# Maximum Entropy Models (for tagging)

Stephen Clark

Lent 2013

Machine Learning for Language Processing: Lecture 2

MPhil in Advanced Computer Science

# Discriminative Models

- Classification requires the class-posterior $P(\omega_j|\boldsymbol{x})$
  - can just directly model the posterior distribution
  - avoids the complexity of modelling the joint distribution $P(\boldsymbol{x}, \omega_j)$

- Form of model called a discriminative model

- Many debates of generative versus discriminative models:
  - discriminative model criterion more closely related to classification process
  - not dependent on generative process being correct
  - joint distribution can be very complicated to accurately model
  - only final posterior distribution needs to be a valid distribution

# Recap on Tagging

- Find the best tag sequence *given the sentence* (conditional probability):

$$\operatorname*{argmax}_{t_1 \ldots t_n} p(t_1 \ldots t_n | w_1 \ldots w_n)$$

- Alternatively maximise $p(t_1 \ldots t_n, w_1 \ldots w_n)$ (joint probability):

$$\begin{aligned}
\operatorname*{argmax}_{t_1 \ldots t_n} p(t_1 \ldots t_n | w_1 \ldots w_n) &= \operatorname*{argmax}_{t_1 \ldots t_n} \frac{p(t_1 \ldots t_n, w_1 \ldots w_n)}{p(w_1 \ldots w_n)} \\
&= \operatorname*{argmax}_{t_1 \ldots t_n} p(t_1 \ldots t_n, w_1 \ldots w_n)
\end{aligned}$$

# Recap on Markov Model Tagging

- Maximise the joint probability:

$$p(t_1 \dots t_n, w_1 \dots w_n) = p(t_1 \dots t_n) p(w_1 \dots w_n | t_1 \dots t_n)$$

- Tag sequence probability (first order Markov Model):

$$p(t_1 \dots t_n) \approx p(t_1) p(t_2 | t_1) p(t_3 | t_2) \cdots p(t_n | t_{n-1})$$

- Word sequence probability (given the tags):

$$p(w_1 \dots w_n | t_1 \dots t_n) \approx p(w_1 | t_1) p(w_2 | t_2) \cdots p(w_n | t_n)$$

# Problems with Markov Model Taggers

- unreliable zero or very low counts
  - does a zero count indicate an impossible event?

  $\Longrightarrow$ *smoothing* the counts solves this problem

- Words not seen in the data are especially problematic
  $\Longrightarrow$ would like to include word internal information
  e.g. capitalisation or suffix information

- Cannot incorporate diverse pieces of evidence for predicting tags
  e.g. global document information

# Feature-based Models

- Features encode evidence from the context for a particular tag:

| | |
|---|---|
| (title caps, `NNP`) | Citibank, Mr. |
| (suffix `-ing`, `VBG`) | running, cooking |
| | |
| (next word `Inc.`, `I-ORG`) | Lotus Inc. |
| (previous word `said`, `I-PER`) | said Mr. Vinken |

# Complex Features

- Features can be arbitrarily complex
  - e.g. document level features
    (document = `cricket` & current word = `Lancashire`, `I-ORG`)
    $\Longrightarrow$ hopefully tag `Lancashire` as `I-ORG` not `I-LOC`

- Features can be combinations of atomic features
  - (current word = `Miss` & next word = `Selfridges`, `I-ORG`)
    $\Longrightarrow$ hopefully tag `Miss` as `I-ORG` not `I-PER`

- Features are not assumed to be (conditionally) independent (given the label)
  – unlike the Naive Bayes classifier

# Feature-based Tagging

- How do we incorporate features into a probabilistic tagger?
- Hack the Markov Model tagger to incorporate features
- Maximum Entropy (MaxEnt) Tagging
  - principled way of incorporating features
  - requires sophisticated estimation method

# Features in Maximum Entropy Models

- Features encode elements of the context $C$ useful for predicting tag $t$

- Features are binary valued functions, e.g.

$$f_i(C,t) = \begin{cases} 1 & \text{if } \texttt{word}(C) = \texttt{Moody} \ \ \& \ \ t = \text{I-ORG} \\ 0 & \text{otherwise} \end{cases}$$

- $\texttt{word}(C) = \texttt{Moody}$ is a *contextual predicate*

- Features determine (`contextual_predicate, tag`) pairs

# The Model

$$p(t|C) = \frac{1}{Z(C)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C,t)\right)$$

- $f_i$ is a feature
- $\lambda_i$ is a weight (large value implies informative feature)
- $Z(C)$ is a normalisation constant ensuring a proper probability distribution
- Also known as a *log-linear* model
- Makes no independence assumptions about the features
- Can be used as a general classifer (outside of tagging, e.g. text classification)

# Tagging with Maximum Entropy Models

- The conditional probability of a tag sequence $t_1 \dots t_n$ is

$$p(t_1 \dots t_n | w_1 \dots w_n) \approx \prod_{i=1}^{n} p(t_i | C_i)$$

given a sentence $w_1 \dots w_n$ and contexts $C_1 \dots C_n$

- The context includes previously assigned tags (for a fixed history)

- Beam search or Viterbi is used to find the most probable sequence (Ratnaparkhi, 1996)

- Later in the course we will see an alternative (more principled) conditional formulation of the global probability (in the form of CRFs)

# Model Estimation

$$p(t|C) = \frac{1}{Z(C)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C,t)\right)$$

- Model estimation involves setting the weight values $\lambda_i$
- The model should reflect the data
  $\Longrightarrow$ use the data to $constrain$ the model
- What form should the constraints take?
  $\Longrightarrow$ constrain the $expected\ value$ of each feature $f_i$

# The Constraints

$$E_p f_i = \sum_{C,t} p(C,t) f_i(C,t) = K_i$$

- Expected value of each feature must satisfy some constraint $K_i$
- A natural choice for $K_i$ is the average empirical count:

$$K_i = \quad E_{\tilde{p}} f_i = \frac{1}{N} \sum_{j=1}^{N} f_i(C_j, t_j)$$

derived from the training data $(C_1, t_1), \ldots, (C_N, t_N)$

# Choosing the Maximum Entropy Model

- The constraints do not *uniquely* identify a model

- From those models satisfying the constraints:
  *choose the Maximum Entropy model*

- Conditional entropy of a model $p$:

$$H(p) = -\sum_{C,t} \tilde{p}(C)p(t|C)\log p(t|C)$$

# The Maximum Entropy Model

- The maximum entropy model is the *most uniform model*
  $\Longrightarrow$ makes no assumptions in addition to what we know from the data
- MaxEnt model is also the *Maximum Likelihood Log-Linear* model
- Set the weights to give the MaxEnt model satisfying the constraints
  $\Longrightarrow$ use *Generalised Iterative Scaling* (GIS)

# Generalised Iterative Scaling (GIS)

- Set $\lambda_i^{(0)}$ equal to some arbitrary value (e.g. zero)

- Repeat until convergence:

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} + \frac{1}{C} \log \frac{E_{\tilde{p}} f_i}{E_{p^{(t)}} f_i}$$

  where

$$C = \max_{x,y} \sum_{i=1}^{n} f_i(x, y)$$

- Many formulations of GIS specify the need for a "correction feature", but see Curran and Clark (2003)

# Smoothing

- Models which satisfy the constraints exactly tend to *overfit* the data
- In particular, empirical counts for low frequency features can be unreliable
  - often leads to very large weight values
- Common smoothing technique is to ignore low frequency features
  - but low frequency features may be important
- Use a *prior* distribution on the parameters
  - encodes our knowledge that weight values should not be too large

# Smoothing

- Standard technique is to use a $Gaussian\ prior$ over the parameters (Chen and Rosenfeld 1999)
  - penalises models with extreme feature weights
- This is a form of *maximum a posteriori* (MAP) estimation
- Can be thought of as relaxing the model constraints - requires a modification to the update rule
- Can also be thought of as a form of *regularisation*

# Pos Tagger Features

- The tagger uses binary valued features, e.g.

$$f_i(x,y) = \begin{cases} 1 & \text{if } \texttt{word}(x) = \texttt{the} \;\; \& \;\; y = \text{DT} \\ 0 & \text{otherwise} \end{cases}$$

- $\texttt{word}(\mathrm{x}) = \texttt{the}$ is a *contextual predicate*
- Contextual predicates:

| | |
|---|---|
| $t_{i-1} = X$ | previous tag history |
| $t_{i-2}t_{i-1} = XY$ | previous two tags history |
| $w_i = X$ | current word |
| $w_{i-1} = X$ | previous word |
| $w_{i-2} = X$ | previous previous word |
| $w_{i+1} = X$ | next word |
| $w_{i+2} = X$ | next next word |

# Pos Tagger Features for Rare Words

- These predicates apply to words seen less than 5 times in the data

$X$ is prefix of $w_i$, $|X| \leq 4$
$X$ is suffix of $w_i$, $|X| \leq 4$
$w_i$ contains a digit
$w_i$ contains uppercase char
$w_i$ contains a hyphen

- Otherwise the current word predicate applies

# Evaluation Measures

**Acc** overall per-word accuracy

**Uword** accuracy on previously unseen words

**Utag** accuracy on previously unseen word-tag pairs

**Amb** accuracy on words seen with more than one tag in the Treebank

- Training data sections 2-21, development section 00, testing section 23 from the WSJ Penn Treebank

# Results on the Development Set

| Tagger | Acc | Uword | Utag | Amb |
|---|---|---|---|---|
| MXPOST | 96.59 | 85.81 | 30.04 | 94.82 |
| base | 96.58 | 85.70 | 29.28 | 94.82 |
| smoothed | **96.75** | **86.74** | **33.08** | **95.06** |

- MXPOST is Ratnaparkhi's original tagger (feature cutoff 5, no smoothing)
- Gaussian smoothing improves results

# Results with varying feature cut-offs

| Cut-off | ACC | UWORD | UTAG | AMB |
|---|---|---|---|---|
| $\geq 1$ | **96.82** | **87.20** | 30.80 | **95.07** |
| $\geq 2$ | 96.77 | 87.02 | 31.18 | 95.00 |
| $\geq 3$ | 96.72 | 86.62 | 31.94 | 94.94 |
| $\geq 4$ | 96.72 | 87.08 | **34.22** | 94.96 |

- No cutoff gives best results
- Gaussian smoothing allows all features to be used without overfitting

# Results on the Test Set

| Tagger | Acc | Uword | Utag | Amb |
|---|---|---|---|---|
| MXPOST | 97.05 | 83.63 | **30.20** | 95.44 |
| C&C | **97.27** | **85.21** | 28.98 | **95.69** |

# Cross-validation results

| Tagger | Acc | $\sigma$ | Uword | Utag | Amb |
|---|---|---|---|---|---|
| MXPOST | 96.72 | 0.12 | 85.50 | **32.16** | 95.00 |
| TNT | 96.48 | 0.13 | 85.31 | 0.00 | 94.26 |
| C&C | **96.86** | 0.12 | **86.43** | 30.42 | **95.08** |

# Performance

- Training takes around 10 minutes for 100 GIS iterations
- Tagging is very fast (around 100,000 words per second)

# Named Entity Tagging

- Language independent NER for CoNLL-02, CoNLL-03 competitions
- English, German, Dutch
- LOC, PER, ORG, MISC, O

# Contextual Predicates used by the NE tagger

| Condition | Contextual predicate |
|---|---|
| $f(w_i) < 5$ | $X$ is prefix/suffix of $w_i$, $\lvert X \rvert \leq 4$<br>$w_i$ contains a digit<br>$w_i$ contains uppercase character<br>$w_i$ contains a hyphen |
| $\forall w_i$ | $w_i = X$<br>$w_{i-1} = X$, $w_{i-2} = X$<br>$w_{i+1} = X$, $w_{i+2} = X$ |
| $\forall w_i$ | $\text{POS}_i = X$<br>$\text{POS}_{i-1} = X$, $\text{POS}_{i-2} = X$<br>$\text{POS}_{i+1} = X$, $\text{POS}_{i+2} = X$ |
| $\forall w_i$ | $\text{NE}_{i-1} = X$<br>$\text{NE}_{i-2}\text{NE}_{i-1} = XY$ |

# Additional Contextual Predicates

| Condition | Contextual predicate |
|---|---|
| $f(w_i) < 5$ | $w_i$ contains period |
| | $w_i$ contains punctuation |
| | $w_i$ is only digits |
| | $w_i$ is a number |
| | $w_i$ is {upper,lower,title,mixed} case |
| | $w_i$ is alphanumeric |
| | length of $w_i$ |
| | $w_i$ has only Roman numerals |
| | $w_i$ is an initial (X.) |
| | $w_i$ is an acronym (ABC, A.B.C.) |

# Additional Contextual Predicates

| Condition | Contextual predicate |
|---|---|
| $\forall w_i$ | memory NE tag for $w_i$<br>unigram tag of $w_{i+1}$<br>unigram tag of $w_{i+2}$ |
| $\forall w_i$ | $w_i$ in a gazetteer<br>$w_{i-1}$ in a gazetteer<br>$w_{i+1}$ in a gazetteer |
| $\forall w_i$ | $w_i$ not lowercase and $f_{\mathsf{lc}} > f_{\mathsf{uc}}$ |
| $\forall w_i$ | unigrams of word type<br>bigrams of word types<br>trigrams of word types |

# The Word Type Features

- `Moody` $\Longrightarrow$ `Aa`
- `A.B.C.` $\Longrightarrow$ `A.A.A.`
- `1,345.00` $\Longrightarrow$ `0,0.0`
- `Mr. Smith` $\Longrightarrow$ `Aa. Aa`

# Baseline Results on English Data

| English | PRECISION | RECALL | $F_{\beta=1}$ |
|---|---|---|---|
| LOCATION | 90.78% | 90.58% | 90.68% |
| MISC | 85.80% | 81.24% | 83.45% |
| ORGANISATION | 82.24% | 80.09% | 81.15% |
| PERSON | 92.02% | 92.67% | 92.35% |
| OVERALL | **88.53%** | **87.41%** | **87.97%** |

- Reuters newswire data
- 200,000 words training, 50,000 words test

# Full System Results on English Data

| English | PRECISION | RECALL | $F_{\beta=1}$ |
|---|---|---|---|
| LOCATION | 91.75% | 93.20% | 92.47% |
| MISC | 88.34% | 82.97% | 85.57% |
| ORGANISATION | 83.54% | 85.53% | 84.52% |
| PERSON | 94.26% | 95.39% | 94.82% |
| OVERALL | **90.15%** | **90.56%** | **90.35%** |

- Good NER performance requires a wide range of features
- One of the best performing systems in CoNLL-03

# German Results

| German | PRECISION | RECALL | $F_{\beta=1}$ |
|---|---|---|---|
| LOCATION | 70.91% | 71.11% | 71.01% |
| MISC | 68.51% | 46.12% | 55.13% |
| ORGANISATION | 68.43% | 50.19% | 57.91% |
| PERSON | 88.04% | 72.05% | 79.25% |
| OVERALL | **75.61%** | **62.46%** | **68.41%** |

- German newspaper text (200k training, 50k test)
- German is harder than English (capitalisation)

# Conclusion

- Tagging (and other NLP tasks) require a wide range of features for good performance
- Maximum entropy models (with Gaussian smoothing) can handle a large number of diverse features
- GIS is relatively simple and performs well for maximum entropy taggers

# Other Work

- MaxEnt (CRF) models for wide-coverage CCG parsing (Clark & Curran, 2007)
- Statistical parsing requires a wide range of features for good performance
- Generative parsing models lack the flexibility of maximum entropy models
- Training is computationally expensive and requires dynamic programming methods
- GIS is too slow for parsing models - use more general numerical optimisation methods

# References


- Investigating GIS and Smoothing for Maximum Entropy Taggers. James R. Curran and Stephen Clark. Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics (EACL-03), pp.91-98, Budapest, Hungary, 2003

- Adwait Ratnaparkhi. (1996). A Maximum Entropy Model for Part-Of-Speech Tagging. In Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP), University of Pennsylvania.

- Stanley F. Chen and Ronald Rosenfeld. A Gaussian Prior for Smoothing Maximum Entropy Models. Technical Report CMU-CS-99-108, Computer Science Department, Carnegie Mellon University, 1999.

- Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.