Probabilistic Finite States Machines

Franck Thollard

Jean-Monnet University
Saint-Etienne
France

Alicante 2006
Outline

The Power of expression

Learnability Issues

Learning the structure

Estimating the probabilities : Smoothing

Using the models

Franck Thollard
Probabilistic Finite States Machines
Outline

The Power of expression

Learnability Issues

Learning the structure

Estimating the probabilities : Smoothing

Using the models

Franck Thollard
Probabilistic Finite States Machines
The probability of a sequence
Computation using the chain rule

\[
P(he \ reads \ a \ book) = P(he) \times P(\text{reads}|he) \times P(a|he \ reads) \times P(\text{book}|he \ reads \ a)
\]
The probability of a sequence
Computation using the chain rule

\[
P(he \ reads \ a \ book) = P(he) \times P(reads|he) \\
\times P(a|he \ reads) \\
\times P(book|he \ reads \ a)
\]

and more generally:

\[
P(w_1 \ w_2 \ldots w_n) = P(w_1) \times P(w_2|w_1) \times \ldots \times P(w_n|w_1 w_2 \ldots w_{n-1})
\]
The probability of a sequence
Computation using the chain rule

\[ P(\text{he reads a book}) = P(\text{he}) \times P(\text{reads}|\text{he}) \times P(\text{a}|\text{he reads}) \times P(\text{book}|\text{he reads a}) \]

and more generally:

\[ P(w_1 \ldots w_n) = P(w_1) \times P(w_2|w_1) \times \ldots \times P(w_n|w_1w_2\ldots w_{n-1}) \]

Definition: \( w_1w_2\ldots w_{n-1} \) is called the **history**.
Some models
See (Vidal et al., 2005) for a survey

$n$-grams / MM $\iff k$-testables automata
Some models
See (Vidal et al., 2005) for a survey

- $n$-grams / MM $\Leftrightarrow$ $k$-testables automata
- Probabilistic Automata without cycles
Some models
See (Vidal et al., 2005) for a survey

- $n$-grams / MM $\Leftrightarrow$ \( k \)-testables automata
- Probabilistic Automata without cycles
- Probabilistic Deterministic Automata
Some models

See (Vidal et al., 2005) for a survey

- $n$-grams / MM $\leftrightarrow$ $k$-testables automata
- Probabilistic Automata without cycles
- Probabilistic Deterministic Automata
- Residual Automata (Esposito & al.’02)
Some models
See (Vidal et al., 2005) for a survey)

- \( n \)-grams / MM ⇔ \( k \)-testables automata
- Probabilistic Automata without cycles
- Probabilistic Deterministic Automata
- Residual Automata (Esposito & al.’02)
- Probabilistic non-deterministic Automata (⇔ HMM)
Some models
See (Vidal et al., 2005) for a survey

- $n$-grams / MM $\leftrightarrow k$-testables automata
- Probabilistic Automata without cycles
- Probabilistic Deterministic Automata
- Residual Automata (Esposito & al.’02)
- Probabilistic non-deterministic Automata ($\leftrightarrow$ HMM)
- PCFG: Probabilistic Context-Free Grammars
Some models
See (Vidal et al., 2005) for a survey

- $n$-grams / MM $\leftrightarrow k$-testables automata
- Probabilistic Automata without cycles
- Probabilistic Deterministic Automata
- Residual Automata (Esposito & al.’02)
- Probabilistic non-deterministic Automata ($\leftrightarrow$ HMM)
- PCFG: Probabilistic Context-Free Grammars

Note: models define a pdf on $\Sigma^n$, for each $n$
add of $\text{eos}$ symbol $\Rightarrow$ pdf on $\Sigma^*$
The $n$-grams model (1/3)

**Assumption**: the history is supposed bound
The \( n \)-grams model (1/3)

**Assumption:** the history is supposed bound

Example: history of size one \( \Rightarrow \) 2-grams (known as bigram)
The \( n \)-grams model (1/3)

**Assumption:** the history is supposed bound

Example: history of size one \( \Rightarrow \) 2-grams (known as *bigram*)

\[
P(he \ reads \ a \ book) = P(he) \times P(\text{reads}|he) \\
\times P(a|\text{reads}) \times P(\text{book}|a)
\]
The $n$-grams model (2/3)

**Estimating $n$-grams probabilities**

The probabilities are estimated using a corpus by counting occurrences of the $n$-uplets:

$$P(book|a) = \frac{Ct(a\ book)}{Ct(a)}$$
The $n$-grams (3/3)

Automata representation of $n$-grams

- the (0.9)
- student (0.4)
- eating (0.4)
- empanadas (1)
- they (0.1)
- mans (0.6)
- like (0.6)
- a (0.4)
- book (1)
Cyclic Automaton

unbounded history

A Probabilistic automaton

Note: $P(\text{papers} | \ldots)$ depends on an unbound history.
Choosing a model

<table>
<thead>
<tr>
<th>Model</th>
<th>Pro</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$-gram</td>
<td>Easy to build, Good estimates</td>
<td>Assumption Big model</td>
</tr>
</tbody>
</table>
### Choosing a model

<table>
<thead>
<tr>
<th>Model</th>
<th>Pro</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{n}-gram</td>
<td>Easy to build, Good estimates</td>
<td>Assumption, Big model</td>
</tr>
<tr>
<td>A-PDFA</td>
<td>History larger</td>
<td>Building the structure</td>
</tr>
</tbody>
</table>

Franck Thollard

Probabilistic Finite States Machines
## Choosing a model

<table>
<thead>
<tr>
<th>Model</th>
<th>Pro</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$-gram</td>
<td>Easy to build, Good estimates</td>
<td>Assumption, Big model</td>
</tr>
<tr>
<td>A-PDFA</td>
<td>History larger</td>
<td>Building the structure</td>
</tr>
<tr>
<td>PDFA</td>
<td>Unbound history, Parsing time, Correct results, Size of the model</td>
<td>Building the structure</td>
</tr>
</tbody>
</table>

Franck Thollard

Probabilistic Finite States Machines
Choosing a model

<table>
<thead>
<tr>
<th>Model</th>
<th>Pro</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-gram</td>
<td>Easy to build, Good estimates</td>
<td>Assumption, Big model</td>
</tr>
<tr>
<td>A-PDFA</td>
<td>History larger</td>
<td>Building the structure</td>
</tr>
<tr>
<td>PDFA</td>
<td>Unbound history, Parsing time</td>
<td>Building the structure</td>
</tr>
<tr>
<td></td>
<td>Correct results, Size of the model</td>
<td></td>
</tr>
</tbody>
</table>
Choosing a model

<table>
<thead>
<tr>
<th>Model</th>
<th>Pro</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>n-gram</td>
<td>Easy to build, Good estimates</td>
<td>Assumption, Big model</td>
</tr>
<tr>
<td>A-PDFA</td>
<td>History larger</td>
<td>Building the structure</td>
</tr>
<tr>
<td>PDFA</td>
<td>Unbound history, Parsing time, Correct results, Size of the model</td>
<td>Building the structure</td>
</tr>
<tr>
<td>PFA / HMM</td>
<td>Unbound history, Hand made topology, Size of the model</td>
<td>Parsing time or &quot;Loose&quot; of proba in non determinism</td>
</tr>
</tbody>
</table>
Outline

The Power of expression

Learnability Issues
  Learning criteria
  Theoretical results

Learning the structure

Estimating the probabilities: Smoothing

Using the models

Franck Thollard
Probabilistic Finite States Machines
Machine Learning Assumption

Formal Language Learning

INERENCE

aabb
aaabb
aaaaabbb

FORMAL LANGUAGE
What learning means?

\[ \begin{align*}
    & D \\
    \rightarrow & \text{Generation(D,C)} \\
    \leftarrow & G \\
    \rightarrow & \text{Inference} \\
    & D_1, D_2, \ldots, D_n \\
    \rightarrow & G_1, G_2, \ldots, G_n \\
    \end{align*} \]
Identification in the limit

\[ G_{n+k} = G_n \]
The Probabilistic PAC

Proba–D–PAC

\[ \begin{align*}
D & \xrightarrow{D} \text{GENERATION } D,C \\
C & \xrightarrow{G} \text{Inference}
\end{align*} \]

\[ \begin{align*}
D_1 & \\
D_2 & \\
\ldots & \\
D_n &
\end{align*} \]

\[ \begin{align*}
G_1 & \\
G_2 & \\
\ldots & \\
G_n &
\end{align*} \]

Franck Thollard
Probabilistic Finite States Machines
The Probabilistic PAC

Proba–D–PAC

\[ D(G, G_n) \]
The Probabilistic PAC

Proba–D–PAC

D(G, G_n) < ε
The Probabilistic PAC

Proba–D–PAC

\[ \Pr[D(G, G_n) < \varepsilon] > \delta \]
The Probabilistic PAC

Proba–D–PAC

\[ \Pr[D(G, G_n) < \varepsilon] > \delta \]

Link between
Precision / Confidance / #examples / class complexity / ...

Franck Thollard
Probabilistic Finite States Machines
The Power of expression
Learnability Issues
Learning the structure
Smoothing
Using the models

Theoretical results

Identification in the limit
With Proba One

- The class of **recursively enumerable** languages can be identified in the limit with probability one (Horning, 1969).
- The class of **probabilistic automata** can be identified in the limit with probability one with a polynomial algorithm (Carrasco’99).
The class of **recursively enumerable** languages can be identified in the limit with probability one (Horning, 1969).

The class of **probabilistic automata** cannot be identified in the limit with probability one using a polynomial number of examples (de la Higuera & Oncina’04).
Theoretical results

Proba-Poly-PAC (1/2)
Possible

- Proba-$d_\infty$-PAC PFA (Angluin’88, lemma 14)
Theoretical results

Proba-Poly-PAC (1/2)

Possible

- Proba-$d_\infty$-PAC PFA (Angluin’88, lemma 14)
- Proba-KL-PAC Unigram with unknown vocabulary (McAllester & Shapire’00)
Proba-Poly-PAC (1/2)
Possible

- Proba-\(d_{\infty}\)-PAC PFA (Angluin’88, lemma 14)
- Proba-KL-PAC Unigram with unknown vocabulary (McAllester & Shapire’00)
- Proba-KL-PAC PFA on \(\Sigma^n\), \(\Sigma\) and \(n\) known, nb of states known (Abe & Warmuth’92)
Proba-Poly-PAC (1/2)
Possible

- Proba-$d_\infty$-PAC PFA (Angluin’88, lemma 14)
- Proba-KL-PAC Unigram with unknown vocabulary (Mc Allester & Shapire’00)
- Proba-KL-PAC PFA on $\Sigma^n$, $\Sigma$ and $n$ known, nb of states known (Abe & Warmuth’92)
- Proba-KL-PAC Acyclic PDFA on $\Sigma^n$, nb States known, $\Sigma$ and $n$ known (Ron & al.’95)
Theoretical results

Proba-Poly-PAC (1/2)
Possible

- Proba-$d_\infty$-PAC PFA (Angluin’88, lemma 14)
- Proba-KL-PAC Unigram with unknown vocabulary (McAllester & Shapire’00)
- Proba-KL-PAC PFA on $\Sigma^n$, $\Sigma$ and $n$ known, nb of states known (Abe & Warmuth’92)
- Proba-KL-PAC Acyclic PDFA on $\Sigma^n$, nb States known, $\Sigma$ and $n$ known (Ron & al.’95)
- Proba-KL-PAC PDFA Struct. (Thollard & Clark’04)
Proba-Poly-PAC (1/2)
Possible

- Proba-$d_\infty$-PAC PFA (Angluin’88, lemma 14)
- Proba-KL-PAC Unigram with unknown vocabulary (Mc Allester & Shapire’00)
- Proba-KL-PAC PFA on $\Sigma^n$, $\Sigma$ and $n$ known, nb of states known (Abe & Warmuth’92)
- Proba-KL-PAC Acyclic PDFA on $\Sigma^n$, nb States known, $\Sigma$ and $n$ known (Ron & al.’95)
- Proba-KL-PAC PDFA Struct. (Thollard & Clark’04)
- Proba-KL-PAC PDFA + infos (Clark & Thollard’04)
Proba-Poly-PAC (1/2)

Possible

- **Proba-**$d_\infty$-PAC PFA (Angluin’88, lemma 14)
- **Proba-KL**-PAC Unigram with unknown vocabulary (Mc Allester & Shapire’00)
- **Proba-KL**-PAC PFA on $\Sigma^n$, $\Sigma$ and $n$ known, nb of states known (Abe & Warmuth’92)
- **Proba-KL**-PAC Acyclic PDFA on $\Sigma^n$, nb States known, $\Sigma$ and $n$ known (Ron & al.’95)
- **Proba-KL**-PAC PDFA Struct. (Thollard & Clark’04)
- **Proba-KL**-PAC PDFA + infos (Clark & Thollard’04)
- **Proba-Var**-PAC PDFA (Palmer & Goldberg’05)
Sketch proof

2 steps:

1. If the sample is "good" the result of the algorithm will be good.
The Power of expression

Learnability Issues

Learning the structure

Smoothing

Using the models

Theoretical results

Sketch proof

2 steps:

1. If the sample is "good" the result of the algorithm will be good.

2. Under certain assumptions, the probability of having a "good" sample is high.
Proba-Poly-PAC (2/2)

Impossible

- Proba AFN on $\Sigma^n$, $\Sigma$ unknown (Abe & Warmuth’92)
- PDF of unknown class on $\{0, 1\}^n$ (Kearns & al.’94)
- Proba-KL-PAC Cyclic Aut, without aut information (Clark & Thollard’02)
Typical family of problematic cases

\[ (1-p) \] \[ a \ (p) \] \[ b \ (1-p) \] \[ .05 \] \[ b \ (.05) \] \[ a \ (.9) \]
Outline

The Power of expression

Learnability Issues

Learning the structure
  Generic Algorithm
  Ordering Merge
  Compatibility tests
  Others learning schemes

Estimating the probabilities : Smoothing
Template algorithm

The common strategy follows two steps

- Building a maximum likelihood estimate of the data
The common strategy follows two steps

- Building a maximum likelihood estimate of the data
- Generalizing using state merging operations.
Learning a ML estimate: the PPTA

PPTA of the learning set
EA = \{\lambda,aac,aaa, aac, abc,aac,abc, abc, \lambda, a, ab \}
Learning a ML estimate: the PPTA

PPTA of the learning set

\[ EA = \{ \lambda, aac, aaa, aac, abc, aac, abc, abc, \lambda, a, ab \} \]
Learning a ML estimate: the PPTA

PPTA of the learning set
EA = \{\lambda, aac, aaa, aac, abc, aac, abc, abc, \lambda, a, ab\}

Note: String "aba" has null probability
Generalization

Choosing two states

Franck Thollard
Probabilistic Finite States Machines
Choosing two states

0 (2/11) \rightarrow 1 (1/9)

a_{9/11} \rightarrow a \rightarrow c (3/4)

b (4/9) \rightarrow 2 (1/4)

c (3/4) \rightarrow 4 (3/3)

Choosing state 2 and 3

Franck Thollard

Probabilistic Finite States Machines
Merging states 2 and 3

Merging 2 and 3: can lead to non determinism
### Generalization

Merging states 4 and 5

![Diagram showing states and transitions]

#### Merging states 4 and 5

- **State 0 (2/11)**
  - Transition to **State 1 (1/9)** on input `a` with probability `9/11`

- **State 1 (1/9)**
  - Transition to **State 2 (1/8)** on input `b` with probability `4/9`

- **State 2 (1/8)**
  - Transition to **State 4 (3/3)** on input `c` with probability `3/8`
  - Transition to **State 5 (3/3)** on input `c` with probability `3/8`

- **State 4 (3/3)**
  - Transition to **State 5 (3/3)** on input `a` with probability `1/8`

- **State 5 (3/3)**
  - Transition to **State 3 (1/1)** on input `a` with probability `1/8`
Generalization

After the "determinization"

Note: String "aba" has non null probability

Franck Thollard
Probabilistic Finite States Machines
**Generic Algorithm**

**Input:** A multiset of string EA, a real $\alpha$

**Output:** A PDFA

```
begin
  Building-PPTA(EA);
  A ← PPTA;
  while (q_i,q_j) ← Choosing-Two-States(A) do
    if Compatible (i,j,\alpha) then
      A ← Merge (A,q_i,q_j);
    end
  end
  Return A;
end
```
Ordering merges

PPTA built on the multiset
\[ EA = \{ \lambda, aac, aac, abc, aac, aac, abc, abc, \lambda, a, ab \} \]
Merging ordering

Franck Thollard
Probabilistic Finite States Machines
Alergia (Carrasco & Oncina’94)

HMM-infer (Stolcke’94): looking at each merge at each time → not tracktable on big data sets
Ordering Merge

Merging ordering

Alergia (Carrasco & Oncina’94):

HMM-infer (Stolcke’94): looking at each merge at each time → not tracktable on big data sets

LAPPTA (Ron & al.’95): building of acyclic automata
Alergia (Carrasco & Oncina’94) :
HMM-infer (Stolcke’94): looking at each merge at each time → not tracktable on big data sets
LAPPTA (Ron & al.’95): building of acyclic automata
EDSM (Lang’98): merging ordering based on the quantity of information
Alergia (Carrasco & Oncina’94):

HMM-infer (Stolcke’94): looking at each merge at each time → not tractable on big data sets

LAPPTA (Ron & al.’95): building of acyclic automata

EDSM (Lang’98): merging ordering based on the quantity of information

DDSM (Thollard’01): ordering adapted from the EDSM algorithm.
Compatibility tests

Alergia (Carrasco’94): Statistic test based on Hoeffding bounds
Compatibility tests

Alergia (Carrasco’94): Statistic test based on Hoeffding bounds
LAPPTA (Ron & al.’95): Statistic test based on similarity measure
Compatibility tests

**Alergia (Carrasco’94):** Statistic test based on Hoeffding bounds

**LAPPTA (Ron & al.’95):** Statistic test based on similarity measure

**Youg-Lai & Tompa’00:** Same as Alergia but emphasis on low frequency problem
Compatibility tests

Alergia (Carrasco’94): Statistic test based on Hoeffding bounds

LAPPTA (Ron & al.’95): Statistic test based on similarity measure

Youg-Lai & Tompa’00: Same as Alergia but emphasis on low frequency problem

MDI (Thollard & al.’00): Tradeoff between size and distance to the data
Compatibility tests

Alergia (Carrasco’94): Statistic test based on Hoeffding bounds
LAPPTA (Ron & al.’95): Statistic test based on similarity measure
Youg-Lai & Tompa’00: Same as Alergia but emphasis on low frequency problem
MDI (Thollard & al.’00): Tradeoff between size and distance to the data
M-Alergia (Kermorvant & Dupont’02): Statistical test based on multinomial test
Compatibility tests

Alergia (Carrasco’94): Statistic test based on Hoeffding bounds

LAPPTA (Ron & al.’95): Statistic test based on similarity measure

Youg-Lai & Tompa’00: Same as Alergia but emphasis on low frequency problem

MDI (Thollard & al.’00): Tradeoff between size and distance to the data

M-Alergia (Kermorvant & Dupont’02): Statistical test based on multinomial test

Alergia (Habrard & al.’03): Defines and deal with uniform noise
Other learning schemes

- Splitting/merging strategy (Brant’96; Ostendorf and Singer’97)
- Incremental learning (Carrasco’99, Thollard & Clark’04, Callut & Dupont’04; Denis & al.’06)
Outline

The Power of expression

Learnability Issues

Learning the structure

Estimating the probabilities : Smoothing
  Why smoothing ?
  How to smooth ?
  Further work on Smoothing

Franck Thollard
Probabilistic Finite States Machines
Why smoothing (1/2)

Reality
Why smoothing (2/2)

What we see!
Smoothing discrete spaces

The farm example

Let us be in front of a farm.... and we observe:

- 3 chickens
- 2 ducks

Estimating probabilities using the maximum likelihood

Note: need to work with **close vocabulary**
The farm example

Let us be in front of a farm.... and we observe:

- 3 chickens
- 2 ducks

Estimating probabilities using the maximum likelihood

Pr(chicken) = ?

Note: need to work with close vocabulary
Smoothing discrete spaces

The farm example

Let us be in front of a farm.... and we observe:

- 3 chickens
- 2 ducks

Estimating probabilities using the maximum likelihood

Pr(chicken) = 3/5

Note: need to work with **close vocabulary**
Smoothing discrete spaces

The farm example

Let us be in front of a farm.... and we observe:

- 3 chickens
- 2 ducks

Estimating probabilities using the maximum likelyhood

\[ \Pr(\text{pig}) = 0 \]

Note: need to work with close vocabulary
Smoothing discrete spaces

The farm example

Let us be in front of a farm.... and we observe:

- 3 chickens
- 2 ducks

Estimating probabilities using the maximum likelihood

Pr(lion) = 0

Note: need to work with close vocabulary
The smoothing problem

Recall the chain rule

\[ P(w_1 w_2 \ldots w_n) = P(w_1) \times P(w_2|w_1) \times \ldots \times \prod_{n=3}^{N} P(w_n|w_{n-2} w_{n-1}) \]

\( n \)-grams: the history \( w_1 w_2 \ldots w_{n-1} \) is bound by \( n - 1 \).

\( n \)-grams the problem needs to be considered **only** with null probability
Smoothing \( n \)-grams

- Good-Turing estimator: Non ML estimate
Smoothing \( n \)-grams

- Good-Turing estimator: Non ML estimate
- Discounting
Smoothing $n$-grams

- Good-Turing estimator: Non ML estimate
- Discounting
- Interpolation of many models
Smoothing $n$-grams

- Good-Turing estimator: Non ML estimate
- Discounting
- Interpolation of many models
- Non shadowing while discounting
How to smooth ?

Smoothing $n$-grams

- Good-Turing estimator : Non ML estimate
- Discounting
- Interpolation of many models
- Non shadowing while discounting
- Skipping model
How to smooth?

Smoothing \( n \)-grams

- Good-Turing estimator: Non ML estimate
- Discounting
- Interpolation of many models
- Non shadowing while discounting
- Skipping model
- Clustering / Latent analysis / ...
How to smooth?

Smoothing automata

- estimating the first null transition?
- where to go then? in the same automaton? where?
How to smooth?

The smoothing problem

Franck Thollard
Probabilistic Finite States Machines
How to smooth?

Smoothing automata

- LAPPTA (Ron & al.'95): Creation of a "small frequency state"
Smoothing automata

- LAPPTA (Ron & al.’95): Creation of a ”small frequency state”
- Alergia2 (Young-lai & Tompa’00): Emphasis on small frequency transitions
Smoothing automata

- LAPPTA (Ron & al.’95): Creation of a ”small frequency state”
- Alergia2 (Young-lai & Tompa’00): Emphasis on small frequency transitions
- Error-correcting (Dupont & Amengual’00)
Smoothing automata

- LAPPTA (Ron & al.’95): Creation of a "small frequency state"
- Alergia2 (Young-lai & Tompa’00): Emphasis on small frequency transitions
- Error-correcting (Dupont & Amengual’00)
- Discounting (Thollard’01): Back-off to a unigram
Smoothing automata

- LAPPTA (Ron & al.’95): Creation of a ”small frequency state”
- Alergia2 (Young-lai & Tompa’00): Emphasis on small frequency transitions
- Error-correcting (Dupont & Amengual’00)
- Discounting (Thollard’01): Back-off to a unigram
- Discounting (Llorens & al.’02): Back-off to aut
Smoothing automata

- LAPPTA (Ron & al.’95): Creation of a ”small frequency state”
- Alergia2 (Young-lai & Tompa’00): Emphasis on small frequency transitions
- Error-correcting (Dupont & Amengual’00)
- Discounting (Thollard’01): Back-off to a unigram
- Discounting (Llorens & al.’02): Back-off to aut
- Discounting (Mc Allester & Shapire’00): Theoretical discounting for the unigram.
Smoothing automata

- LAPPTA (Ron & al.’95): Creation of a ”small frequency state”
- Alergia2 (Young-lai & Tompa’00): Emphasis on small frequency transitions
- Error-correcting (Dupont & Amengual’00)
- Discounting (Thollard’01): Back-off to a unigram
- Discounting (Llorens & al.’02): Back-off to aut
- Discounting (Mc Allester & Shapire’00): Theoretical discounting for the unigram.
- Additive smoothing (Thollard & Clark’04): Theoretical justification.
How to smooth?

Smoothing techniques

Error-correcting
(Dupont & Amengual, 2000)

Needs a model of error/correction → set up time

Franck Thollard
Probabilistic Finite States Machines
How to smooth?

Smoothing techniques: Discounting (1/2)

Pro: Fast at parsing time; Easy to compute
Cons: Not as good as Error-correcting

Franck Thollard
Probabilistic Finite States Machines
How to smooth?

Smoothing techniques: Discounting (2/2)

Better than discounting to unigram

Llorens & al.’02

Franck Thollard
Probabilistic Finite States Machines
How to smooth?

**Smoothing techniques : Discounting (2/2)**

Very slow at parsing time

Franck Thollard
Probabilistic Finite States Machines
Further work on Smoothing

Smoothing: Further work

$n$-grams: dealing with big $n$ without too much smoothing
Further work on Smoothing

Smoothing: Further work

\textit{n-grams}: dealing with big $n$ without too much smoothing

\textbf{Automata}

- Estimating where to continue parsing (hard to tackle)
- Avoiding multiple smoothing for the same sentence (hard to tackle)
- Discounting to $n$-grams (easy to do, but need a good automaton)
- Setting a variable smoothing parameter ....
Further work on Smoothing

Around the inference

Franck Thollard
Probabilistic Finite States Machines
Around the inference

Clustering (Dupont & Chase’98)
Further work on Smoothing

Around the inference

Clustering (Dupont & Chase’98)
Interpolating automata (Thollard’01)
The Power of expression

Learnability Issues

Learning the structure

Smoothing

Using the models

Further work on Smoothing

Around the inference

Clustering (Dupont & Chase’98)

Interpolating automata (Thollard’01)

Bagging (Thollard & Clark’02)

Franck Thollard

Probabilistic Finite States Machines
Further work on Smoothing

Around the inference

Clustering (Dupont & Chase’98)
Interpolating automata (Thollard’01)
Bagging (Thollard & Clark’02)
Boosting (Thollard & al’02)
Typing automata (Kermorvant & de la Higuera’02)
Outline

The Power of expression

Learnability Issues

Learning the structure

Estimating the probabilities: Smoothing

Using the models

Off-line evaluating

“Real world” Applications

Franck Thollard

Probabilistic Finite States Machines
Evaluating a language model

- The perplexity
Evaluating a language model

- The perplexity
- The shannon game
Evaluating a language model

- The perplexity
- The shannon game
- The error rate
Evaluating a language model
The perplexity

Information theory
Evaluating a language model
The perplexity

- Information theory
- Geometric averaging
The Entropy

**Entropy:** $H(D) \overset{\text{def}}{=} - \sum_{x \in \Sigma^*} D(x) \log D(x)$

- Average number of questions needed to find a word of the vocabulary.
- Expectation of the optimal coding ($\log \frac{1}{D(x)}$)

The entropy is maximal for the uniform distribution.
The Kullback-Leibler divergence

**Relativ Entropy:**

\[ KL(C, A) \overset{\text{def}}{=} \sum_{x \in \Sigma^*} C(x) \log \frac{C(x)}{A(x)} = E_C \left[ \log \frac{C(x)}{A(x)} \right] \]

\[ = - \sum_{x \in \Sigma^*} C(x) \log(A(x)) - \left( - \sum_{x \in \Sigma^*} C(x) \log(C(x)) \right) \]

- crossentropy btw A and C
- \( H(C) \)
Properties of the KL

\[ KL(C, A) = \infty \quad \text{if} \quad \exists x : C(x) \neq 0 \land A(x) = 0, \]
Properties of the KL

- $KL(C, A) = \infty$ if $\exists x : C(x) \neq 0 \land A(x) = 0$, 
- $KL(C, A) \geq 0$, 

Franck Thollard

Probabilistic Finite States Machines
Properties of the KL

- \( KL(C, A) = \infty \) if \( \exists x : C(x) \neq 0 \land A(x) = 0 \),
- \( KL(C, A) \geq 0 \),
- \( KL(C, A) = 0 \iff \forall x \in \Sigma^*, C(x) = A(x) \),
Properties of the KL

- \( KL(C, A) = \infty \) if \( \exists x : C(x) \neq 0 \land A(x) = 0 \),
- \( KL(C, A) \geq 0 \),
- \( KL(C, A) = 0 \iff \forall x \in \Sigma^*, C(x) = A(x) \),
- Not symmetric: \( KL(C, A) \neq KL(A, C) \),
Properties of the KL

- \( KL(C, A) = \infty \) if \( \exists x : C(x) \neq 0 \land A(x) = 0 \),
- \( KL(C, A) \geq 0 \),
- \( KL(C, A) = 0 \iff \forall x \in \Sigma^*, C(x) = A(x) \),
- Not symmetric: \( KL(C, A) \neq KL(A, C) \),
- No triangle inequality: \( KL(C, A) \nless KL(C, B) + KL(B, A) \).
Properties of the KL

- $KL(C, A) = \infty$ if $\exists x : C(x) \neq 0 \land A(x) = 0$,
- $KL(C, A) \geq 0$,
- $KL(C, A) = 0 \iff \forall x \in \Sigma^*, C(x) = A(x)$,
- Not symmetric: $KL(C, A) \neq KL(A, C)$,
- No triangle inequality: $KL(C, A) \nleq KL(C, B) + KL(B, A)$.
- Greater than $L_1$: $KL(C, A) \geq \frac{1}{2\ln 2} \left( \| C - A \|_1 \right)^2$.
Off-line evaluating

From KL to perplexity

\[- \sum_{x \in \Sigma^*} C(x) \log(A(x)) \quad \left( - \left( \sum_{x \in \Sigma^*} C(x) \log(C(x)) \right) \right) \]

\[H(C)\]
From KL to perplexity

\[- \sum_{x \in \Sigma^*} C(x) \log(A(x)) - \left( - \left( \sum_{x \in \Sigma^*} C(x) \log(C(x)) \right) \right) \]

Suppressing constant factor $H(C)$
Off-line evaluating

From KL to perplexity

\[- \sum_{x \in \Sigma^*} C(x) \log(A(x))\]

Estimating the distribution $C$
From KL to perplexity

\[- \sum_{x \in \Sigma^*} \tilde{C}(x) \log(A(x))\]

Estimating the distribution \(C\)

Franck Thollard
Probabilistic Finite States Machines
From KL to perplexity

\[- \sum_{x \in S} \frac{c_t \cdot S(x)}{|S|} \log(A(x))\]

Estimating the distribution $C$ on $S$
From KL to perplexity

\[
- \frac{1}{\|S\|} \sum_{s_i=1}^{\|S\|} \log(A(s_i))
\]

Estimating the distribution \( C \) on \( S \)
From KL to perplexity

\[ PP = 2 \left[ -\frac{1}{\|S\|} \sum_{s_i=1}^{\|S\|} \log(A(s_i)) \right] \]

Renormalization (as log is base 2)
Perplexity as averaging

\[ PP(S|A) = \left[ \prod_{i=1}^{||S||} A(a_i) \right] ^{-\frac{1}{||S||}} \]
Off-line evaluating

Evaluating language models:
The shannon game

**Input:** a sentence with missing words

**Output:** the probability of the missing words
Evaluating language models: The shannon game

**Input:** a sentence with missing words

**Output:** the probability of the missing words

**Interests:**

- For left-right model $\iff$ perplexity,
- Allows to take into account the right context.
Evaluating language models:
weighting latices

Speech recognition or OCR works in 2 steps.
First step provides a weighted lattice
The goal is to reweight the lattice and compute WER.
Applications issues

- Information extraction improves precision (Freitag’97),

Franck Thollard
Probabilistic Finite States Machines
Applications issues

- **Information extraction** improves precision (Freitag’97),
- **Speech to text**: faster at parsing time than bigrams (Experimentation at FT, R&D),
Applications issues

- **Information extraction** improves precision (Freitag’97),
- **Speech to text**: faster at parsing time than bigrams (Experimentation at FT, R&D),
- **Language Modeling**: much compact models (Thollard’01),
Applications issues

- **Information extraction** improves precision (Freitag’97),
- **Speech to text**: faster at parsing time than bigrams (Experimentation at FT, R&D),
- **Language Modeling**: much compact models (Thollard’01),
- **Noun phrase chunking**: competitive results (∼ 90%) (Thollard & Clark’04),
Applications issues

- **Information extraction** improves precision (Freitag’97),
- **Speech to text**: faster at parsing time than bigrams (Experimentation at FT, R&D),
- **Language Modeling**: much compact models (Thollard’01),
- **Noun phrase chunking**: competitive results (∼ 90%) (Thollard & Clark’04),
- **Body rule generation**: better and more compact than $n$-gram (Infante-Lopez’04).
Applications issues

- **Information extraction** improves precision (Freitag’97),
- **Speech to text**: faster at parsing time than bigrams (Experimentation at FT, R&D),
- **Language Modeling**: much compact models (Thollard’01),
- **Noun phrase chunking**: competitive results (≈ 90%) (Thollard & Clark’04),
- **Body rule generation**: better and more compact than $n$-gram (Infante-Lopez’04).
- **Eye tracking modeling**: 4th at a PASCAL Challenge (Thollard and Largeron’05)