Probabilistic Languages
Some Models of Translation

- IBM Models 1-5
- Hidden Markov Model
- Phrase-Based Models

Q: What do all of these things have in common?
Definition 1: Natural language
“Language”

Definition 1: *Natural* language

Tends to not be well-defined.
“Language”

Definition 1: *Natural* language

Tends to not be well-defined.
Definition 1: *Formal language*

Well-defined, so that a computer can process it: a (possibly infinite) set of strings.

- All of the English words in a dictionary.
- All sequences of any length over those words.
- All English sentences with non-zero $p(e | f)$ for some French sentence $f$, according to your model.
We need efficient algorithms and data structures to:

- Encode all of the strings in the language.
- Assign probabilities to all of those strings.
  - Via products such as $p(e)p(f \mid e)$.
- Find the string with the highest probability.
- Compute expectations over substrings.
- Compute mappings between strings.
Regular Languages
Regular Languages

\[ L_1 = \{ \text{a a a, b a b, a a b, a b b} \} \]
Regular Languages

\[ L_1 = \{ a, aa, aaa, \ldots \} \]

\[ L_2 = a^* = \{ a, aa, aaa, \ldots \} \]
Regular Languages

\[ \mathcal{L}_1 = \{ a \, a \, a, a \, b \, a, a \, a \, b, a \, b \, b \} \]

\[ \mathcal{L}_2 = a^* = \{ a, aa, aaaa, \ldots \} \]

\[ \mathcal{L}_3 = \{ "the \ north \ wind \ howls" \} \]
Regular Languages

\[ \mathcal{L}_1 = \{ a^3, a \cdot b \cdot a, a^2 \cdot b, a \cdot b^2 \} \]

\[ \mathcal{L}_2 = a^* = \{ a, aa, aaa, \ldots \} \]

\[ \mathcal{L}_3 = \{ \text{“the north wind howls”} \} \]
Regular Languages

\[ \mathcal{L}_1 = \{ a \ a \ a, a \ b \ a, a \ a \ b, a \ b \ b \} \]

\[ \mathcal{L}_2 = a^* = \{ a, aa, aaaa, \ldots \} \]

\[ \mathcal{L}_3 = \{ "the \ north \ wind \ howls" \} \]
Regular Languages

\[ \mathcal{L}_1 = \{ a \ a \ a, a \ b \ a, a \ a \ b, a \ b \ b \} \]

\[ \mathcal{L}_2 = a^* = \{ a, aa, aaaa, \ldots \} \]

\[ \mathcal{L}_3 = \{ "the \ north \ wind \ howls" \} \]
Regular Languages

\[ \mathcal{L}_1 = \{ \text{a a a, a b a, a a b, a b b} \} \]

\[ \mathcal{L}_2 = a^* = \{ a, aa, aaaa, \ldots \} \]

\[ \mathcal{L}_3 = \{ \text{“the north wind howls”} \} \]

finite-state automata
Regular Languages
Regular Languages

\{ \epsilon \} \text{ is regular}
Regular Languages

\{ \epsilon \} \text{ is regular}
Regular Languages

\{ \epsilon \} \text{ is regular}
Regular Languages

\{\epsilon\} \text{ is regular} 

\{a\} \text{ is regular}
Regular Languages

\{\varepsilon\} \text{ is regular}

\{a\} \text{ is regular}
Regular Languages

\{ \epsilon \} \text{ is regular}

\{ a \} \text{ is regular}

\mathcal{L}_1 \cup \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}
Regular Languages

\{\varepsilon\} is regular

\{a\} is regular

\mathcal{L}_1 \cup \mathcal{L}_2\  is regular if  \mathcal{L}_1\ and\ \mathcal{L}_2\ are regular
Regular Languages

\(\{\epsilon\}\) is regular

\(\{a\}\) is regular

\(L_1 \cup L_2\) is regular if \(L_1\) and \(L_2\) are regular
Regular Languages

\{\varepsilon\} \text{ is regular}

\{a\} \text{ is regular}

\mathcal{L}_1 \cup \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}
Regular Languages

\( \{\epsilon\} \) is regular

\( \{a\} \) is regular

\( \mathcal{L}_1 \cup \mathcal{L}_2 \) is regular if \( \mathcal{L}_1 \) and \( \mathcal{L}_2 \) are regular

\( \mathcal{L}_1 \cdot \mathcal{L}_2 \)
Regular Languages

\{ \epsilon \} \text{ is regular}

\{ a \} \text{ is regular}

\mathcal{L}_1 \cup \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}

\mathcal{L}_1 \cdot \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}
Regular Languages

\{ \varepsilon \} \text{ is regular} \quad \text{ \[ \begin{array}{c}
\varepsilon \\
\end{array} \]
}

\{ a \} \text{ is regular} \quad \text{ \[ \begin{array}{c}
a \\
\end{array} \]
}

\mathcal{L}_1 \cup \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}

\mathcal{L}_1 \cdot \mathcal{L}_2 

Regular Languages

\{ \epsilon \} \text{ is regular}

\{ a \} \text{ is regular}

\mathcal{L}_1 \cup \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}

\mathcal{L}_1 \cdot \mathcal{L}_2
Regular Languages

\{ \varepsilon \} \text{ is regular}

\{ a \} \text{ is regular}

\mathcal{L}_1 \cup \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}

\mathcal{L}_1 \cdot \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}

\mathcal{L}_1^* \text{ is regular if } \mathcal{L}_1 \text{ is regular}
Regular Languages

\{ ε \} is regular

\{ a \} is regular

\mathcal{L}_1 \cup \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}

\mathcal{L}_1 \cdot \mathcal{L}_2

\mathcal{L}_1^*
Regular Languages

\{\varepsilon\} \text{ is regular} \\
\{a\} \text{ is regular}

\mathcal{L}_1 \cup \mathcal{L}_2 \text{ is regular if } \mathcal{L}_1 \text{ and } \mathcal{L}_2 \text{ are regular}

\mathcal{L}_1 \cdot \mathcal{L}_2 \\
\mathcal{L}_1^*
Regular Languages

Not all languages are regular!
Regular Languages

Not all languages are regular!

\[ \mathcal{L}_4 = \{ab, aabb, aaabbb, \ldots\} = \forall n \in [1, \infty) a^n b^n \]
Regular Languages

Not all languages are regular!

\[ \mathcal{L}_4 = \{ab, aabb, aaabbb, \ldots\} = \forall n \in [1, \infty) a^n b^n \]

We’ll talk about such context-free languages next week.
Regular Languages

Not all languages are regular!

\[ \mathcal{L}_4 = \{ab, aabb, aaabbb, \ldots\} = \forall n \in [1, \infty) a^n b^n \]

We’ll talk about such context-free languages next week.

But not all languages are context-free, either!
Probabilistic Regular Languages

We want a function:

\[ f : \mathcal{L} \rightarrow \mathbb{R}^+ \]
Probabilistic Regular Languages

We want a function:

\[ f : \mathcal{L} \rightarrow \mathbb{R}^+ \]

such that:

\[ f(w) \in [0, 1] \]

\[ \sum_w f(w) \in [0, 1] \]
Probabilistic Regular Languages

We want a function:

\[ f : \mathcal{L} \rightarrow \mathbb{R}^+ \]
We want a function:

\[ f : \mathcal{L} \rightarrow \mathbb{R}^+ \]
Minimization
Minimization
Other Algorithms

- Shortest path (e.g. Dijkstra, A*): most probable
- Determinization (not all can be determinized)
- Epsilon-removal
- Lazy composition (e.g. intersection): $p(e)p(f|e)$
Other Algorithms

- Shortest path (e.g. Dijkstra, A*): most probable
- Determinization (not all can be determinized)
- Epsilon-removal
- Lazy composition (e.g. intersection): $p(e)p(f \mid e)$
Practical Issues

- OpenFST (openfst.org)
  - Efficient C++ implementation.
  - Used in speech recognition (Google, Kaldi @ JHU)
  - Machine translation (JHU → Cambridge, Google)
Some Models of Translation

• IBM Models 1-5
• Hidden Markov Model
• Phrase-Based Models

Q: What do all of these things have in common?

A: They all define *weighted regular languages* over a set of output sentences. Details Thursday.
Questions
Questions

Are natural languages regular?
Questions

Are natural languages regular?

Does it matter for MT if they aren’t?