Language Models for Computer Science

• Why?
  
  compiling C++ to machine language
  translating English to a database query

• What?

  meaning:  from predicate logic
  structure:  from language models

• Structure

  old [men and women]
  [old people] and children
  log x + y
Symbol Sets and Languages

- Symbol set (alphabet) of characters
  \[ S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \]

- A useful set of strings built from \[ S \] possible ZIP codes (digit strings of length 5)

- Another symbol set of characters
  \[ S' = \text{the digits and capital letters} \]

- A useful set of strings built from \[ S' \] GMU course numbers (2-4 caps, then 3 digits)
More Complex Symbol Sets and Languages

• Symbol set (vocabulary) of words

\[ S = \{ \text{while, x, ==, int, ...} \} \]

• A useful set of strings built from \[ S \]

the legal programs of C++

• Symbol set of characters and HTML tags

\[ S = \{ \text{<head>, </head>, <body>, <p>, ...} \} \]

• A useful set of strings built from \[ S \]

HTML web pages
Definitions and Examples

Definitions

A string is a sequence of symbols.
A language is a set of strings.

Examples

Let \( S = \{a,b,c\} \) be the set of symbols.
Some strings over \( S \) : ab, bbc, abcba, ...

Let \( S = \{0,1,+\} \)
Some strings over \( S \) : 0, 1+1, 0+11+1011

Subset: the sums of 0s and 1s:
\{0+0, 0+1, 1+0, 1+1, 0+0+0, 0+0+1, ...\}
Smallest String and Language

∅ (lambda) is the empty string.

It is not a member of ∅. (It's a string, not a symbol.)

Its length is 0.

Ø is the empty set.

Regarded as a language, it is the set of no strings.

Its cardinality (size) is 0.
The Biggest Language (over \(\Sigma\))
\[\Sigma^*\ (\text{Sigma Star})\]

- \(\Sigma^*\) is the language that consists of all possible strings over the symbol set \(\Sigma\).
- For example, \(\{a,b\}^*\) represents all strings over the specific symbol set \(\{a,b\}\).
- So if \(\Sigma = \{a,b\}\),

\[\Sigma^* = \{a,b\}^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, aab\ldots\}\]
Specifying a Language: 
Extension and Intension

Extensional:

\[ L = \{ [], a, b, aa, ab, ba, bb \} \]

Intensional, informal:

\[ L = \text{strings over \{a,b\} of length at most 2} \]

Intensional, formal:

\[ L = \{ x \mid x \in \{a,b\}^* \text{ and } |x| \leq 2 \} \]
Concatenation of Strings

• Put one string after another

• No operator symbol

  ... like multiplication, but

• Order matters: *not* commutative

  (cba) (ba) = cbaba,

  but

  (ba) (cba) = bacba.
Length: | ... |

|ab| = 2 \quad |bcdgh| = 5

|x y| = |x| + |y|

| □ | = 0

symbols: a, b, c, 0, 1, ...

strings: x, y, ...
Identity and Repetition

• Identity element

For any string \(x\), we have \(x[] = x = []x\).

So [] is the identity element for concatenation.

• Repetition

\(x\) concatenated with itself is \(xx\), which can be written \(x^2\).

Concatenating \(x\) with \(x^k\) gives \(x^{k+1}\).

\(x^0 = []\).
Set Operations on Languages

Languages are sets, so

for $L_1$ and $L_2$, these are also languages:

- $L_1 \bigcap L_2$
- $L_1 \bigcup L_2$
- $L_1 \setminus L_2$

and so are these:

- $\emptyset^* \setminus L$
- the empty language, $\emptyset = \{\}$ ≠ {[]}
Another Operation on Languages: Concatenation

Concatenation of languages is based on concatenation of strings.

Definition

The concatenation of \( L_1 \) and \( L_2 \) is \( L_1L_2 \).

It contains every string that is the concatenation of a member of \( L_1 \) with a member of \( L_2 \). That is,

\[
L_1L_2 = \{xy \mid x \in L_1 \land y \in L_2\}
\]

\( L_1L_2 \) can differ from \( L_2L_1 \):

With \( L_1 = \{a, aa\} \) and \( L_2 = \{b, bb, bbb\} \).

\( L_1 \ L_2 = \{ab, abb, abbb, aab, aabb, aabbb\} \)

\( L_2 \ L_1 = \{ba, bba, bbba, baa, bbbaa, bbbba\} \)
Superscripts for Languages

Analogous to superscripts for strings

For any language $L$,

$L^0 = \{\epsilon\}$

for any integer $k$, $L^{k+1} = L \cdot L^k$

Therefore

$L^1 = L \quad L^2 = LL \quad L^3 = LLL$

Example: Let $L = \{a, bb\}$.

$L^2 = LL = \{aa, abb, bba, bbbb\}$

$L^3 = L \cdot L^2 = $

$\{aaa, aabb, abba, abbbb, bbaa, bbabb, bbbba, bbbbbb\}$
Closure * for Languages

\[ L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots \; ; \text{that is,} \]

\[ L^* = \bigcup_{i=0}^{\infty} L^i \]

Suppose that \( L = \{aa\} \).

Then \( L^* = \{L, \; aa, \; aaaa, \; aaaaaa, \ldots\} \)

The "*" is the closure operator or the (Kleene) star operator.
**Operator Precedence**

Example: \( L_1 \cdot L_2 \cdot L_3^* \)

- form \( L_3^* \), the closure of \( L_3 \).
- concatenate \( L_2 \) and \( L_3^* \) to form \( L_2 \cdot L_3^* \)
- take the union of \( L_1 \) and \( L_2 \cdot L_3^* \)
Using Operators to Build Languages

Given all the strings of length 0 and 1, ...

- **Concatenate**: build strings from shorter strings
- **Alternation (Union)**: to get languages from strings
- **Repetition (of alternatives)**: to get unlimited novelty