Cornelia Ferner / Martin Schnöll / Arnold Keller / Werner Pomwenger / Stefan Wegenkittl

Topic-Klassifizierung für automatisierte Produktbewertungen mittels Hidden Markov Modellen

109 - Data Science: Erfassung, Modellierung, Analyse und Visualisierung von Daten

Abstract

Hidden Markov Modelle (HMMs) werden im Fachgebiet Natural Language Processing vor allem zum Part-of-Speech Tagging auf Wortebene verwendet. In dieser Arbeit werden HMMs zur Topic-Klassifizierung von Sätzen bzw. Paragraphen eingesetzt. Klassische Bag-of-Words Methoden werden dadurch um eine Berücksichtigung des Textaufbaus und der typischen Topicabfolgen erweitert. Die HMM-basierte Topic-Klassifizierung erreicht state-of-the-art Performance und bietet den zusätzlichen Vorteil, dass auch innerhalb von Absätzen Topic-Wechsel erkannt werden können.

Keywords:

Natural Language Processing, Topic Classification, Hidden Markov Model, Maximum Entropy Classification

1. Einleitung

Topic-Klassifizierung bezeichnet Methoden, mit denen Absätzen, Sätzen oder sogar einzelnen Wörtern w in natürlichsprachlichen Texten $T = (w_1, \dots, w_i, \dots), w_i \in W$, *Topics* (Themengebiete) $s(w_i) \in S = \{s_1, \dots, s_n\}$ auf einer endlichen Menge von Topics S zugeordnet werden. Im vorliegenden Datensatz werden dazu Online-Produktbewertungen untersucht. Ca. 60% der KundInnen lesen vor dem Einkauf Online-Reviews (Charlton 2015). Durch die Verfügbarkeit aller relevanten Produktinformationen und -bewertungen direkt im Webshop können E-Commerce-Anbieter verhindern, dass KundInnen während der Recherche die Seite verlassen und unter Umständen bei anderen AnbieterInnen kaufen.

Zu einem bestimmten Produkt (z.B. einem Laptop) werden Online-Expertenbewertungen gesammelt und Produktkomponenten mit Hilfe von semantischen Analysen und Methoden des maschinellen Lernens extrahiert. Ziel ist eine aggregierte Darstellung von Bewertungen zu jedem Produkt: Detaillierte Informationen zu dessen Produktkomponenten, den sogenannten Topics oder *Aspects*

(z.B. Keyboard oder Display), erhöhen die Vergleichbarkeit und dienen der inhaltlichen Transparenz von Bewertungen.

Zur Ermittlung dieser Produktkomponenten innerhalb eines Dokumentes wird die Menge S der Topics entweder vorgegeben (Zhang / Oles 2001) oder mittels Clustering (Wartena / Brussee 2008) aus dem Text extrahiert. Topics können innerhalb eines Dokumentes explizit *(„The keyboard is comfortable to use.")* oder implizit *(„It has enough volume to fill a room.")* genannt werden. Explizite Nennungen der Topics werden mittels regelbasierten Methoden (Hu / Liu 2004, Blair-Goldensohn et al. 2008) erkannt. Dabei wird mittels *Part-of-Speech* (POS) *Tagging* die Zuordnung der einzelnen Wörter zu Wortarten ermittelt und nach bestimmten Wortgruppen (z.B. nur Hauptwörter) gefiltert.

Die Merkmalsextraktion wird häufig nach dem *Bag-of-Words* (BoW) Prinzip vorgenommen. Dabei spielt die Häufigkeitsverteilung der auftretenden Wörter in jedem Topic eine Rolle, nicht jedoch die Abfolge der Topics im Gesamttext.

Zu den Standardmodellen für die implizite Topic-Detektion zählen probabilistische Klassifikatoren wie *Naive Bayes* (McCallum / Nigam 1998, Wang / Manning 2012), *Maximum Entropy* (MaxEnt) (Zhang / Oles 2001) und *Support Vector Machines* (SVM) (Wang / Manning 2012). Für den vorliegenden Datensatz ergibt sich mit diesen Methoden eine stabile Performance bei der Klassifizierung ganzer Absätze, eine Klassifizierung auf Wortebene ist jedoch nicht möglich.

2. Methode

Diese Arbeit verwendet eine Kombination aus *Hidden Markov Models* (HMMs) (Rabiner / Juang 1986) und einem MaxEnt-Klassifikator in einem mehrstufigen Prozess zur impliziten Topic-Detektion. HMMs modellieren Systeme mit stochastischen Zustandswechseln auf einer Menge $S = \{s_1, \dots, s_n\}$ von Zuständen mit Übergangswahrscheinlichkeiten $P[s_k \to s_l] = a_{kl}$ mit $1 \leq k, l \leq n$, wobei $A = (a_{kl})$ Übergangsmatrix genannt wird. In der vorliegenden Anwendung wird die Abfolge der Topics im Text, $s(w_i) \in S$, als Markov-Sequenz aufgefasst. Diese Zustände sind jedoch nicht direkt beobachtbar (*Hidden States*), sondern können nur indirekt über ihre Emissionsverteilungen $P[v_j|s_k] = b_{jk}$ auf dem Merkmalsraum $V = \{v_1, \dots, v_m\}$ rückgeschlossen werden. $B = (b_{jk})$ bezeichnet die sogenannte Emissionsmatrix, die für jedes Topic s_k die Wahrscheinlichkeit einer Emission des Wortes v_j angibt. Im Fachgebiet Natural Language Processing kommen HMMs vor allem bei POS Tagging (Church 1988) und *Named Entity Recognition* (NER) (Bikel et al. 1997) zum Einsatz. Bei beiden Methoden wird die Bedeutung eines Wortes im Satz modelliert.

Zur Bestimmung der Topic-Abfolge auf Satzebene werden die Reviews als Sequenz von Wörtern interpretiert und folgendes HMM $\lambda = (S; V; A; B; \pi)$ definiert:

S ist die Menge aller vorgegebenen Topics. V, das Alphabet der möglichen Beobachtungen, ergibt sich aus allen in den Expertenberichten vorkommenden Wörtern v_j. Für jedes Topic wird auf diesem Wortraum eine Wahrscheinlichkeitsverteilung nach folgender Methode ermittelt:

Anstatt die Worthäufigkeiten pro Topic zu verwenden, wird ein MaxEnt-Klassifikator auf den gesamten Wortraum trainiert. Dieser liefert pro Wort v_j eine Gewichtung w'_{jk} für jedes Topic s_k. Mit Hilfe eines Softmax-Verfahrens werden diese Werte in eine Verteilung umgewandelt und mit einem Faktor σ_e skaliert:

$$w_{ji} = \frac{1}{1 + e^{-w'_{jk} \cdot \sigma_e}}$$

Durch Normieren der Spalten der Matrix $W = (w_{jk})$ entsteht die Emissionsmatrix B von λ, welche die Wahrscheinlichkeit angibt, im Zustand s_k die Beobachtung v_j zu machen.

Die Übergangswahrscheinlichkeiten für die Übergangsmatrix A wird aus den Trainingsdaten geschätzt. Damit auch Übergänge zugelassen werden, die während des Trainings nicht beobachtet wurden, wird ein Pseudocount σ_t addiert. Auch die Startverteilung der Topics π wird aus den Trainingsdaten geschätzt.

Im Decoding-Schritt wird die Klassifizierung der vorliegenden Wort-Sequenzen zu den Topics mittels zweier Algorithmen ermittelt: Zur weiteren Optimierung wird die Performance des Viterbi- jener des Forward/Backward (FW/BW)-Algorithmus gegenübergestellt. Während der Viterbi-Algorithmus die Topic-Zuweisung global optimiert, errechnet der FW/BW-Algorithmus die lokal optimalen Zuweisungen (Jurafski / Martin 2008).

HMMs zur Topic-Detektion wurden schon von (McCallum et al. 2000) und (Jin / Ho 2009) untersucht. Im Vergleich zu deren Arbeiten werden in diesem Projekt keine kombinierten Features als Hidden States verwendet, sondern der Schwerpunkt auf die Optimierung der Wahrscheinlichkeitsverteilungen gelegt. Primäres Ziel ist es, die durch einen Standard-MaxEnt-Klassifikator erreichbare Performance durch die Einbeziehung der Übergangsmatrix der Topics signifikant zu verbessern.

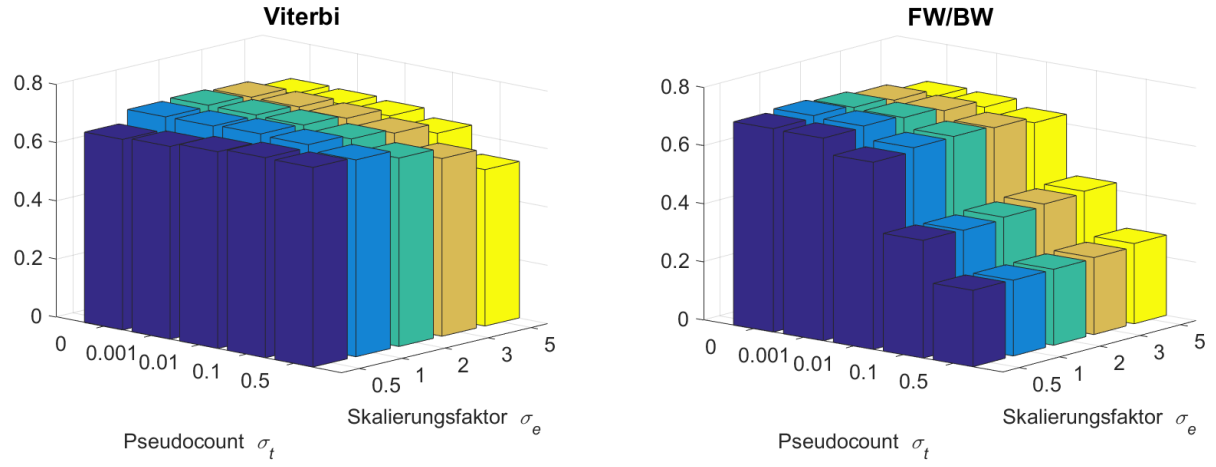

Abbildung 1: Ermittlung der optimalen Skalierungsfaktoren und Pseudocounts

3. Resultate

Zur Klassifizierung wurden vom Projektpartner[1] manuell annotierte Trainingsdatensätze zur Verfügung gestellt. Der Corpus umfasst 500 Expertenreviews zu Laptops unterschiedlicher Hersteller und enthält Zuordnungen zu 17 Topics (u.a. Display, Keyboard, Sound, Performance, Warranty, Software, Temperature, Noise, Build/Case, Ports/Specifications), sowohl auf Satz- als auch Paragraphenebene.

Ein Standard-MaxEnt-Klassifikator erreicht für dieses Testset eine Übereinstimmung von 63.68% auf Satzebene. Die Klassifizierung mittels HMM erfolgt für dieses Testset mit $n = 17$ Topics und $m = 35000$ Wörtern im Dictionary. Abbildung 1 zeigt die Ermittlung des optimalen Skalierungsfaktors σ_e und des Pseudocounts σ_t noch auf Wortebene, sowohl für den Viterbi- als auch den FW/BW-Algorithmus.

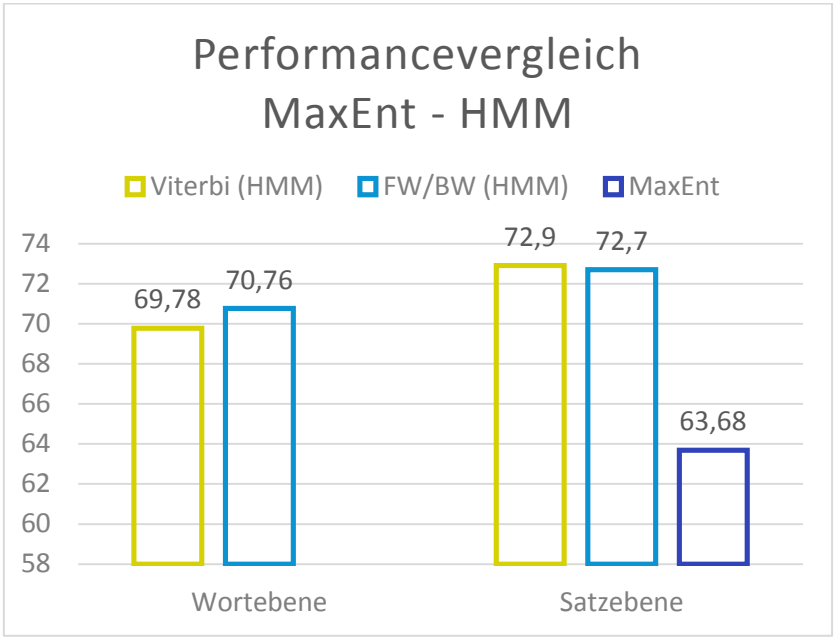

Abbildung 2: Performancevergleich von MaxEnt-Algorithmus und HMM

Mit optimierten Parametern wird mittels HMM eine Performance von 69.78% (Viterbi) und 70.76% (FW/BW) schon auf Wortebene erreicht. Wird jedem Satz das Topic zugewiesen, das den meisten Wörtern in diesem Satz zugeteilt wurde, steigt die Übereinstimmungsrate auf Satzebene auf 72.90% bzw. 72.70%, wie in Abbildung 2 ersichtlich ist.

4. Diskussion

Abbildung 3 zeigt eine Topic-Zuweisung mit dem Viterbi-Algorithmus im Vergleich zur manuellen Annotation für einen Ausschnitt eines Laptop-Reviews. Dabei ist ersichtlich, dass die HMM-Annotation an zwei Stellen vom Trainingsdatensatz abweicht: Einmal wird fälschlicherweise das Topic Keyboard anstatt Warranty zugewiesen. Der Wechsel von Keyboard auf Build/Case hingegen erscheint inhaltlich nachvollziehbar, ist in den Trainingsdaten momentan aber nicht abgebildet. Mit herkömmlichen

[1] Fact AI KG – www.fact.ai

Methoden (SVM, MaxEnt) kann auf Wortebene wegen der zu hohen Topic-Fluktuation keine zufriedenstellende Performance erreicht werden. Bei HMMs kann beim Schätzen der Übergangswahrscheinlichkeiten (bzw. auch bei der Optimierung des Pseudocounts) ein Trade-off zwischen zu häufigem und zu konservativem Topic-Wechsel erfolgen.

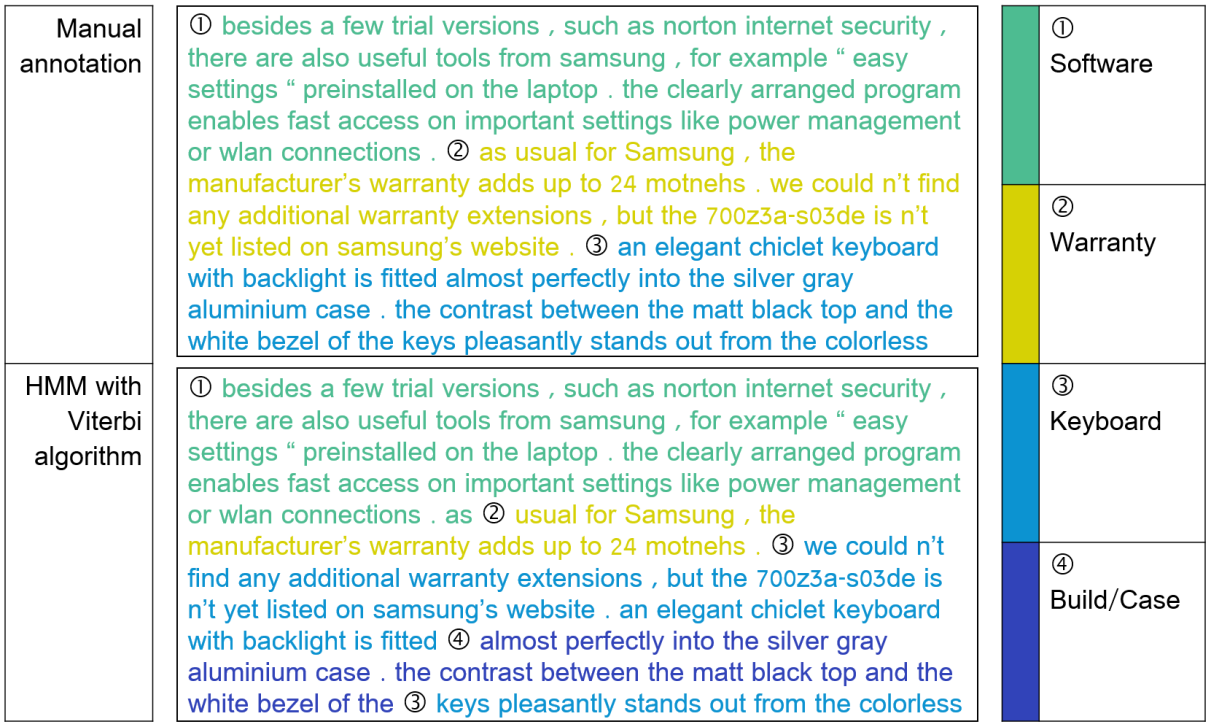

Abbildung 3: Vergleich der Topic-Annotationen: Manuelle Annotation (oben) und HMM (unten).

Durch die vorgestellte Kombination von HMMs und MaxEnt zur Topic-Klassifizierung wird eine höhere Performance erzielt als mit dem Standard-MaxEnt-Klassifikator. Da HMMs Dokumente als Wort-Sequenzen modellieren, besteht zusätzlich der Vorteil, dass eine Annotation auch unterhalb der Satz-Ebene möglich ist. Aktuell wird an der Verbesserung der Schätzer für Emissions- und Transitionsmatrix gearbeitet.

Literaturliste/Quellenverzeichnis:

Bikel, Daniel et al. (1997): Nymble: a high-performance learning name-finder. In: Proceedings of the fifth conference on applied natural language processing, Association for Computational Linguistics, 194-201.

Blair-Goldensohn, Sasha et al. (2008): Building a sentiment summarizer for local service reviews. In: WWW Workshop on NLP in the Information Explosion Era.

Charlton, Graham (2015): Ecommerce consumer reviews: why you need them and how to use them. https://econsultancy.com/blog/9366-ecommerce-consumer-reviews-why-you-need-them-and-how-to-use-them/ (23.11.2015)

Church, Kenneth (1988): A stochastic parts program and noun phrase parser for unrestricted text. In: Proceedings of the second conference on applied natural language processing, Association for Computational Linguistics, 136-143.

Hu, Minqing/Liu, Bing (2004): Mining Opinion Features in Customer Reviews. In: 19th National Conference of Artificial Intelligence, 755-760.

Jin, Wei/Ho, Hung Hay (2009): A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining. In: Proceedings of the International Conference on Machine Learning, 465-472.

Jurafsky, Dan/Martin, James H. (2008): Speech and Language Processing. Upper Saddle River, NJ: Prentice Hall.

McCallum, Andrew/Nigam, Kamal (1998): A comparison of event models for Naïve Bayes text classification. In: AAAI-98 workshop on learning for text categorization 752, 45-48.

McCallum, Andrew et al. (2000): Maximum Entropy Markov Models for Information Extraction and Segmentation. In: ICML 17, 591-598.

Rabiner, Lawrence/Juang, Biing-Hwang (1986): An introduction to hidden Markov models. In: ASSP Magazine, IEEE 3 (1), 4-16.

Wang, Sida/Manning, Christopher (2012): Baselines and bigrams: Simple, good sentiment and topic classification. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers 2, 90-94.

Wartena, Christian/Brussee, Rogier (2008): Topic Detection by Clustering Keywords. In: Database and Expert Systems Application, IEEE International Workshop on, 54-58.

Zhang, Tong/Oles, Frank (2001): Text Categorization Based on Regularized Linear Classification Methods. In: Information Retrieval (4), 5-31.