Following this chain leads eventually to concepts such as *entertainer* and *person*. The second chain, which starts from musical instrument, has a completely different path leading eventually to such concepts as musical instrument, device, and physical object. Both paths do eventually join at the very abstract synset *whole*, *unit*, and then proceed together to *entity* which is the top (root) of the noun hierarchy (in WordNet this root is generally called the **unique beginner**).

**unique
beginner**

C.3 Word Similarity: Thesaurus Methods

In Chapter 6 we introduced the embedding and cosine architecture for computing the similarity between two words. A thesaurus offers a different family of algorithms that can be complementary.

Although we have described them as relations between words, similar is actually a relationship between word *senses*. For example, of the two senses of *bank*, we

```

Sense 3
bass, basso --
(an adult male singer with the lowest voice)
=> singer, vocalist, vocalizer, vocaliser
=> musician, instrumentalist, player
=> performer, performing artist
=> entertainer
=> person, individual, someone...
=> organism, being
=> living thing, animate thing,
=> whole, unit
=> object, physical object
=> physical entity
=> entity
=> causal agent, cause, causal agency
=> physical entity
=> entity

Sense 7
bass --
(the member with the lowest range of a family of
musical instruments)
=> musical instrument, instrument
=> device
=> instrumentality, instrumentation
=> artifact, artefact
=> whole, unit
=> object, physical object
=> physical entity
=> entity

```

Figure C.4 Hyponymy chains for two separate senses of the lemma *bass*. Note that the chains are completely distinct, only converging at the very abstract level *whole, unit*.

might say that the financial sense is similar to one of the senses of *fund* and the riparian sense is more similar to one of the senses of *slope*. In the next few sections of this chapter, we will compute these relations over both words and senses.

The thesaurus-based algorithms use the structure of the thesaurus to define word similarity. In principle, we could measure similarity by using any information available in a thesaurus (meronymy, glosses, etc.). In practice, however, thesaurus-based word similarity algorithms generally use only the hypernym/hyponym (*is-a* or subsumption) hierarchy. In WordNet, verbs and nouns are in separate hypernym hierarchies, so a thesaurus-based algorithm for WordNet can thus compute only noun-noun similarity, or verb-verb similarity; we can't compare nouns to verbs or do anything with adjectives or other parts of speech.

The simplest thesaurus-based algorithms are based on the intuition that words or senses are more similar if there is a shorter **path** between them in the thesaurus graph, an intuition dating back to Quillian (1969). A word/sense is most similar to itself, then to its parents or siblings, and least similar to words that are far away. We make this notion operational by measuring the number of edges between the two concept nodes in the thesaurus graph and adding one. Figure C.5 shows an intuition; the concept *dime* is most similar to *nickel* and *coin*, less similar to *money*, and even less similar to *Richter scale*. A formal definition:

$$\text{pathlen}(c_1, c_2) = 1 + \text{the number of edges in the shortest path in the}$$

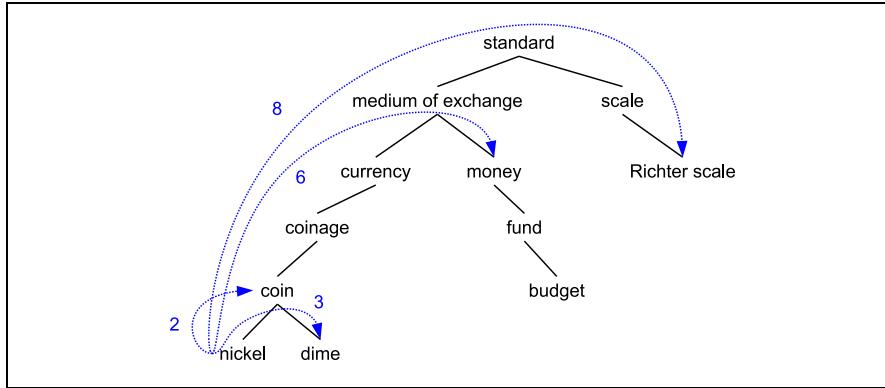


Figure C.5 A fragment of the WordNet hypernym hierarchy, showing path lengths (number of edges plus 1) from *nickel* to *coin* (2), *dime* (3), *money* (6), and *Richter scale* (8).

path-length
based similarity

word similarity

information-
content

thesaurus graph between the sense nodes c_1 and c_2

Path-based similarity can be defined as just the path length, transformed either by log (Leacock and Chodorow, 1998) or, more often, by an inverse, resulting in the following common definition of **path-length based similarity**:

$$\text{sim}_{\text{path}}(c_1, c_2) = \frac{1}{\text{pathlen}(c_1, c_2)} \quad (\text{C.17})$$

For most applications, we don't have sense-tagged data, and thus we need our algorithm to give us the similarity between words rather than between senses or concepts. For any of the thesaurus-based algorithms, following Resnik (1995), we can approximate the correct similarity (which would require sense disambiguation) by just using the pair of senses for the two words that results in maximum sense similarity. Thus, based on sense similarity, we can define **word similarity** as follows:

$$\text{wordsim}(w_1, w_2) = \max_{\substack{c_1 \in \text{senses}(w_1) \\ c_2 \in \text{senses}(w_2)}} \text{sim}(c_1, c_2) \quad (\text{C.18})$$

The basic path-length algorithm makes the implicit assumption that each link in the network represents a uniform distance. In practice, this assumption is not appropriate. Some links (e.g., those that are deep in the WordNet hierarchy) often seem to represent an intuitively narrow distance, while other links (e.g., higher up in the WordNet hierarchy) represent an intuitively wider distance. For example, in Fig. C.5, the distance from *nickel* to *money* (5) seems intuitively much shorter than the distance from *nickel* to an abstract word *standard*; the link between *medium of exchange* and *standard* seems wider than that between, say, *coin* and *coinage*.

It is possible to refine path-based algorithms with normalizations based on depth in the hierarchy (Wu and Palmer, 1994), but in general we'd like an approach that lets us independently represent the distance associated with each edge.

A second class of thesaurus-based similarity algorithms attempts to offer just such a fine-grained metric. These **information-content word-similarity** algorithms still rely on the structure of the thesaurus but also add probabilistic information derived from a corpus.

Following Resnik (1995) we'll define $P(c)$ as the probability that a randomly selected word in a corpus is an instance of concept c (i.e., a separate random variable, ranging over words, associated with each concept). This implies that $P(\text{root}) = 1$ since any word is subsumed by the root concept. Intuitively, the lower a concept

in the hierarchy, the lower its probability. We train these probabilities by counting in a corpus; each word in the corpus counts as an occurrence of each concept that contains it. For example, in Fig. C.5 above, an occurrence of the word *dime* would count toward the frequency of *coin*, *currency*, *standard*, etc. More formally, Resnik computes $P(c)$ as follows:

$$P(c) = \frac{\sum_{w \in \text{words}(c)} \text{count}(w)}{N} \quad (\text{C.19})$$

where $\text{words}(c)$ is the set of words subsumed by concept c , and N is the total number of words in the corpus that are also present in the thesaurus.

Figure C.6, from Lin (1998), shows a fragment of the WordNet concept hierarchy augmented with the probabilities $P(c)$.

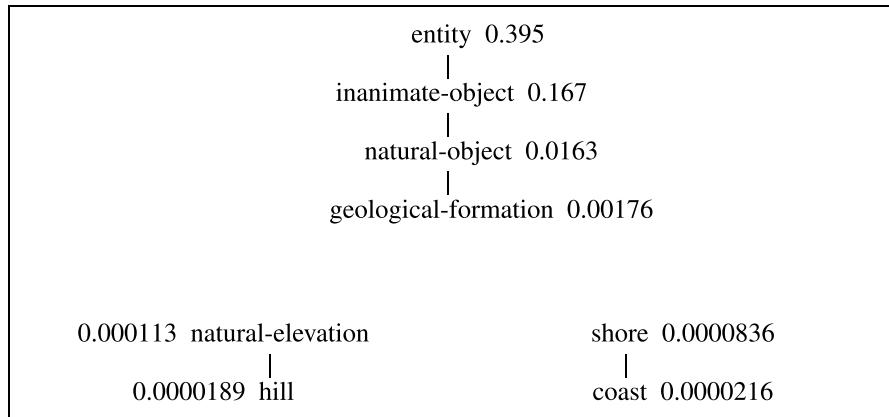


Figure C.6 A fragment of the WordNet hierarchy, showing the probability $P(c)$ attached to each concept, adapted from a figure from Lin (1998).

We now need two additional definitions. First, following basic information theory, we define the information content (IC) of a concept c as

$$\text{IC}(c) = -\log P(c) \quad (\text{C.20})$$

Lowest common subsumer LCS

Second, we define the **lowest common subsumer** or **LCS** of two concepts:

$\text{LCS}(c_1, c_2)$ = the lowest common subsumer, that is, the lowest node in the hierarchy that subsumes (is a hypernym of) both c_1 and c_2

Resnik similarity

There are now a number of ways to use the information content of a node in a word similarity metric. The simplest way was first proposed by Resnik (1995). We think of the similarity between two words as related to their common information; the more two words have in common, the more similar they are. Resnik proposes to estimate the common amount of information by the **information content of the lowest common subsumer of the two nodes**. More formally, the **Resnik similarity** measure is

$$\text{sim}_{\text{resnik}}(c_1, c_2) = -\log P(\text{LCS}(c_1, c_2)) \quad (\text{C.21})$$

Lin (1998) extended the Resnik intuition by pointing out that a similarity metric between objects A and B needs to do more than measure the amount of information in common between A and B. For example, he additionally pointed out that the more **differences** between A and B, the less similar they are. In summary:

- **Commonality:** the more information A and B have in common, the more similar they are.
- **Difference:** the more differences between the information in A and B, the less similar they are.

Lin measures the commonality between A and B as the information content of the proposition that states the commonality between A and B:

$$\text{IC}(\text{common}(A,B)) \quad (\text{C.22})$$

He measures the difference between A and B as

$$\text{IC}(\text{description}(A,B)) - \text{IC}(\text{common}(A,B)) \quad (\text{C.23})$$

where $\text{description}(A,B)$ describes A and B. Given a few additional assumptions about similarity, Lin proves the following theorem:

Similarity Theorem: The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are.

$$\text{sim}_{\text{Lin}}(A,B) = \frac{\text{common}(A,B)}{\text{description}(A,B)} \quad (\text{C.24})$$

Applying this idea to the thesaurus domain, Lin shows (in a slight modification of Resnik's assumption) that the information in common between two concepts is twice the information in the lowest common subsumer $\text{LCS}(c_1, c_2)$. Adding in the above definitions of the information content of thesaurus concepts, the final **Lin similarity** function is

$$\text{sim}_{\text{Lin}}(c_1, c_2) = \frac{2 \times \log P(\text{LCS}(c_1, c_2))}{\log P(c_1) + \log P(c_2)} \quad (\text{C.25})$$

For example, using sim_{Lin} , Lin (1998) shows that the similarity between the concepts of *hill* and *coast* from Fig. C.6 is

$$\text{sim}_{\text{Lin}}(\text{hill}, \text{coast}) = \frac{2 \times \log P(\text{geological-formation})}{\log P(\text{hill}) + \log P(\text{coast})} = 0.59 \quad (\text{C.26})$$

Jiang-Conrath distance

A similar formula, **Jiang-Conrath distance** (Jiang and Conrath, 1997), although derived in a completely different way from Lin and expressed as a distance rather than similarity function, has been shown to work as well as or better than all the other thesaurus-based methods:

$$\text{dist}_{\text{JC}}(c_1, c_2) = 2 \times \log P(\text{LCS}(c_1, c_2)) - (\log P(c_1) + \log P(c_2)) \quad (\text{C.27})$$

We can transform dist_{JC} into a similarity by taking the reciprocal.

Finally, we describe a **dictionary-based** method that is related to the Lesk algorithm for word sense disambiguation we will introduce in Section C.6.1. The intuition of **extended gloss overlap**, or **extended Lesk** measure (Banerjee and Pedersen, 2003) is that two concepts/senses in a thesaurus are similar if their glosses contain overlapping words. We'll begin by sketching an overlap function for two glosses. Consider these two concepts, with their glosses:

Extended gloss overlap extended Lesk

- *drawing paper*: paper that is specially prepared for use in drafting
- *decal*: the art of transferring designs from specially prepared paper to a wood or glass or metal surface.

For each n -word phrase that occurs in both glosses, Extended Lesk adds in a score of n^2 (the relation is non-linear because of the Zipfian relationship between lengths of phrases and their corpus frequencies; longer overlaps are rare, so they should be weighted more heavily). Here, the overlapping phrases are *paper* and *specially prepared*, for a total similarity score of $1^2 + 2^2 = 5$.

Given such an overlap function, when comparing two concepts (synsets), Extended Lesk not only looks for overlap between their glosses but also between the glosses of the senses that are hypernyms, hyponyms, meronyms, and other relations of the two concepts. For example, if we just considered hyponyms and defined $\text{gloss}(\text{hypo}(A))$ as the concatenation of all the glosses of all the hyponym senses of A, the total relatedness between two concepts A and B might be

$$\begin{aligned}\text{similarity}(A,B) = & \text{overlap}(\text{gloss}(A), \text{gloss}(B)) \\ & + \text{overlap}(\text{gloss}(\text{hypo}(A)), \text{gloss}(\text{hypo}(B))) \\ & + \text{overlap}(\text{gloss}(A), \text{gloss}(\text{hypo}(B))) \\ & + \text{overlap}(\text{gloss}(\text{hypo}(A)), \text{gloss}(B))\end{aligned}$$

Let RELS be the set of possible WordNet relations whose glosses we compare; assuming a basic overlap measure as sketched above, we can then define the **Extended Lesk** overlap measure as

$$\text{sim}_{\text{eLesk}}(c_1, c_2) = \sum_{r,q \in \text{RELS}} \text{overlap}(\text{gloss}(r(c_1)), \text{gloss}(q(c_2))) \quad (\text{C.28})$$

$\text{sim}_{\text{path}}(c_1, c_2) = \frac{1}{\text{pathlen}(c_1, c_2)}$
$\text{sim}_{\text{Resnik}}(c_1, c_2) = -\log P(\text{LCS}(c_1, c_2))$
$\text{sim}_{\text{Lin}}(c_1, c_2) = \frac{2 \times \log P(\text{LCS}(c_1, c_2))}{\log P(c_1) + \log P(c_2)}$
$\text{sim}_{\text{JC}}(c_1, c_2) = \frac{1}{2 \times \log P(\text{LCS}(c_1, c_2)) - (\log P(c_1) + \log P(c_2))}$
$\text{sim}_{\text{eLesk}}(c_1, c_2) = \sum_{r,q \in \text{RELS}} \text{overlap}(\text{gloss}(r(c_1)), \text{gloss}(q(c_2)))$

Figure C.7 Five thesaurus-based (and dictionary-based) similarity measures.

Figure C.7 summarizes the five similarity measures we have described in this section.

Evaluating Thesaurus-Based Similarity

Which of these similarity measures is best? Word similarity measures have been evaluated in two ways, introduced in Chapter 6. The most common intrinsic evaluation metric computes the correlation coefficient between an algorithm's word similarity scores and word similarity ratings assigned by humans. There are a variety

of such human-labeled datasets: the RG-65 dataset of human similarity ratings on 65 word pairs (Rubenstein and Goodenough, 1965), the MC-30 dataset of 30 word pairs (Miller and Charles, 1991). The WordSim-353 (Finkelstein et al., 2002) is a commonly used set of of ratings from 0 to 10 for 353 noun pairs; for example (*plane*, *car*) had an average score of 5.77. SimLex-999 (Hill et al., 2015) is a more difficult dataset that quantifies similarity (*cup*, *mug*) rather than relatedness (*cup*, *coffee*), and including both concrete and abstract adjective, noun and verb pairs. Another common intrinsic similarity measure is the TOEFL dataset, a set of 80 questions, each consisting of a target word with 4 additional word choices; the task is to choose which is the correct synonym, as in the example: *Levied is closest in meaning to: imposed, believed, requested, correlated* (Landauer and Dumais, 1997). All of these datasets present words without context.

Slightly more realistic are intrinsic similarity tasks that include context. The Stanford Contextual Word Similarity (SCWS) dataset (Huang et al., 2012) offers a richer evaluation scenario, giving human judgments on 2,003 pairs of words in their sentential context, including nouns, verbs, and adjectives. This dataset enables the evaluation of word similarity algorithms that can make use of context words. The *semantic textual similarity* task (Agirre et al. 2012, Agirre et al. 2015) evaluates the performance of sentence-level similarity algorithms, consisting of a set of pairs of sentences, each pair with human-labeled similarity scores.

Alternatively, the similarity measure can be embedded in some end-application, such as question answering or spell-checking, and different measures can be evaluated by how much they improve the end application.

C.4 Word Sense Disambiguation: Overview

**word sense
disambiguation
WSD**

The task of selecting the correct sense for a word is called **word sense disambiguation**, or **WSD**. WSD algorithms take as input a word in context and a fixed inventory of potential word senses and outputs the correct word sense in context. The input and the senses depends on the task. For machine translation from English to Spanish, the sense tag inventory for an English word might be the set of different Spanish translations. For automatic indexing of medical articles, the sense-tag inventory might be the set of MeSH (Medical Subject Headings) thesaurus entries.

When we are evaluating WSD in isolation, we can use the set of senses from a dictionary/thesaurus resource like WordNet. Figure C.4 shows an example for the word *bass*, which can refer to a musical instrument or a kind of fish.²

WordNet Sense	Spanish Translation	Roget Category	Target Word in Context
bass ⁴	lubina	FISH/INSECT	... fish as Pacific salmon and striped bass and...
bass ⁴	lubina	FISH/INSECT	... produce filets of smoked bass or sturgeon...
bass ⁷	bajo	MUSIC	... exciting jazz bass player since Ray Brown...
bass ⁷	bajo	MUSIC	... play bass because he doesn't have to solo...

Figure C.8 Possible definitions for the inventory of sense tags for *bass*.

lexical sample

It is useful to distinguish two WSD tasks. In the **lexical sample** task, a small

² The WordNet database includes eight senses; we have arbitrarily selected two for this example; we have also arbitrarily selected one of the many Spanish fishes that could translate English *sea bass*.