Sub-sentence Level Topic Classification

A Semi-discriminative Approach for a Small Dataset

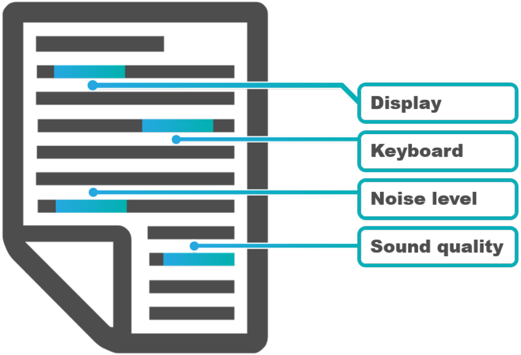
Cornelia Ferner and Stefan Wegenkittl

September 18, 2019

ECMLPKDD
Würzburg 2019

FH Salzburg
Informationstechnik &
System-Management

Problem Statement



Decomposing a detailed expert product review into sections discussing different ”topics”. The sample dataset[1] is about laptops with 17 predefined topics.

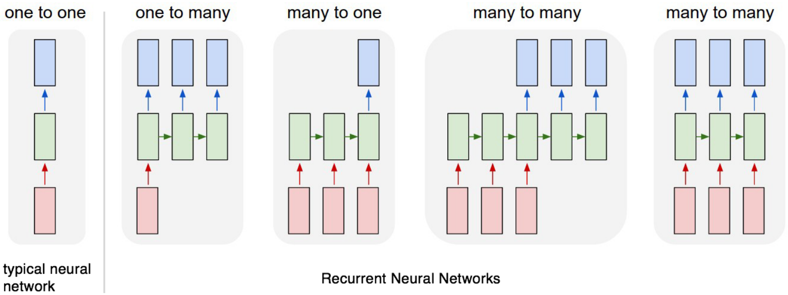
[1] https://github.com/factai/corpus-laptop-topic

Problem Statement

- Task: identify topics in laptop product reviews
- Topics are either **laptop parts** (e.g. display, keyboard), **parameters** (e.g. performance, battery) or **review specifics** (e.g. introduction, verdict)
- We define this as a **sequence classification task**:
    - $\neq$ document classification, because we have more than one topic per document
    - $\neq$ unsupervised topic detection, because we pre-define the topics we are looking for and have an annotated dataset

Recurrent Neural Networks

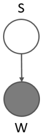
RNN, LSTM, GRU



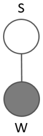
Recurrent networks process sequences of vectors: either in the input or in the output or both [2].

[2] http://karpathy.github.io/2015/05/21/rnn-effectiveness

Sentence Classification



(a) Naive Bayes
generative



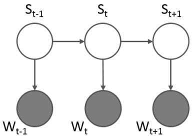
(b) Logistic Regression
discriminative

Comparison of the generative Naive Bayes and the discriminative multinomial Logistic Regression classifier. For NLP tasks, the discriminative classifier has been shown to outperform the generative one `[Klein2002]`.

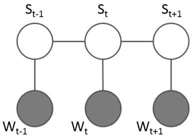
$$P(S, W) = P(W \mid S) \cdot P(S) \qquad P(S = j \mid W) = \frac{1}{Z} \prod_{t=1}^{n} e^{\lambda_{w_t j} + \mu_j}$$

Multinomial Logistic Regression: resulting model has highest entropy → **Maximum Entropy** classifier (MaxEnt, ME)

Sequence Classification



(a) Hidden Markov Model
generative



(b) Linear-chain CRF
discriminative

Comparison of the Hidden Markov Model and the (linear-chain) Conditional Random Field `[Lafferty2001]`.

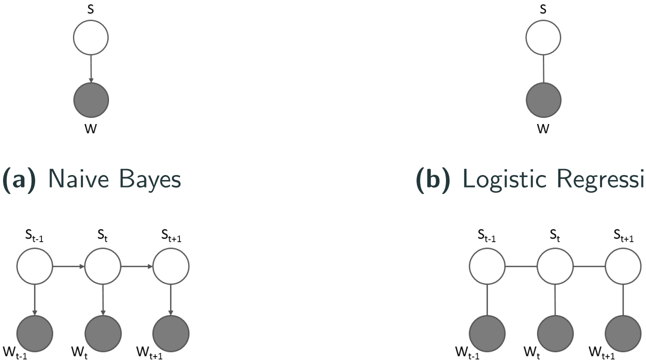
$$P\left(\boldsymbol{S}, \boldsymbol{W}\right) = \prod_{t=1}^{n} P\left(S_{t+1} \mid S_t\right) \cdot P\left(W_t \mid S_t\right)$$

$$P\left(\boldsymbol{S} \mid \boldsymbol{W}\right) = \frac{1}{Z_W} \prod_{t=1}^{n} e^{\sum_i \lambda_i f_i(S_t, S_{t-1}, W})$$

Method Comparison

generative

discriminative



(a) Naive Bayes

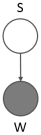
(b) Logistic Regression

(c) Hidden Markov Model

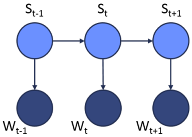
(d) Linear-chain CRF

Method Comparison

generative

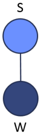


(a) Naive Bayes

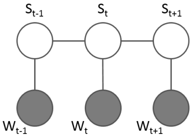


(c) Hidden Markov Model

discriminative



(b) Logistic Regression



(d) Linear-chain CRF

MaxEnt Classifier

Dictionary and Topics

- $C = \{1, \ldots, c\}$ ... set of topics
- $D = \{1, \ldots, d\}$ ... set of words (dictionary)
- $\boldsymbol{W} = (w_1, \ldots, w_n)$ ... input sequence of words
- $\boldsymbol{S} = (s_1, \ldots, s_n)$ ... sequence of topics

MaxEnt Classifier

Dictionary and Topics

- $C = \{1, \ldots, c\}$ ... set of topics
- $D = \{1, \ldots, d\}$ ... set of words (dictionary)
- $\boldsymbol{W} = (w_1, \ldots, w_n)$ ... input sequence of words
- $\boldsymbol{S} = (s_1, \ldots, s_n)$ ... sequence of topics

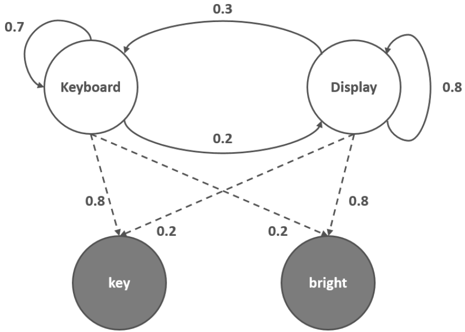
Maximum Entropy

$$P\left(S_1 = j, \ldots, S_n = j \mid W_1 = w_1, \ldots, W_n = w_n\right) = \frac{1}{Z_{ME}} \prod_{t=1}^{n} e^{\lambda_{w_t j} + \mu_j} \tag{1}$$

Hidden Markov Model

Hidden Markov Model

- transition probabilities:
  $A = a_{ij} = P\left(S_t = j \mid S_{t-1} = i\right)$
- emission probabilities:
  $B = b_{jk} = P\left(W_t = k \mid S_t = j\right)$
- initial state probabilities:
  $\pi_i = P\left(S_1 = i\right)$
- $M = (C, D, A, B, \pi)$



A minimal HMM example.

ME+HMM

Using a stationary HMM for generating the words:

$$P\left(\bar{W} = \boldsymbol{W}, \bar{S} = \boldsymbol{S}\right) = \prod_{t=1}^{n} \underbrace{P\left(W_t = w_t \mid S_t = s_t\right)}_{b_{w_t s_t}} \cdot \underbrace{P\left(S_t = s_t \mid S_{t-1} = s_{t-1}\right)}_{a_{s_{t-1} s_t}} \tag{2}$$

ME+HMM

Using a stationary HMM for generating the words:

$$P\left(\bar{W} = \boldsymbol{W}, \bar{S} = \boldsymbol{S}\right) = \prod_{t=1}^{n} \underbrace{P\left(W_t = w_t \mid S_t = s_t\right)}_{b_{w_t s_t}} \cdot \underbrace{P\left(S_t = s_t \mid S_{t-1} = s_{t-1}\right)}_{a_{s_{t-1} s_t}} \qquad (2)$$

MaxEnt assumes independency of the words. Assuming this for the HMM, too, gives $a_{s_{t-1},s_t} = a_{s_t} = P(S_t = s_t)$:

$$P\left(\bar{W} = \boldsymbol{W}, \bar{S} = \boldsymbol{S}\right) = \prod_{t=1}^{n} b_{w_t,s_t} \cdot a_{s_t} \qquad (3)$$

ME+HMM

Using a stationary HMM for generating the words:

$$P\left(\bar{W} = \boldsymbol{W}, \bar{S} = \boldsymbol{S}\right) = \prod_{t=1}^{n} \underbrace{P\left(W_t = w_t \mid S_t = s_t\right)}_{b_{w_t s_t}} \cdot \underbrace{P\left(S_t = s_t \mid S_{t-1} = s_{t-1}\right)}_{a_{s_{t-1} s_t}} \tag{2}$$

MaxEnt assumes independency of the words. Assuming this for the HMM, too, gives $a_{s_{t-1},s_t} = a_{s_t} = P(S_t = s_t)$:

$$P\left(\bar{W} = \boldsymbol{W}, \bar{S} = \boldsymbol{S}\right) = \prod_{t=1}^{n} b_{w_t,s_t} \cdot a_{s_t} \tag{3}$$

Dividing by $P(\bar{W} = \boldsymbol{W}) = Z_{HMM}$ yields

$$P(\bar{S} = \boldsymbol{S} \mid \bar{W} = \boldsymbol{W}) = \frac{1}{Z_{HMM}} \prod_{t=1}^{n} b_{w_t,s_t} \cdot a_{s_t} \tag{4}$$

ME+HMM

Let $s_t = j \forall t \in \{1, \ldots, n\}$ and set equations (1) and (4) equal:

$$\frac{1}{Z_{ME}} \prod_{t=1}^{n} e^{\lambda_{w_t j} + \mu_j} = \frac{a_j^n}{Z_{HMM}} \prod_{t=1}^{n} b_{w_t j} \tag{5}$$

ME+HMM

Let $s_t = j \forall t \in \{1, \dots, n\}$ and set equations (1) and (4) equal:

$$\frac{1}{Z_{ME}} \prod_{t=1}^{n} e^{\lambda_{w_t j} + \mu_j} = \frac{a_j^n}{Z_{HMM}} \prod_{t=1}^{n} b_{w_t j} \tag{5}$$

Rewriting the equation and solving for a single emission probability b_{jk} gives:

$$b_{jk} = e^{\lambda_{kj} + \mu_j} \cdot \frac{Z_{HMM}}{Z_{ME} a_j} = e^{\lambda_{kj} + \mu_j} \cdot \frac{P(\bar{W} = \boldsymbol{w})}{Z_{ME} a_j} \tag{6}$$

Training

Implementation:

1. Train the MaxEnt classifier on the training data on sentence-level. This yields the λ_{kj}.

Training

Implementation:

1. Train the MaxEnt classifier on the training data on sentence-level. This yields the λ_{kj}.
2. Compute the HMM emission probabilities b_{jk}:
   a) Estimate the overall word frequency $\hat{p}_w$ from the training data set by counting them.
   b) Substitute Z_{HMM} by $\hat{p}_w$.
   c) Normalize with respect to $\sum_{i=1}^{d} b_{jk} = 1$ instead of dividing by $Z_{ME} a_j$.

Training

Implementation:

1. Train the MaxEnt classifier on the training data on sentence-level. This yields the λ_{kj}.
2. Compute the HMM emission probabilities b_{jk}:
   a) Estimate the overall word frequency $\hat{p}_w$ from the training data set by counting them.
   b) Substitute Z_{HMM} by $\hat{p}_w$.
   c) Normalize with respect to $\sum_{i=1}^{d} b_{jk} = 1$ instead of dividing by $Z_{ME} a_j$.
3. Estimate the HMM transition probabilities a_{ij} from the training data (apply smoothing).
4. Estimate the initial probabilities π_i from the training data.

Training

Implementation:

1. Train the MaxEnt classifier on the training data on sentence-level. This yields the λ_{kj}.
2. Compute the HMM emission probabilities b_{jk}:
   a) Estimate the overall word frequency $\hat{p}_w$ from the training data set by counting them.
   b) Substitute Z_{HMM} by $\hat{p}_w$.
   c) Normalize with respect to $\sum_{i=1}^{d} b_{jk} = 1$ instead of dividing by $Z_{ME} a_j$.
3. Estimate the HMM transition probabilities a_{ij} from the training data (apply smoothing).
4. Estimate the initial probabilities π_i from the training data.

→ Apply the model $M = (C, D, A, B, \pi)$ to the test data set (word-level).

Testing

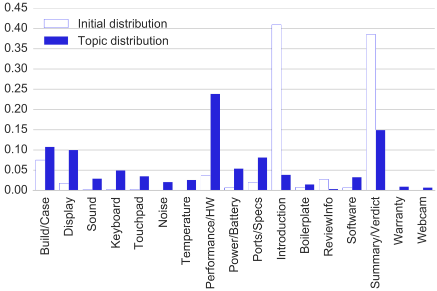
M is used to decode the observed sequence W_t by assigning the most likely sequence of topics S_t^*:

- Viterbi decoding:
  - globally optimal solution
  - compute $S_t^* = \arg\max_S P\left(W, S\right)$
- Posterior decoding:
  - uses the forward-backward algorithm
  - locally optimal solution
  - compute $S_t^* = \{s_i \mid s_i = \arg\max_k \sum_S P(s_i = k|W)\}$

Laptop Review Dataset

Laptop review dataset[3]:

- 3076 reviews annotated at sentence level
- 240 146 sentences with topic label
- 17 topics
- average review length: 78 sentences



The relative distribution on sentence-level of the 17 review topics and the topics' likeliness to be the first in a review.

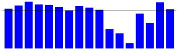
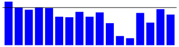
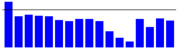
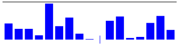
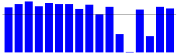
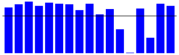
[3] https://github.com/factai/corpus-laptop-topic

Classification Results

Algorithm	Accuracy	Precision	Recall	F1 score
MaxEnt	70.00%	71.46%	70.00%	70.13%
HMM (Viterbi)	60.16%	68.92%	60.16%	61.89%
HMM (posterior)	53.60%	68.59%	53.60%	56.95%
CRF	39.86%	49.63%	39.86%	40.08%
ME+HMM (Viterbi)	75.41%	77.40%	75.41%	74.30%
ME+HMM (posterior)	**76.84**%	**78.74**%	**76.84**%	**75.62**%

Results for all classifiers on the given laptop review dataset using 5-fold cross validation.
Accuracy, precision, recall and F1 score are weighted by the number of sentences in each topic.

Classification Results

Algorithm	Accuracy per Class
MaxEnt	
HMM (Viterbi)	
HMM (posterior)	
CRF	
ME+HMM (Viterbi)	
ME+HMM (posterior)	

Sparklines indicating the accuracy results for each topic. Each horizontal line denotes the baseline MaxEnt accuracy of 70%.

Example

(Gold labels)

Otherwise, the approx. 3.3 kilogram heavy case didn't actually knock our socks off: design, workmanship and materials are only second rate. **The input devices could also be a lot better (small touchpad, clattery keyboard, single-rowed enter, etc.).** *The main point of complaint is the enormous noise development, typical for a gamer: the fan is clearly audible during load.*

(ME+HMM)

Otherwise, the approx. 3.3 kilogram heavy case didn't actually knock our socks off: design, workmanship and materials are only second **rate. The input devices** `could also be a lot better (small touchpad`**, clattery keyboard, single-rowed enter, etc.). The main point** *of complaint is the enormous noise development, typical for a gamer: the fan is clearly audible during load.*

Build/Case	**Keyboard**	`Touchpad`	*Noise*

A sample sequence taken randomly from a review. The gold labeling suggests three different topics (top), the ME+HMM model assigns four topics (bottom).

Thank you for your attention!

Literature

Lafferty2001 Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282-289. (2001)

McCallum2000 McCallum, A., Freitag, D., Pereira, F.C.N.: Maximum Entropy Markov Models for Information Extraction and Segmentation. In: Proceedings of the 17th International Conference on Machine Learning. pp. 591-598. (2000)

Ng2002 Ng, A.Y., Jordan, M.I.: On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. In: NeurIPS. pp. 841-848 (2002)

Petrushin2000 Petrushin, V.A.: Hidden Markov Models: Fundamentals and Applications. In: Online Symposium for Electronics Engineer (2000)

Sutton2012 Sutton, C., McCallum, A., et al.: An Introduction to Conditional Random Fields. Foundations and Trends in Machine Learning 4(4), 267-373 (2012)

Dai2015 Dai, A.M., Le, Q.V.: Semi-supervised Sequence Learning. In: Advances in Neural Information Processing Systems. pp. 3079-3087 (2015)

Klein2002 Klein, D., Manning, C.D.: Conditional Structure versus Conditional Estimation in NLP Models. In: Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing. pp. 9-16. EMNLP '02, ACL (2002)

Medlock2008 Medlock, B.W.: Investigating Classification for Natural Language Processing Tasks. Technical report, University of Cambridge, Computer Laboratory (2008)

Side Note: Maximum Entropy Markov Model

Maximum Entropy Markov Model (MEMM) `[McCallum2000]`:

- models probability of current state s_t based on the previous state s_{t-1} and the current observation w_t
- normalization per state
- label bias problem

Side Note: Maximum Entropy Markov Model

HMM: $$P(S, W) = \prod_{t=1}^{n} P(s_t \mid s_{t-1}) P(w_t \mid s_t)$$

MEMM: $$P(S \mid W) = \prod_{t=1}^{n} P(s_t \mid s_{t-1}, w_t) =$$

$$\prod_{t=1}^{n} \frac{1}{Z_{s_{t-1}, w_t}} \exp\left(\sum_i \lambda_i f_i (s_t, s_{t-1}, w_t)\right)$$

CRF: $$P(S \mid W) =$$

$$\frac{1}{Z_W} \prod_{t=1}^{n} \exp\left(\sum_i \lambda_i f_i (s_t, s_{t-1}, W)\right)$$

Side Note: Multinomial Logistic Regression

Multinomial logistic regression:

- ANN without hidden layer and a logistic transfer function followed by a softmax in the output nodes
- Loss function = cross entropy loss
- Stochastic gradient descent for optimization
- Minimizing cross entropy is equivalent to maximizing log-likelihood
- Resulting model has highest entropy
- **Maximum Entropy classifier** (MaxEnt or ME)

Topic Separability

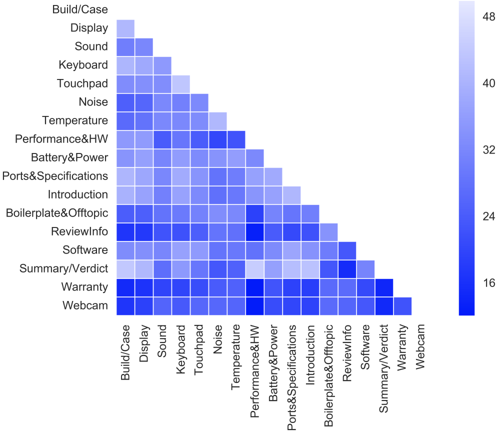
Classification accuracy correlates with class separability.

Percent vocabulary overlap (PVO) measures the amount of vocabulary terms shared by two classes `[Medlock2008]`:

$$PVO\left(S_1, S_2\right) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} \cdot 100 \tag{7}$$

T_i denotes the set of terms in topic S_i.

Topic Separability



PVO between all topics. The diagonal (topic-topic) comparison is 100%. The average PVO is 33.22%.

Evaluation of the MaxEnt Output

MaxEnt			
Sound	Noise	Temperature	Summary
sound	db	cool	verdict
speech	quiet	heat	quietly
bass	noise	hot	lasts
volume	fan	lap	drawbacks
speaker	silent	temperatures	flaws
audio	hear	thighs	compromises
speakers	audible	warm	recommend
headphones	noisy	heats	price
sounded	noiseless	warmer	money
equalizer	fans	warmth	conclusion

Ten highest scoring terms in four exemplary topics when based on MaxEnt weights.