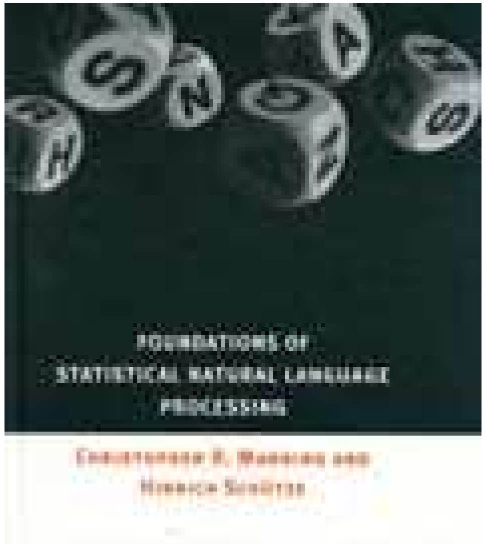
Maximum Entropy Methods in Language Processing

Dietrich Klakow

Literature

Manning, C. D. and H. Schütze:
Foundations of Statistical Natural Language Processing.
The MIT Press. 1999
ISBN 0-262-13360-1.
Chapter 16.2



Motivation and Simple Examples

Introduction

The concept of maximum entropy can be traced back along multiple threads to Biblical times. Only recently however have computers become powerful enough to permit the widescale application of this concept to real world problems in statistical estimation and pattern recognition.

From: „A Maximum Entropy Approach to Natural Language Processing„ by Adam L Berger, Stephen A Della Pietra, Vincent J Della Pietra

Toy example

- Task:
  - Translate German word `in` to English
  - Possibible alternatives:
    - `in, at, within, into, to`

Estimate the probabilities

Normalisation:

$$P(\texttt{in}) + P(\texttt{at}) + P(\texttt{within}) + P(\texttt{into}) + P(\texttt{to}) = 1$$

What is the least biased way of determining the probabilities?

Estimate the probabilities

Uniform Distribution:

$$P(\texttt{in}) = \frac{1}{5}$$

$$P(\texttt{at}) = \frac{1}{5}$$

$$P(\texttt{within}) = \frac{1}{5}$$

$$P(\texttt{into}) = \frac{1}{5}$$

$$P(\texttt{to}) = \frac{1}{5}$$

Estimate the probabilities

Normalisation:

$$P(\texttt{in}) + P(\texttt{at}) + P(\texttt{within}) + P(\texttt{into}) + P(\texttt{to}) = 1$$

Additional observation

$$P(\texttt{in}) + P(\texttt{at}) = 3/10$$

What is the least biased way of determining the probabilities?

Estimate the probabilities

Solution to problem from previous slide:

$$P(\texttt{in}) = \frac{3}{20}$$

$$P(\texttt{at}) = \frac{3}{20}$$

$$P(\texttt{within}) = \frac{7}{30}$$

$$P(\texttt{into}) = \frac{7}{30}$$

$$P(\texttt{to}) = \frac{7}{30}$$

Why „maximum entropy method“?

Formal Definition of Entropy

$$H(V) = \mathrm{E}[-\log(p(V))]$$

$$= \sum_{w_i \in V} - p(w_i)\log(p(w_i))$$

where V is a set of symbols

and is w_i the i - th symbol

Example

V is set of two symbols $V=\{a,b\}$

$P(a)=p$

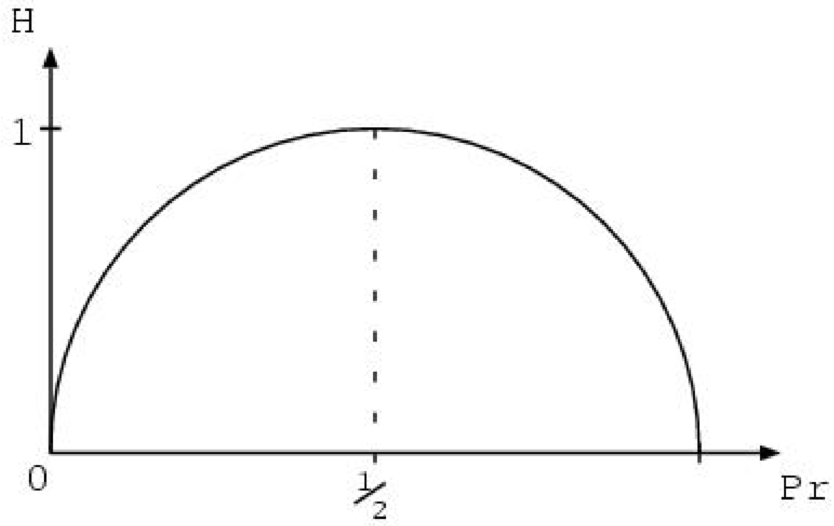
$P(b)=1-p$

$H=-p \log p - (1-p) \log(1-p)$

$p=0 \mapsto H=0$

$p=1 \mapsto H=0$

Entropy H=-p log p – (1-p) log(1-p)



Maximum if probabilities for the two symbols are identical

The Maximum Entropy Method

Maximize the entropy because it gives the least prejudiced distribution.

While maximizing, take constraints into account.

Linear Constraints

What are linear constraints good for

- Formalizing our requirements about the final probability distribution
- Taking into account our knowledge derived from a corpus
- Linear, because nonlinear models are more complex

Extend the translation example to include context

Notation:

x: word in the source language

y: word in the target language

Example sentence fragment:

Source language:

“Er geht *in* die Schule.“

Target language:

“He goes *to* school.”

Indicator Functions (feature functions)

- Try to capture essential information from context

$$f_1(x, y) = \begin{cases} 1 & \text{if } y = \text{"to" and "geht" preceeds "in"} \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(x, y) = \begin{cases} 1 & \text{if } y = \text{"to" and "die Schule" follows "in"} \\ 0 & \text{otherwise} \end{cases}$$

Integrate Constraints into Probabilities

- Empirical expectation value of feature

$$\tilde{p}(f_i) \equiv \sum_{x,y} \tilde{p}(x, y) f_i(x, y)$$

With $\tilde{p}(x, y)$: empirical distribution on corpus (e.g. relative frequencies)

- Expected value of feature derived from unknown model p(y|x)

$$p(f_i) \equiv \sum_{x,y} \tilde{p}(x) p(y \mid x) f_i(x, y)$$

Integrate Constraints into Probabilities

- Requirement: match model to corpus statistics

$$p(f_i) = \tilde{p}(f_i)$$

$\mapsto$

$$\sum_{x,y} \tilde{p}(x) p(y \mid x) f_i(x, y) = \sum_{x,y} \tilde{p}(y, x) f_i(x, y)$$

$\boxed{\text{Linear constraint}}$

Set of possible probability distributions

- All possible probability distributions satisfying constraints:

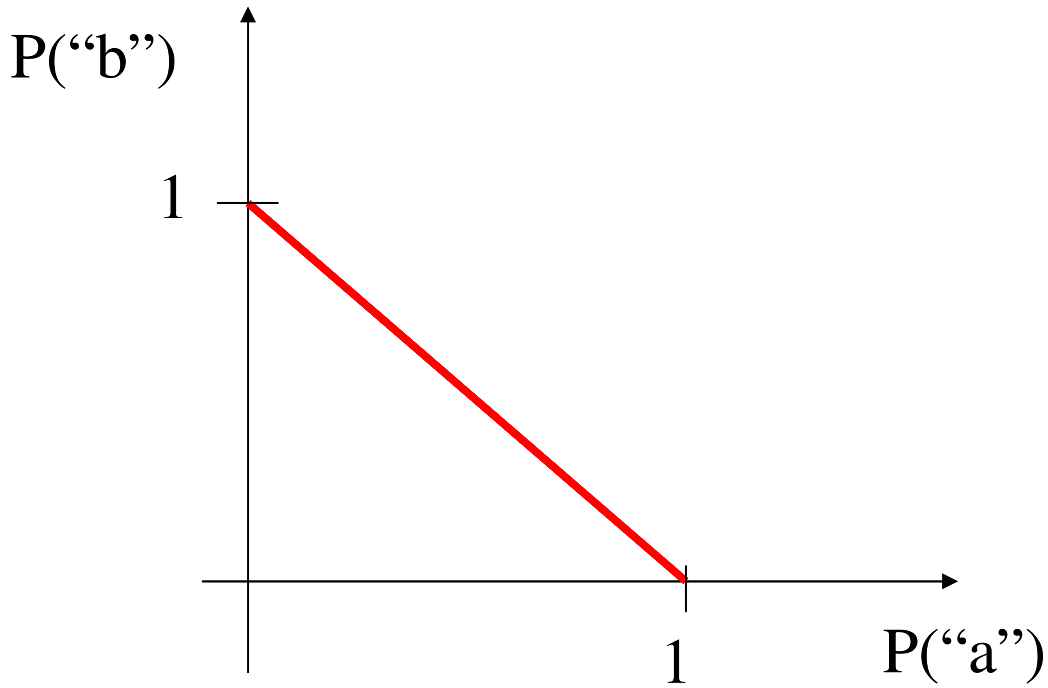
$$C \equiv \{ p \in P \mid p(f_i) = \tilde{p}(f_i) \ \text{for}\ i = 1..n \}$$

P: space of all probability distributions

Question: how does P look like for a probability space consisting of 2 symbols (e.g. the two sides of a coin)

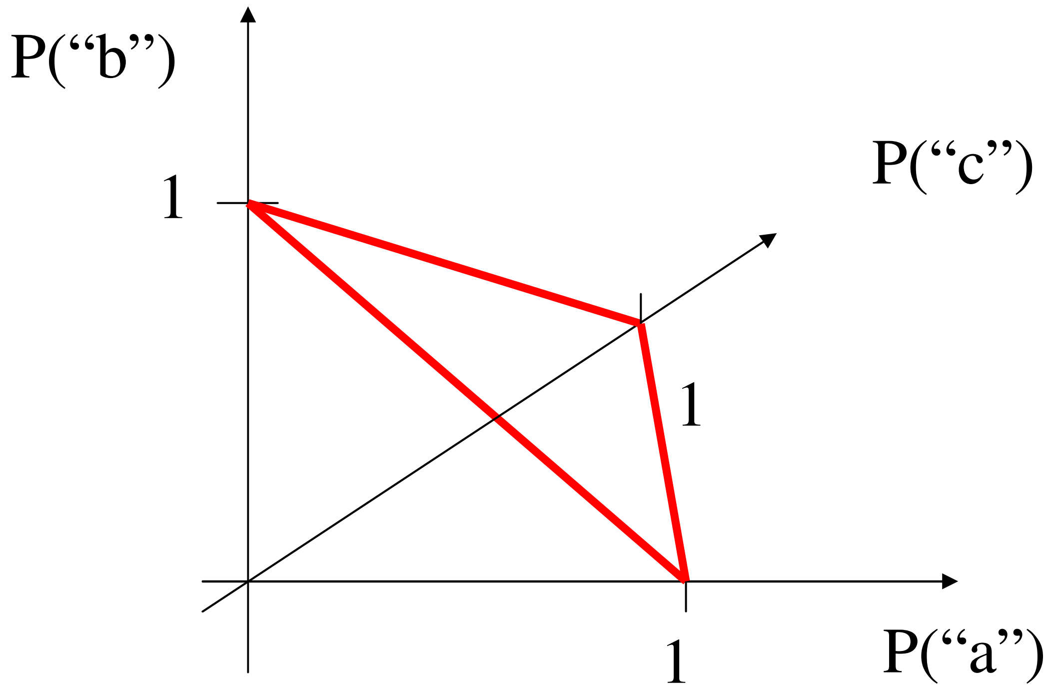
Probability Space as a Simplex

Question: how does P look like for a probability space consisting of 2 symbols (e.g. two sides of a coin)

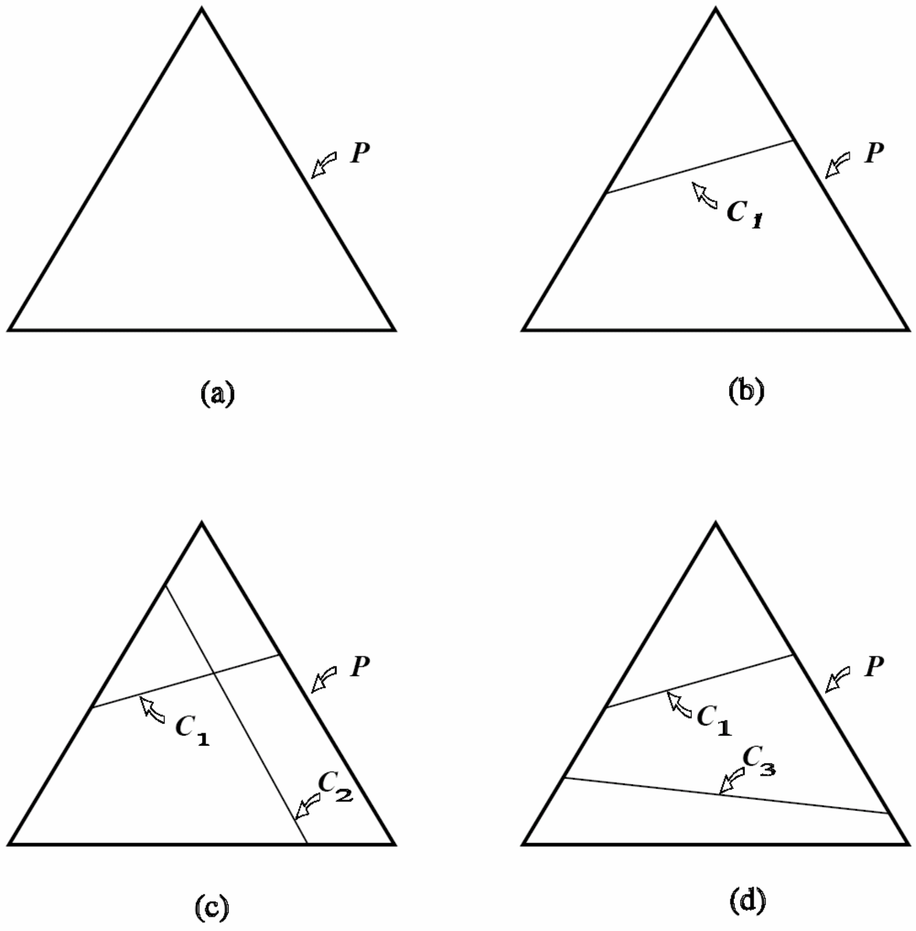


Probability Space as a Simplex

Question: how does P look like for a probability space consisting of 3 symbols (e.g. two sides of a coin)



Examples of Simplex and Constraints



Least biased solution on C

- Entropy:

$$H(p) \equiv -\sum_{x,y} \tilde{p}(x)\, p(y \mid x) \log p(y \mid x)$$

- Maximize entropy

$$\boxed{p_* = \arg\max_{p \in C} H(p)}$$

Example of linear constraints: a trigram language model

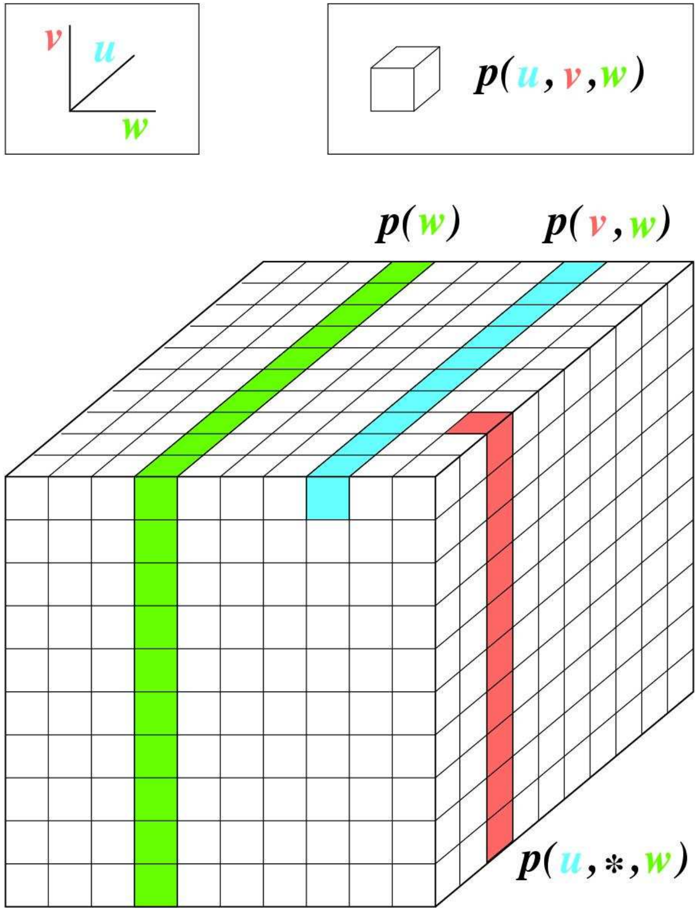
Sequence of words:
u,v,w

Desired probability:

p(w|u,v)

or alternativly

p(u,v,w)



Linear Constraint

w now plays the role of y

The pair u,v plays the role of x

Example feature function:

$$f_{w_k}(x,y) = \begin{cases} 1 & \text{if } \mathrm{y} = \mathrm{w}_\mathrm{k} \\ 0 & \text{otherwise} \end{cases}$$

Resulting Constraint Equation

General constraint equation

$$\sum_{x,y} \tilde{p}(x) p(y \mid x) f_{w_k}(x, y) = \sum_{x,y} \tilde{p}(y, x) f_{w_k}(x, y)$$

Resulting specific constraint equation

$$\sum_{u,v} \tilde{p}(u,v) p(w_k \mid u,v) = \tilde{p}(w_k)$$

Similarly for $\tilde{p}(u_l)$ and $\tilde{p}(v_m)$

Bigram Constraint Equation

Feature function

$$f_{u_l w_k}(x, y) = \begin{cases} 1 & \text{if } \mathrm{y = w_k} \text{ and } \mathrm{u_l} \text{ is directly preceding } \mathrm{w_k} \text{ in } \mathrm{x} \\ 0 & \text{otherwise} \end{cases}$$

Constraint equation

$$\sum_v \tilde{p}(u_l, v)\, p(w_k \mid u_l, v) = \tilde{p}(u_l w_k)$$

Similarly for $\tilde{p}(v_m u_l)$ and $\tilde{p}(v_m * w_k)$

Effective Trigram via Log-Linear Interpolation: Results

Model	**PP**
Bigram	317.7
Linear combination of bigram constraints	302.1
Maximum entropy model (bigram constraints only)	250.1
Trigram	198.4

Training Maximum Entropy Models

Log linear models

- General solution of ME problem:

$$p_\lambda(y \mid x) = \frac{1}{Z_\lambda(x)} \exp\left(\sum_i \lambda_i f_i(x, y) \right)$$

with

λ: parameters still to be determined

Z_λ(h): normalization (calculation costly!!!)

Generalized Iterative Scaling

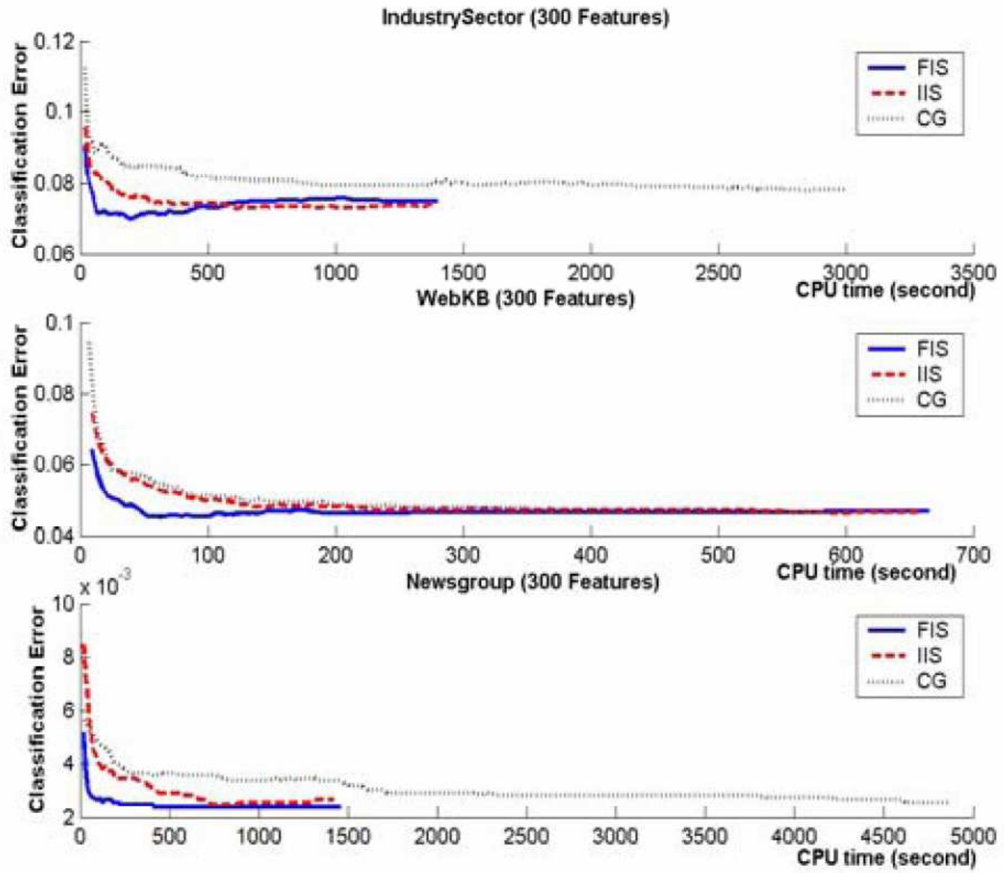
$$\lambda_i^{j+1} = \lambda_i^j + \log\left(\frac{\sum_{x,y} \tilde{p}(x,y) f_i(x,y)}{\sum_{x,y} \tilde{p}(x) p_j(y \mid x) f_i(x,y)}\right)^{e_i}$$

- e_i: scaling of constraint
- A few iterations are sufficient
- Takes quite a lot of CPU time

Alternative Training Schemes

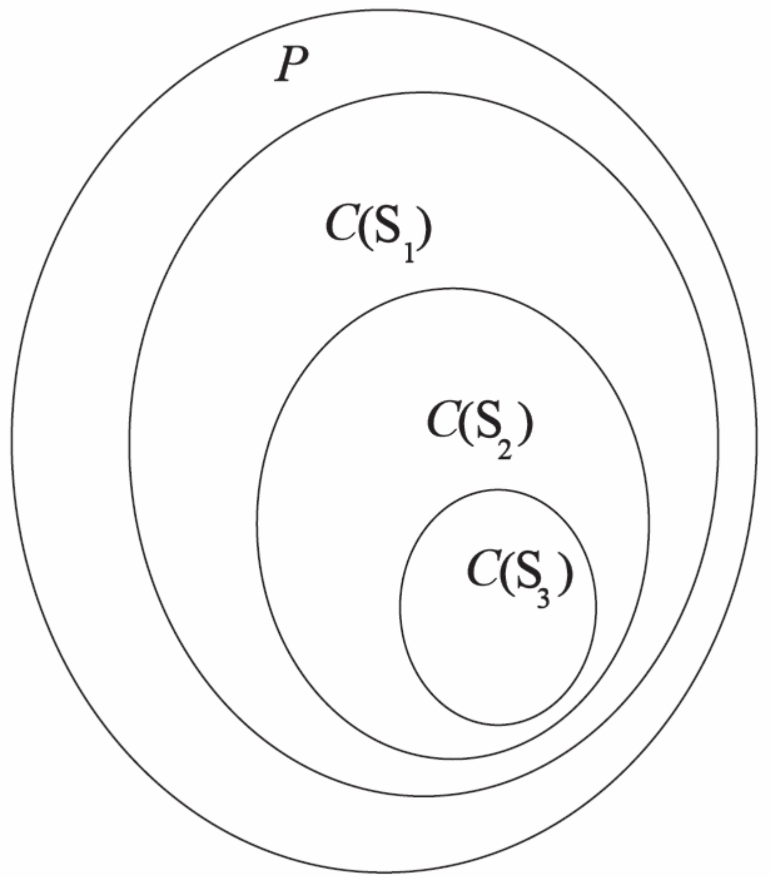
- Improved iterative scaling
- Conjugate gradient
- Fast iterative scaling
- …

Convergence in a Text Classification Task



GIS: not shown on this graph because it has been shown in older publications that IIS is faster

Selecting Feature Set



- Measure change in likelihood when adding a feature
- Slow and expensive process
- No standard solution yet

Other Applications of Max.-Ent. Models

Machine Translation

Translation from a French sentence F to an English sentence E

$$P(F, A \mid E) = \prod_{i=1}^{|E|} p(n(e_i) \mid e_i) \prod_{j=1}^{|F|} p(y_i \mid e_{a_j})\, p(A \mid E, F)$$

with

$p(n \mid e)$: number of French words generated from English word e

$p(f \mid e)$: probability that French word f is generated by e

$p(A \mid E, F)$: probability of particular word order

Text-Classification on Reuters Task

Features

Word w^i	Feature weight α_i	$\log_e \alpha_i$
vs	2.696	0.992
mln	1.079	0.076
cts	12.303	2.510
;	0.448	−0.803
&	0.450	−0.798
000	0.756	−0.280
loss	4.032	1.394
'	0.993	−0.007
"	1.502	0.407
3	0.435	−0.832
profit	9.701	2.272
dlrs	0.678	−0.388
1	1.193	0.177
pct	0.590	−0.528
is	0.418	−0.871
s	0.359	−1.025
that	0.703	−0.352
net	6.155	1.817
lt	3.566	1.271
at	0.490	−0.713
f_{K+1}	0.967	−0.034

Results

"earnings" assigned?	"earnings" correct? YES	NO
YES	1014	53
NO	73	2159

96,2% accurate

Question Answering

- Features:

```
Question word who ↦ Answer candidate is person
Question word who ↦ Answer candidate has two words
Question word where  ↦ Answer candidate is location
…
```

Named Entity Tagging

See:

Maximum Entropy Models for **Named Entity** Recognition

O. Bender, F.J. Och, H. **Ney**

Proceedings of CoNLL-2003

Probabilistic Context Free Grammars

See:

A maximum-entropy-inspired parser

E. Charniak –

Proceedings of NAACL, 2000

Summary

- General framework to train probabilities
  - Include constraints (i.e. observations from corpus)
  - Find least biased probability distribution satisfying all constraints
- Warning:
  - CPU-time intensive
  - Picking the right features important for success