Psychology of Language
Prof. Jon Sprouse

Lecture 10: Introduction to Words and the Lexicon (part 2)
Let’s try a psychological definition

1. There are no transparent acoustic cues that help us identify words.

2. There are no transparent phonetic cues that help us identify words.

3. Combining acoustic/phonetic form and semantic (meaning) does some of the work for us, but breaks down with complex words.

This is disheartening, as it is beginning to look like we can’t create a simple definition of word that can be used to easily identify each and every word in a given language.

But I am sure you all “feel” as though there are words -- we certainly use this concept to discuss language (notice the spaces in this sentence!), and we feel as though we are perceiving individual words when we listen to speech.

A psychological definition: Words are whatever the chunks are that are stored in our brain.
Lexemes and the Lexicon

So let’s abandon the idea of “word”, and try to find a better one for our theory of language processing!

A good place to start in our search for the right mental representation (of the things that we’ve been calling “words”) is the fact that we store these representations in memory.

You may not have thought about it before, but the mapping between the phonetic form of words (the sound) and the semantics of words (the meaning) is completely arbitrary:

<table>
<thead>
<tr>
<th>language</th>
<th>“cat”</th>
</tr>
</thead>
<tbody>
<tr>
<td>arabic</td>
<td>besseh</td>
</tr>
<tr>
<td>ethiopian</td>
<td>domadh</td>
</tr>
<tr>
<td>farsi</td>
<td>gorbeth</td>
</tr>
<tr>
<td>hawaiian</td>
<td>popoki</td>
</tr>
<tr>
<td>icelandic</td>
<td>kottur</td>
</tr>
<tr>
<td>korean</td>
<td>koyangi</td>
</tr>
</tbody>
</table>

"can't....hold...
up...much.....longer"
Lexemes and the Lexicon

This means that we must memorize "words": the thing that we call a "word" is really a pairing of a phonetic representation and a semantic (meaning) representation that is stored in long term memory.

Auditory representation: \([k æ t]\)

Semantic (meaning) representation:

A lexeme is an item stored in the lexicon. In the case of "words", it is (at least) a pair of representations: an auditory representation and a semantic representation.

Lexicon: The section of long term memory where information about "words" are stored (visual form, phonetic form, semantics, etc)
What are the lexemes?

A **lexeme** is an item stored in the lexicon. In the case of “words”, it is (at least) a pair of representations: an auditory representation and a semantic representation.

Instead of using the idea of “words” (which doesn’t work that well), we can use the idea of lexemes to guide our investigation of language processing.

So instead of asking for a complete list of all of the words, we can look for a complete list of all of the lexemes...

Don’t be fooled by the simplicity of this question!

Yes, certain “words” like *cat* are obviously **lexemes** (stored in the lexicon).

But what about words like *snow* and *snows* and *snowy*?
A brief look at possible lexemes

Many words are morphologically simple:

- fierce
- desk
- boot
- at

These words are complete units. There are no sub-parts that have meaning outside of the word. So these are each a separate lexeme.

For example, what does the **erc** in **fierce** mean?

But some words do seem to have sub-parts to them:

- desks
- pencils
- boots

The **s** in these words seems to mean something like **multiple**: we call it the **plural**.

- jumped
- failed
- labeled

Similarly, the **ed** in these words seems to mean something like **in the past**, so we call it the **past tense**.

- preset
- prepay
- prequalified

The **pre** in these words seems to mean something like **before**.
Morphemes

Language researchers use a term to describe the sub-parts of words that have meaning:

**Morpheme**: the smallest unit of word that carries meaning

Some words are only a single morpheme:

- fierce
- desk
- boot
- at

Each of these words is a single morpheme - there is one unit in the word that carries meaning: the entire word itself.

Some words are two morphemes:

- desks
- pencils
- boots
- preset
- prepay
- prequalified

The **s** in these words is a morpheme.

The rest of the word is a morpheme too!

The **pre** in these words is a morpheme.

The rest of the word is a morpheme too!
Types of Morphemes

Morpheme: the smallest unit of word that carries meaning

We can already see that we have different types of morphemes.

Morphemes that can be a stand-alone word are called **free morphemes**

Morphemes that only occur attached to a free morpheme are called **bound morphemes**

And bound morphemes can come on either side of the free morpheme:

- **suffixes** occur at the end of the free morpheme
- **prefixes** occur at the beginning of the free morpheme
A list of morphemes is a list of possible lexemes!

Remember, our goal is to figure out which chunks of *sound/meaning* are stored in the lexicon. In other words, we want a complete list of *lexemes*.

What we’ve done so far is identify the *morphemes*.

A list of *morphemes* is a list of *potential* *lexemes*, but not a list of actual lexemes.

This is because there are two ways to form a complex word:

Option 1: Store the entire word in the lexicon

```
jumped
```

Option 2: Store each morpheme separately, and then put them together when needed

```
jump
  ed
```
Storage versus Computation

Remember, our goal is to figure out which chunks of sound/meaning are stored in the lexicon. In other words, we want a complete list of lexemes.

Under option 1, the full-storage hypothesis, words are stored as entire units in the lexicon. So a list of lexemes would look like this:

- qualify
- qualifies
- qualified
- prequalify
- prequalifies
- prequalified
- screen
- screens
- screened
- prescreen
- prescreens
- prescreened

= 12 lexemes

Under option 2, the full-computation hypothesis, each morpheme is stored as a unit in the lexicon, and combined as needed. So a list of lexemes would look like this:

- qualify
- screen
- s
- ed
- pre

= 5 lexemes
Storage versus Computation

The trade-off between STORAGE and COMPUTATION is actually a general theme of cognitive science!

Full-Storage
Advantage: faster speed
Disadvantage: higher memory cost

Full-Computation
Advantage: lower memory cost
Disadvantage: slower speed

Think about basic mathematics:
You store simple things like 4x5 because this equation happens often.
But you compute 45x54, because this equation doesn’t happen very often!
Diagnosing Storage vs. Computation

So the big questions we need to ask about complex words are:

1. Are complex words stored as a whole (full-storage)?

2. Are complex words stored as smaller pieces (morphemes) and combined as needed (full-computation)?

3. Are some complex words stored as a whole (partial-storage), while others are stored as smaller pieces (morphemes) and combined as needed (partial-computation)?

Since we can’t simply look inside a brain to figure this out, we are going to need some fairly clever experiments to get at the answers...
A real-world analogy

Imagine a world where IKEA works like this:

1. You go up to a counter and ask for a poang.
2. The clerk disappears into a warehouse, and returns with a fully assembled poang.

Here’s the question you need to answer:

Was the poang stored in its fully assembled state in the warehouse?

-or-

Did the clerk assemble it before bringing it to you?

What type of information would help you to answer this question?
What if you timed the clerk?

The two options suggest that they may take different amounts of time:

If the poangs are stored fully assembled, then all the clerk needs to do is go pick it up and bring it to you. This will probably be faster, but will require a very large warehouse!

If the poangs are stored disassembled, the clerk will need to assemble it before bringing it to you. This will probably be slower, but will require a smaller warehouse.
The Lexical Decision Task

The lexical decision task: Stare at this cross. When it changes to letters, tell me whether the letters form a “word” or not.
The lexical decision task: Stare at this cross. When it changes to letters, tell me whether the letters form a “word” or not.
The Lexical Decision Task

The lexical decision task: Stare at this cross. When it changes to letters, tell me whether the letters form a “word” or not.
The Lexical Decision Task

**The lexical decision task:** Stare at this cross. When it changes to letters, tell me whether the letters form a “word” or not.
The Lexical Decision Task

The lexical decision task: Stare at this cross. When it changes to letters, tell me whether the letters form a “word” or not.

The idea behind the task: In order to say “yes” or “no”, you need to look the word up in the lexicon. This means that the task will engage the processes necessary for lexical access.

If we measure the amount of time it takes to respond to a word (the reaction time), we can use it to make inferences about the processes that occurred during lexical access.