





A Semi-discriminative Approach for Sub-sentence Level Topic Classification on a Small Dataset

Cornelia Ferner^(✉) and Stefan Wegenkittl

Salzburg University of Applied Sciences, Urstein Sued 1, 5412 Puch, Salzburg, Austria
{cornelia.ferner,stefan.wegenkittl}@fh-salzburg.ac.at

Abstract. This paper aims at identifying sequences of words related to specific product components in online product reviews. A reliable baseline performance for this topic classification problem is given by a Max Entropy classifier which assumes independence over subsequent topics. However, the reviews exhibit an inherent structure on the document level allowing to frame the task as sequence classification problem. Since more flexible models from the class of Conditional Random Fields were not competitive because of the limited amount of training data available, we propose using a Hidden Markov Model instead and decouple the training of transition and emission probabilities. The discriminating power of the Max Entropy approach is used for the latter. Besides outperforming both standalone methods as well as more generic models such as linear-chain Conditional Random Fields, the combined classifier is able to assign topics on sub-sentence level although labeling in the training data is only available on sentence level.

Keywords: Small data · Topic classification · Hidden Markov Model

1 Introduction

Product comparison websites provide detailed product reviews (further on referred to as “expert” reviews) that usually differ from popular webshops’ user reviews in length, quality and focus. A more concise representation of such expert reviews can be obtained for instance by automated text summarization or aspect-based sentiment analysis. A required subtask is to identify topics discussed in the reviews. The specific task is to assign a set of predefined topics to the sections of laptop reviews, where topics might be product components (e.g. *display*, *keyboard*, *performance*) or review sections (e.g. *introduction*, *verdict*).

A common approach to solve this task is to use a Max Entropy (MaxEnt) classifier which has been proven useful in a series of language classification tasks such as sentiment analysis [12]. However, the expert reviews exhibit some high-level structure on document level such as treating each topic one after another, without changing back and forth, or starting with an introduction and ending with a verdict. In order to exploit the reviews’ structure, the task of assigning

topics is defined as sequence classification task in this paper. This differs from what is known as document classification, as more than one topic per document is assigned. It also differs from unsupervised topic modeling, as the topics of interest are predefined in a labeled dataset with a label assigned to each sentence.

As the MaxEnt is trained to assign one label per sentence, it has no “memory” to recall decisions on previous sentences in the review. Thus, a sequence model such as the Hidden Markov Model (HMM) would be beneficial. An HMM can capture the reviews’ inherent topic patterns by assigning labels at the word-level where topic changes are infrequent. This allows for a more fine-grained labeling even at sub-sentence-level.

The drawback of the HMM is its generative nature. A generative classifier maximizes the joint probabilities $P(w, s) = P(s) P(w | s)$ over the observed input words w and the state labels s . Given label s , the probabilities over the input features $P(w | s)$ need to be generated. Discriminative models such as MaxEnt directly train the conditional probability $P(s | w)$ without the need for modeling $P(w)$ which is considered given in the classification task. The question now arises if it is possible to have a sequence model where the relation between states and observations are modeled by a MaxEnt classifier.

We show that the proposed method of combining the benefits of the HMM with the discriminative power of a MaxEnt classifier successfully solves the sequence classification problem: After a trained MaxEnt classifier has learned to maximally separate the topics’ probability distributions, we transform the MaxEnt based weights into HMM emission probabilities. Applying this method to the laptop review dataset yields superior performance to the standalone models and a more general discriminative sequence model. The combination of MaxEnt and HMM has the additional advantage of assigning topics at word-level, thus allowing for topic changes within sentence boundaries, although the classifier was trained on sentence-level only. For simplicity, we refer to this combined method as ME+HMM in the following.

2 Related Work

The idea of having a discriminative estimator in a sequence model is not new. McCallum et al. [9] proposed the Maximum Entropy Markov Model (MEMM) and eventually the more general Conditional Random Field (CRF) [5]. HMM and the linear-chain CRF form a so called discriminative-generative pair, as do Naive Bayes and logistic regression (MaxEnt) [22]. While, in principle, each classifier of a discriminative-generative pair can be used to solve the same problem, their training procedures differ concerning the optimality criteria. The generative model estimates probabilities based on the feature frequency in the training data. The discriminative model directly optimizes the conditional probabilities. For sufficiently large datasets, Ng et al. [11] provide evidence that the discriminative model produces a lower asymptotic error in classification tasks. The superiority of MaxEnt over Naive Bayes for text classification tasks is already well established [4] and CRFs have been shown to outperform HMMs in tasks

such as chunking [20], table extraction [16] or information extraction [14]. Both shared tasks of the 2015 [1] and 2016 [21] workshop on noisy user-generated text were focused on Named Entity Recognition (NER) and featured successful submissions based on CRFs. Why bother returning to HMMs?

The differences between HMM, MEMM, CRF and our ME+HMM are subtle. Given the review document as a sequence of observed words $W = (w_1, \dots, w_n)$ and a sequence of hidden states $S = (s_1, \dots, s_n)$, all models aim at finding the optimal sequence S^* by maximizing one of the following probabilities:

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

$$\begin{aligned} \text{MEMM:} \quad P(S | W) &= \prod_{t=1}^n P(s_t | s_{t-1}, w_t) = \\ &= \prod_{t=1}^n \frac{1}{Z_{s_{t-1}, w_t}} \exp \left(\sum_i \lambda_i f_i(s_t, s_{t-1}, w_t) \right) \end{aligned} \quad (2)$$

$$\begin{aligned} \text{CRF:} \quad P(S | W) &= \\ &= \frac{1}{Z_W} \prod_{t=1}^n \exp \left(\sum_i \lambda_i f_i(s_t, s_{t-1}, W) \right) \end{aligned} \quad (3)$$

In (2) and (3), λ_i represent learned weights for features f_i that are computed from a combination of words and states. While the HMM in (1) estimates the joint probabilities of hidden states and input words, the MEMM in (2) estimates S conditioned on the input W . Instead of modeling transition and emission probability distributions as in (1), a MEMM models the probability of the current state s_t based on the previous state s_{t-1} and the current observation w_t . The normalization is done per state, distributing the probability “mass” at each state among the succeeding states. This causes the label bias problem, a bias towards states with fewer successors [5]. Linear-chain CRFs as in (3), in contrast, model the joint probability of the entire state sequence given the observed sequence. The normalization term is then a sum over all possible state sequences [22].

The proposed ME+HMM is still a generative model, with the frequency based estimation of emissions replaced by a conditional estimate provided by the MaxEnt classifier. Thus, the task-related superiority of the discriminative *MaxEnt* can be transferred to the HMM¹. Even if the MaxEnt classifier is trained on sentence-level, the ME+HMM can assign labels on word-level which allows for a higher granularity.

The CRF is the most general of the presented approaches and is usually applied for tasks where many features are needed. Manual feature engineering is

¹ This is why we call it a “semi-discriminative approach”.

required, leading to highly complex models and requiring large datasets. Inducing the most meaningful features is then an additional computational effort with CRFs [8]. The comparison with CRFs using a standard set of features similar to ME+HMM thus stands to reason. Word emission and state transition probabilities in a CRF are optimized simultaneously, but separately in our ME+HMM model. Alternative approaches such as windowed neural networks, recurrent neural networks or attention models have not been considered due to the limited size of the training data in terms of numbers of full review documents. Besides, neural network models such as seq2seq have only been applied on much shorter sequences of text (e.g. 20 newsgroups articles) and assign one topic per review [2].

3 Data

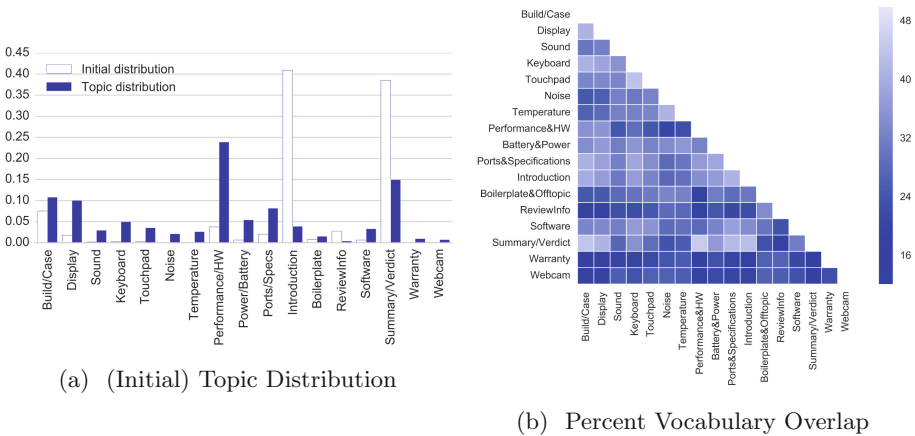


Fig. 1. Dataset analysis. (a) The relative distribution on sentence-level of the seventeen review topics and the topics' likeliness to be the first in a review. (b) The vocabulary overlap between all topics measured in per cent. The diagonal (topic-topic) comparison is 100%. The average PVO is 33.22%.

The performance of the ME+HMM model is assessed on a dataset of expert reviews on laptops collected from several product testing websites². The full dataset contains 3076 reviews manually annotated on sentence-level with one out of 17 predefined target topics. Not all topics are laptop related, some refer to specific review sections. Figure 1a lists the 17 topics and provides a detailed overview of the topics' (initial) distribution. The smallest topic set is *review info* (review metadata) with 887 sentences and the largest is *performance/hw* with 57819 sentences, which accounts for almost 25% of available data. Topics vital for a laptop rating are discussed regularly throughout the reviews (e.g. *performance/hw*), while topic *webcam* for instance only occurs if a laptop possesses this

² The dataset is available at <https://github.com/factai/corpus-laptop-topic>.

component. Nonetheless, these minor topics are interesting aspects to analyze. Note that some topics never start a review at all, while about 80% of reviews either start with an *introduction* or a *summary*. This suggests that the label *summary* might be ambiguous and not easily separable from the *introduction*.

The laptop dataset is different from well-known benchmarking datasets. Each review is available as one file to exploit the sequence information of the topics. Moreover, the expert reviews are much longer and more detailed than user reviews: The average review length in the laptop dataset is 78 sentences. Concerning the granularity, classical benchmarking datasets for topic classification provide only one topic per document. Topics for each sentence are provided in datasets designed for aspect-based sentiment analysis. However, Pontiki et al. [18], for instance, do not provide full documents, meaning that the sentence sequence is not reproducible³. While full documents are available in [17], the reviews in this dataset consist of typically up to fifteen sentences only and do not exhibit a latent topic structure.

3.1 Topic Separability

As classification accuracy correlates with class separability, percent vocabulary overlap (PVO) is used to measure the amount of vocabulary terms shared by two topics [10]. T_i denotes the set of terms occurring in topic S_i :

$$PVO(S_1, S_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} \cdot 100 \quad (4)$$

Figure 1b suggests that the topics *webcam*, *warranty* and *review info* use a more distinct vocabulary, whereas the non-laptop topics (e.g. *summary*) are not as well separable. Given the considerable overlap between topics, the frequency-based estimation of emission probabilities in a standard HMM is not a good choice.

The ability of the MaxEnt classifier to optimize the discrimination between classes can be exploited for the HMM. Table 1 gives an overview of the ten highest weighted words in four exemplary topics as learned from a MaxEnt classifier. Even for the seemingly similar topics *sound* and *noise*, these high scoring words do not overlap. The same is true for a linear-chain CRF. The discriminative models MaxEnt and CRF learn to select significant features to separate the topics. Some variance between the two models is only observed with the topic *summary*, as the CRF also ranks two specific product names (“ideapad”, “aspire”) high. This analysis suggests that the MaxEnt weights could improve the performance of a standard HMM.

³ The sentence IDs provided in [18] are neither consecutive nor contiguous.

Table 1. Ten highest scoring terms in four exemplary topics when based on MaxEnt or CRF weights. (*) BOS denotes the beginning of the sequence in the CRF.

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
CRF			
Sound	Noise	Temperature	Summary
speakers	fan	degrees	BOS(*)
sound	noise	cool	\$
audio	db	temperatures	price
bass	quiet	°C	comparison
volume	silent	heat	verdict
music	fans	warm	db
stereo	load	temperature	life
headphones	audible	cooling	performance
speaker	idle	Hot	ideapad
loud	loud	lap	aspire

4 Methods

At first, the MaxEnt classifier is trained on the labelled sentences. Let $C = \{1, \dots, c\}$ be the set of topics and $D = \{1, \dots, d\}$ the dictionary. As topic labels are available for each sentence $\mathbf{W} = (w_1, \dots, w_n)$, the input to the *MaxEnt* classifier are bag-of-words (BoW) vectors $\mathbf{V} = (v_1, \dots, v_d)$ based on absolute word counts in the sentence: $v_i = \sum_{t=1}^n \mathbb{1}(w_t = i)$. The *MaxEnt* classifier assigns topic $j \in C$ given the input sentence with the probability

$$P(S_1 = j, \dots, S_n = j \mid V_1 = v_1, \dots, V_d = v_d) = \frac{1}{Z_1} \exp \left(\sum_{i=1}^d \lambda_{ij} v_i + n \mu_j \right) \quad (5)$$

where $Z_1 = Z_1(v_1, \dots, v_d)$ is a normalization constant. In (5), the bias term μ_j has been scaled for the length of the input n to account for the simple counting

strategy. Note that in most standard settings, MaxEnt is applied to tf-idf values such that the adjustment related to sequence length is not necessary.

Since the BoW vector are very sparse, it is more efficient to iterate over the words in the sentence instead of the dictionary:

$$P(S_1 = j, \dots, S_n = j \mid W_1 = w_1, \dots, W_n = w_n) = \frac{1}{Z_1} \prod_{t=1}^n \exp(\lambda_{w_t j} + \mu_j) \quad (6)$$

Next, the HMM is initialized. $M = (A, B, C, D, \pi)$ defines an HMM over the set of hidden states C and the set of observations D . An HMM starts in some state s_1 with the probability π_{s_1} and emits an observation w_1 following the emission probability distribution of state s_1 . Then, the model transitions to a new state and again emits an observation. By this, the random sequence of topics \mathbf{S} generates the sequence of observations \mathbf{W} , the words in the review document.

The probability of a transition from state s_{t-1} to s_t is given by the transition probability matrix $A \in \mathbb{R}^{c \times c}$. The emission probability matrix $B \in \mathbb{R}^{c \times d}$ denotes the probability of observing w_t in topic s_t . $\pi \in \mathbb{R}^c$ determines the initial distribution of states:

$$a_{ij} = P(S_t = j \mid S_{t-1} = i) \quad (7)$$

$$b_{jk} = P(W_t = k \mid S_t = j) \quad (8)$$

$$\pi_i = P(S_1 = i) \quad (9)$$

4.1 Emission Probabilities

A straightforward estimate of emission probabilities is counting the word occurrences within each topic or using their tf-idf values. We propose to rely on the discriminative power of the MaxEnt classifier instead and transform the conditional probability distribution of the previously trained classifier into emission probabilities.

Using a stationary HMM for generating the words, we have

$$P(\bar{W} = \mathbf{W}, \bar{S} = \mathbf{S}) = \prod_{t=1}^n \underbrace{P(W_t = w_t \mid S_t = s_t)}_{b_{w_t s_t}} \cdot \underbrace{P(S_t = s_t \mid S_{t-1} = s_{t-1})}_{a_{s_{t-1} s_t}} \quad (10)$$

The MaxEnt assumes that words are independent (by relying on frequencies v_i and ignoring word order). Assuming this for the HMM, too, gives $a_{s_{t-1} s_t} = a_{s_t} = P(S_t = s_t)$ which is independent of step $t - 1$ and (because of the stationarity) independent of t , too. Thus, (10) becomes $\prod_{t=1}^n b_{w_t s_t} \cdot a_{s_t}$. Dividing by the probability $P(\bar{W} = \mathbf{W}) = Z_2$ of the given sequence w_1, \dots, w_n yields

$$P(\bar{S} = \mathbf{S} \mid \bar{W} = \mathbf{W}) = \frac{1}{Z_2} \prod_{t=1}^n b_{w_t s_t} \cdot a_{s_t} \quad (11)$$

We want equivalence for the HMM and MaxEnt models for $s_1 = s_2 = \dots = s_n$. Now, let $s_t = j \ \forall t \in \{1, \dots, n\}$ so that equaling (6) and (11)

$$\frac{1}{Z_1} \prod_{t=1}^n \exp(\lambda_{w_t j} + \mu_j) = \frac{a_j^n}{Z_2} \prod_{t=1}^n b_{w_t j} \quad (12)$$

is, for instance, solved by applying Bayes' theorem to the emission frequencies $b_{i,j}$

$$b_{jk} = \exp(\lambda_{kj} + \mu_j) \cdot \frac{Z_2}{Z_1 p_j} = \exp(\lambda_{kj} + \mu_j) \cdot \frac{P(\bar{W} = \mathbf{W})}{Z_1 a_j} \quad (13)$$

In practice, the HMM emissions are thus computed by

1. training a MaxEnt on the labeled sequences assuming independence between words yielding the λ_{kj}
2. estimating the overall word frequency in the training corpus \hat{p}_w by counting
3. translate MaxEnt weights into emission probabilities by substituting \hat{p}_w for Z_2 in (13) and normalizing with respect to $\sum_{i=1}^d b_{jk} = 1$ instead of dividing by $Z_1 p_j$

4.2 Transition Probabilities

The initial distribution π_i and the transition probabilities a_{ij} are estimated from the training data using the smoothed relative frequencies of word-wise topic changes (additive smoothing). A pseudo-count $\alpha > 0$ serves as regularization term to prevent zero probabilities for unseen transitions in the training data [6]:

$$\hat{a}_{i\cdot} = \frac{a_{i\cdot} + \alpha}{\sum_{j=1}^c (a_{ij} + \alpha)} \quad \text{for } i = 1, \dots, c \quad (14)$$

Tuning the smoothing parameter α also allows for more or less conservative topic changes, even within sentence limits.

4.3 Decoding

The model M is used to decode the sequence W by assigning the most likely sequence of hidden states w.r.t. the joint distribution. Two different dynamic programming algorithms are used to solve this decoding problem [15]: On the one hand, the Viterbi algorithm computes the globally optimal solution $S^* = \arg \max_S P(W, S)$. The posterior decoding algorithm, on the other hand, generates locally optimal solutions $S^* = \{s_i \mid s_i = \arg \max_k \sum_S P(s_i = k \mid W)\}$. The performance of the algorithms is evaluated in the following experiments.

5 Experiments and Results

The laptop review dataset is used for several experiments: At first, the performance of the *MaxEnt* classifier is reported as baseline. Next, the differences of the HMM decoding algorithms are investigated on the laptop review dataset by applying a standard, frequency based HMM. The results are compared to those of the combined algorithm ME+HMM and a linear-chain CRF with comparable features.

The MaxEnt is trained using count-based BoW features for each sentence. The HMM and the CRF decode on word-level. For better comparison, the results of HMM, CRF and ME+HMM are reported on sentence-level by assigning to each sentence its most frequent topic. All classifiers are implemented in *Python3*. Table 2 reports weighted accuracy, precision, recall and F1 scores as defined in the documentation of the Python package scikit-learn [13] for all classifiers. Most algorithms are also taken from the scikit-learn package. If not stated otherwise, default parameter settings are applied. Except for lowercasing, the data is not preprocessed for the experiments. Especially, stopwords are not removed, as it would corrupt the text sequence for the HMM.

5.1 MaxEnt as Baseline

The implementation of the MaxEnt classifier is the `SGDClassifier` from scikit-learn with `loss='log'` and tf-idf vectors as input. `alpha` is set to 0.00001, the `class_weight` is `auto` and the number of iterations is 1000. The MaxEnt achieves an overall accuracy of 70%. A closer look at the individual topic results (see sparkline in Table 2) reveals that some topics are harder to classify while others reach accuracy scores of more than 80%. Low performance is mostly related to either poor vocabulary separability (e.g. *introduction* and *summary*) or little evidence in the dataset (e.g. *review info*). Topics related to laptop specific content yield the best performance.

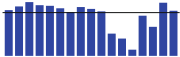



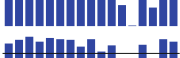

5.2 Standard HMM

For comparison, a standard HMM using word counts per topic as emission probabilities has been implemented⁴. Transition probabilities and initial distribution are estimated as well. The smoothing parameter $\alpha = 0.0001$. Implementation details for the Viterbi and the posterior decoding algorithm such as log space and scaling as provided in [7] are considered. The experiment serves to determine the performance difference of the algorithms. The overall accuracy for the Viterbi algorithm reaches 60.16%, thus outperforming the 53.6% of the posterior algorithm (see Table 2).

As expected, the standard HMM cannot compete with the performance of the MaxEnt classifier, irrespective of the decoding algorithm. The additional information about topic transitions is not enough to compensate for the less competitive emission probabilities. A performance loss is observed for both algorithms.

⁴ The HMM algorithm is no longer supported in the sklearn library.

Table 2. 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. The sparklines indicate the accuracy results for each topic in the same order as in Fig. 1. Each horizontal line denotes the baseline MaxEnt accuracy of 70%.

Algorithm	Accuracy per Class	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%

The only exception is topic *build/case* which is due to the HMM assigning the first topic as default when topics have equal probability. The precision scores do not differ considerably from the MaxEnt.

5.3 MaxEnt Emissions for HMM (ME+HMM)

For combining the MaxEnt probability distributions with a HMM, the weights λ_{w_tj}, μ_j are extracted from the trained MaxEnt classifier to calculate the conditional probabilities per word and topic following (13). During training, the normalized word frequencies \hat{p}_w are stored. Transition probabilities and initial distribution remain the same as for the standard HMM in Sect. 5.2.

These more distinctive emission probabilities raise the performance of the classifier (see Table 2). ME+HMM performs not only better than the standard HMM, but also outperforms the MaxEnt classifier by approximately 7% on average. Except for *ports/specifications*, all laptop-related topics achieve over 80% accuracy. A performance drop when compared to the MaxEnt is only noticeable for the topics *review info* and *summary/verdict*. The low performance of *review info* is due to the topic being under-represented in the training data. The performance of the topic *summary* might be caused by issues in the training data, as will be discussed in Sect. 6.

5.4 Comparison of ME+HMM and CRF

For compatibility with sklearn, the `sklearn-crfsuite`⁵ is chosen as implementation for the linear-chain CRF. The training algorithm is set to `lbfgs` (gradient descent) and the L2 regularization coefficient to 100. Although way below the capabilities of a general CRF, only the current and previous word identities and the beginning of a sequence (BOS) are used as features to allow for a fair comparison. Being a standard method for tasks like NER with a restricted set of states, the CRF cannot handle topics that overlap as much as in the given laptop dataset. With estimating both emission and transition probabilities simultaneously, the CRF has too many degrees of freedom to capture the less frequent topics, thus the overall accuracy reaches only 40%. The combined model ME+HMM has its strengths with longer, subsequent sequences of topics. For the transition probabilities, a discriminative estimation is not necessary.

6 Discussion

A CRF with basic features was implemented for a fair comparison to the other models. In this setting, the CRF does not perform well. With a larger set of hand-crafted features, the CRF will eventually perform comparably to the proposed model. The current trend towards deep learning models is supposed to mitigate the feature engineering requirement. Those models should implicitly learn input representations, but require a careful architecture design and an abundance of training data, especially for modeling long input sequences. For the given problem setting, ME+HMM fills the gap by performing with standard features: no manual effort is required and the size of the given dataset is sufficient.

Concerning the decoding algorithm, the results suggest that for ME+HMM the posterior algorithm is slightly superior to the Viterbi algorithm, as opposed to the standard HMM. Schwartz [19] noted that the Viterbi algorithm most likely does not find the optimal path in case its probability is low and many other paths have almost equal probability. In this case, the posterior decoding may outperform the Viterbi algorithm.

The experiments have shown that the model ME+HMM is superior to other classifiers in assigning topics on sentence-level. Although the dataset is designed as a sentence classification task, the document structure of the expert reviews can be exploited. This allows to assign more than one label per sentence which is convenient for contrasting or comparison sentences (e.g. “on the one hand, on the other hand”), for concessive clauses (e.g. “although”, “despite”) or enumerations. Table 3 is an illustrative example taken from a review, where the ME+HMM classification (bottom) differs from the gold annotation (top). In the gold annotation, only the topic *keyboard* is assigned to the second sentence, although also the *touchpad* is discussed, as accurately captured by ME+HMM. Although the advantage of intra-sentence topic changes cannot be captured directly due to

⁵ <https://sklearn-crfsuite.readthedocs.io/en/latest/index.html>.

the lack of granularity in the dataset, the example suggests that word-level topic assignments could be promising and reveal additional insight on the product, as in the example case.

Table 3. A sample sequence taken randomly from a review. The gold labeling suggests three different topics (top), the ME+HMM model assigns four topics (bottom).

(Gold labels)			
Otherwise, the approx. 3.3 kg 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.). <i>The main point of complaint is the enormous noise development, typical for a gamer: the fan is clearly audible during load</i>			
(ME+HMM)			
Otherwise, the approx. 3.3 kg 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 <i>of complaint is the enormous noise development, typical for a gamer: the fan is clearly audible during load</i>			
Build/Case	Noise	Keyboard	Touchpad

Another interesting insight from the experiments is the low performance of the topic *summary*. A closer investigation reveals that *summary* is often misclassified as *introduction*. The topic distribution in Fig. 1a illustrates an unbalance between *introduction* and *summary*, although it can be assumed that most of the reviews consist of both an *introduction* and a *summary*. However, the dataset consists of more than three times as many *summary* sentences. It could still be argued that summaries in this dataset are simply longer, i.e. consist of more sentences, but also the distribution of initial topics suggests that some sentences might misleadingly be labeled as *summary*. Thus, the label quality of the sequence models is probably even higher as the numbers suggest.

7 Conclusion and Future Work

Faced with a new dataset for sentence-level topic classification on laptop reviews, we introduce the model ME+HMM, a combination of MaxEnt-based weights and an HMM. The expert laptop reviews are detailed articles with an inherent topic structure. The MaxEnt classifier in general performs well on language classification tasks, but can profit from a sequence model that also captures the transitions between topics within one review. On the given dataset, the new model ME+HMM improves the performance of the standalone MaxEnt classifier and also outperforms more general models such as a linear-chain CRF with comparable features. Although the ME+HMM is trained on sentence-level, labels are assigned at word-level, which allows for detecting intra-sentence topic changes. The ME+HMM relies on well established concepts and incorporates

preliminary knowledge: A frequency-based estimation of transitions is reasonable for the infrequent topic changes on word-level. Concerning the emissions, the conditional estimation performs best. The combination of MaxEnt and HMM eliminates the excessive degrees of freedom of a generalized model leading to an approach with less complexity for comparable tasks.

The results from the topic classification task can be included in tasks such as aspect-based sentiment analysis. For automated text summarization or generation, it would be interesting to see the ME+HMM model generate topic sequences as outlines.

Acknowledgments. We would like to thank our colleagues at fact.ai for the inspiring discussions and the collection and provision of their dataset [3].

References

1. Baldwin, T., de Marneffe, M.C., Han, B., Kim, Y.B., Ritter, A., Xu, W.: Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In: *Proceedings of the Workshop on Noisy User-generated Text*, pp. 126–135 (2015)
2. Dai, A.M., Le, Q.V.: Semi-supervised sequence learning. In: *Advances in Neural Information Processing Systems*, pp. 3079–3087 (2015)
3. fact.ai: Aggregated Text Corpus of Laptop Expert Reviews with Annotated Topics (2018). <https://github.com/factai/corpus-laptop-topic>
4. 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, EMNLP 2002*, pp. 9–16. ACL (2002)
5. 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, ICML 2001*, pp. 282–289 (2001)
6. Manning, C.D., Raghavan, P., Schütze, H., et al.: *Introduction to Information Retrieval*, vol. 1. Cambridge University Press, Cambridge (2008)
7. Manning, C.D., Schütze, H.: *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge (1999)
8. McCallum, A.: Efficiently inducing features of conditional random fields. In: *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pp. 403–410. Morgan Kaufmann Publishers Inc. (2002)
9. McCallum, A., Freitag, D., Pereira, F.C.N.: Maximum entropy Markov models for information extraction and segmentation. In: *Proceedings of the Seventeenth International Conference on Machine Learning, ICML 2000*, pp. 591–598 (2000)
10. Medlock, B.W.: *Investigating Classification for Natural Language Processing Tasks*. University of Cambridge, Computer Laboratory, Technical report (2008)
11. Ng, A.Y., Jordan, M.I.: On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes. In: *Advances in Neural Information Processing Systems*, pp. 841–848 (2002)
12. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs Up? Sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing-Volume 10*, pp. 79–86. Association for Computational Linguistics (2002)

13. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
14. Peng, F., McCallum, A.: Information extraction from research papers using conditional random fields. *Inf. Process. Manage.* **42**(4), 963–979 (2006)
15. Petrushin, V.A.: Hidden Markov models: fundamentals and applications. In: *Online Symposium for Electronics Engineer* (2000)
16. Pinto, D., McCallum, A., Wei, X., Croft, W.B.: Table extraction using conditional random fields. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 235–242. ACM (2003)
17. Pontiki, M., et al.: SemEval-2016 task 5: aspect based sentiment analysis. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 19–30 (2016)
18. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: SemEval-2014 task 4: aspect based sentiment analysis. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*, pp. 27–35 (2014)
19. Schwartz, A.S.: Posterior decoding methods for optimization and accuracy control of multiple alignments. Ph.D. thesis, EECS Department, University of California, Berkeley (2007)
20. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp. 134–141. ACL (2003)
21. Strauss, B., Toma, B., Ritter, A., de Marneffe, M.C., Xu, W.: Results of the WNUT16 named entity recognition shared task. In: *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pp. 138–144 (2016)
22. Sutton, C., McCallum, A., et al.: An introduction to conditional random fields. *Found. Trends® Mach. Learn.* **4**(4), 267–373 (2012)