Predicting Polarities of Entity-Centered Documents without Reading their Contents

Christian Beyer
Otto-von-Guericke Univ. Magdeburg
christian.beyer@ovgu.de

Uli Niemann
Otto-von-Guericke Univ. Magdeburg
uli.niemann@ovgu.de

Vishnu Unnikrishnan
Otto-von-Guericke Univ. Magdeburg
vishnu.unnikrishnan@ovgu.de

Eirini Ntoutsi
Leibniz Univ. Hannover
ntoutsi@kbs.uni-hannover.de

Myra Spiliopoulou
Otto-von-Guericke Univ. Magdeburg
myra@ovgu.de

ABSTRACT

Opinion stream mining algorithms learn and adapt a polarity model as new opinionated texts arrive. Text understanding is computationally expensive though, and sensitive to the emergence of new words. In this work, we study polarity prediction for opinions on given entities and investigate how prediction quality is affected when we ignore the text of past opinions but exploit the entity-opinion link and the past polarity scores on it. We model each entity as a trajectory of polarity scores and propose learning algorithms that exploit these trajectories for polarity prediction. We study the performance of our approach on the Tools & Home Improvement products of the Amazon Reviews Dataset [1].

CCS CONCEPTS

•**Theory of computation** → *Streaming models;* Models of learning;

KEYWORDS

Stream Mining, Document Polarity, Prediction, Trajectory

ACM Reference format:
Christian Beyer, Uli Niemann, Vishnu Unnikrishnan, Eirini Ntoutsi, and Myra Spiliopoulou. 2018. Predicting Polarities of Entity-Centered Documents without Reading their Contents. In *Proceedings of SAC 2018: Symposium on Applied Computing , Pau, France, April 9–13, 2018 (SAC 2018),* 4 pages.
DOI: 10.1145/3167132.3172870

1 INTRODUCTION

Opinion stream mining algorithms exploit the contents of the opinionated documents for learning, and remember past information on polarity and word distribution in them. Each opinion refers to a specific entity, e.g. to a product, so it seems necessary to additionally consider the entity, in order to properly interpret the text referring to it. Since text preprocessing is complex, we question how much of it is needed. In Fig. 1 we show the polarity labels of two products (solid black lines), a predictor of this polarity which solely exploits past observed label values and a predictor which utilizes also the review text: the text-ignorant predictor approximates the arriving labels remarkably well. Motivated by this observation, we propose a framework of entity-centered predictors, and juxtapose the performance of text-ignorant and text-aware ones.

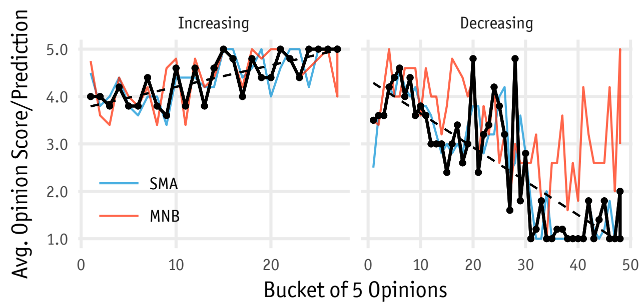

Figure 1: Polarity evolution for two Amazon products as bucket average of scores (solid black line) and predictions by Simple Moving Average (SMA) and Multinomial Naive Bayes (MNB) for a bucket/window of 5 opinions

2 RELATED WORK

We aim to predict the label of the next opinion to a given entity, without exploiting any content. The association between an opinionated document and its "target entity" has been studied intensively in recent years, see e.g. the works of [3, 4, 6]. Our work is orthogonal to this research thread, because we do not investigate target extraction. We rather concentrate on polarity prediction for documents with known target entities, e.g. products in one of the many platforms for product ratings, hotel reviews etc. Moreover, we investigate the prediction problem in a stream setting.

In the stream setting, let $t_1, t_2, \ldots, t_j, \ldots$ be the arrival timepoints of opinionated documents, also denoted as *instances* hereafter. An opinion stream classifier learns a model M_j from all instances seen until and including t_j. At t_{j+1} it reads the arriving instance, predicts its label, has the true label disclosed and then adapts the model to it, deriving model M_{j+1}. Forgetting is a central aspect of stream learning algorithms. An overview of forgetting mechanisms can be found in [1]. In this study, we use the gradually forgetting opinion stream classifier of [5] as a reference strategy to compare with our own methods that do not exploit content.

[1] http://jmcauley.ucsd.edu/data/amazon/

DOI: 10.1145/3167132.3172870

3 ENTITY-CENTERED PREDICTORS

We study a stream of opinions, each one linked to an entity from a set $E = \{e_1, e_2, ..., e_n\}$. We denote an opinion, or *instance*, as a pair $i_{k,j}$, where $e_k \in E$ is the entity and t_j is the timepoint of the opinion's arrival. These opinions constitute the entitiy's *trajectory* $traj(e_k, t_j) = [i_{k,k_1}, \ldots, i_{k,k_l}]$, where t_j is the current timepoint and $t_{k_1}, \ldots, t_{k_l} \leq t_j$ are the timepoints to which opinions on e_k have arrived. Whenever we refer to only one entity and there is no risk of confusion, we denote as t_j the timepoint of the most recent arrival of an opinion on it. Since some entities are more popular than others, their trajectories may vary substantially in length.

3.1 Prediction Framework

Similarly to a conventional opinion stream classifier, we want to predict the polarity label $\hat{y}(i_{k,j}) \in \mathcal{L}$, where $\mathcal{L}$ is the set of labels and the real label is denoted with y. To keep the notation simple, we use the term "label" also for numerical polarity scores.

We propose *entity-centered predictors*, each of which builds a model per entity e_k: when an opinion i_{k,k_j} on e_k arrives at t_{k_j} with $j > 1$, the predictor uses the model $M_{k,k_{j-1}}$ learnt until the previous opinion for this entity (at $t_{k_{j-1}}$) to predict the label, and then uses this new opinion to adapt the model into M_{k,k_j}. Obviously, the model of e_k is only updated when a new opinion arrives for e_k.

We distinguish between *text-ignorant* and *text-aware* predictors. Further, the predictors differ on how much past data they remember. Among the forgetting mechanisms for stream classification [1], we opt for a sliding window of length w instances per entity.

3.2 Entity-centered text-ignorant predictors

Let $i_{k,j}$ be the opinion arriving at t_j and referring to entity e_k. We propose following algorithms that learn one model per entity:

- The predictor *Prior* assigns to $i_{k,j}$ the most frequently observed label $\hat{y}$ in $traj(e_k, t_{j-1})$.
- The window-based predictor *WPrior* considers only the w most recent instances in $traj(e_k, t_{j-1})$ to compute $\hat{y}(i_{k,j})$.
- The predictor *HMM* learns for each entity a Gaussian HMM with three states.
- *WHMM* is the window-based counterpart of *HMM*.
- For a set of non-negative integer labels $\mathcal{L}$, the window-based predictor *Regression* learns a linear regression model over the labels of the trajectory of e_k, predicts the next value as real number, and sets as label $\hat{y}(i_{k,j})$ the closest integer (rounding).
- The window-based predictor *Simple Moving Average (SMA)* sets $\hat{y}(i_{k,j})$ to the average over all labels within the window, again after rounding.

3.3 Entity-centered text-aware predictors

The text-aware predictors consider the contents of the opinions on an entity e_k. For vectorization of the opinions, we use bag-of-words after stopword removal. We propose following algorithms that build one model per entity:

- The one-nearest neighbor *KNN* assigns to $i_{k,j}$ the label of the instance in $traj(e_k, t_{j-1})$ that is most similar to $i_{k,j}$.
- *WKNN* is its window-based counterpart.

4 EVALUATION FRAMEWORK

Our approach differs from conventional opinion stream mining, since we learn one model per entity, and the entity-centered learners see much less information than a global model would exploit. To run prequential evaluation properly in this scenario, we align the stream to the entities and devise κ^+ statistics that are appropriate for an evaluation per entity.

4.1 Aligning the Stream to the Entities

A predictor that exploits the target entity e of an opinion i to assess the label of i can do so after seeing at least one opinion on e. We set $w \geq 1$ as lower threshold to the number of opinions to be used for prediction. We build a *training stream* and an *evaluation stream*, by first removing all entities with less than $2 \cdot w$ opinions, and then placing the first w opinions of each retained entity in the training stream and the remaining ones (w or more) in the evaluation stream, respecting their order of arrival.

4.2 Entity-ignorant baselines

We consider the following entity-ignorant baselines:

- The global, all-past-based, text-ignorant *GPrior* remembers all opinions ever seen; it ignores the texts but remembers the labels. It assigns to $i_{k,j}$ the most frequently observed label, independently of the entity.
- The global, window-based, text-aware one-nearest-neighbor *GKNN* considers the N most recently observed opinions and assigns to $i_{k,j}$ the label of its nearest neighbor among those N. In our experiments, we set $N = 1000$.
- We denote as *MNB* the Multinomial Naive Bayes proposed in of [5]. For this algorithm, we chose as a feature space the 1000 most frequent words over the whole stream. For experiments where the dataset did not fit in memory, we extracted the 1000 words from the first half of the dataset.
- *MNBF* is the "fading" MNB variant proposed in [5].

4.3 Entity-Centered Evaluation

We train the predictors on the training stream, and test them on the evaluation stream with prequential evaluation. The entity-ignorant baselines exploit the whole training stream, while our entity-centered algorithm build one model per entity. The κ^+ statistic proposed in [7] is used but must be entity-centered, i.e. computed for each entity separately. For each pair of predictors, we then count the number of times one of them has better κ^+ values than the other.

For numerical label values, we evaluate on the Root Mean Squared Error per entity e

$$RMSE(S_e, M_e) = \sqrt{\frac{1}{|S_e|} \sum_{x \in S_e} (\hat{y}_x - y_x)^2}$$

where S_e is the part of the evaluation stream referring to entity e, M_e is the entity-centered model, $\hat{y}_x$ is the predicted label for instance $x \in S_e$ and y_x is the true one.

In this study, we simplified the evaluation by considering all entity-centered predictors as a single, *conglomerated* model. This model is a naive type of ensemble, where each opinion receives a single prediction, from the entity-centered predictor responsible for it. We then computed $RMSE(S, M)$ for the whole stream. This might have been unfair to the entity-centered methods.

For a set of nominal labels $\mathcal{L}$, we evaluate on accuracy and on balanced accuracy, expressed as $\frac{\sum_{l \in \mathcal{L}} recall(l)}{|\mathcal{L}|}$. We again simplified the evaluation by computing $recall(S, l) \equiv recall(l)$ instead of $recall(S_e, l)$, and by putting all the entity-centered models together into a conglomerated model.

5 EXPERIMENTAL EVALUATION

We vary the minimum number of opinions per entity $w = 5, 10, 15, 20$. For the window-based methods we consider windows of $W = 5, 10, 15, 20$ instances. Since W cannot be larger than w, we vary w and W simultaneously, i.e. $W = w$.

5.1 Datasets of the Experiments

We use two subsets of the Amazon dataset [2], the 'Tools and Home Improvement' (Tools hereafter), and the set of opinions on watches from the 'Clothing, Shoes and Jewelry' subset, extracted from the 5-core version [2] (Watches hereafter). Tools contains 260,659 entities and 1,926,047 opinions, Watches contains 1,221 entities and 13,027 opinions. In the following, we report on our results for Tools [3].

In Fig. 2, we see the distribution of opinion labels (1 to 5 stars) for Tools: we see that it is very skewed, the most frequent label is "5"; this holds also for Watches. The customers' preference for positive rather than negative opinions has manifested itself over a longer period of time (cf. Fig. 2 (right)).

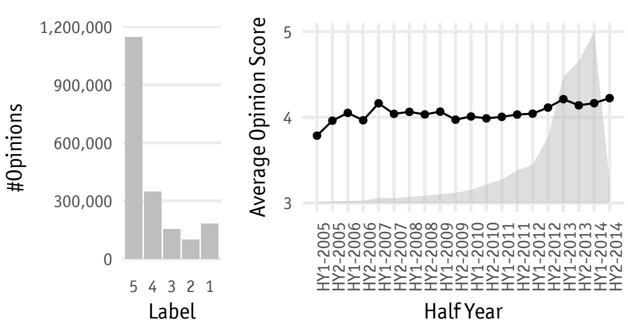

Figure 2: Distribution of opinion labels (left) and average product rating over time (right) for Tools. The height of the histogram depicts rating density within each half year.

		#predictions: 1247724	Acc.	RMSE	Bal. Acc.	κ+ loc. p.	Exec. Time [min:sec]
Entity-Centered	Text-ignorant	Prior	0.592	1.505	0.242	0.000	00:06
		WPrior	0.521	1.567	<u>0.259</u>	0.000	00:08
		SMA	0.385	<u>1.332</u>	0.244	0.000	00:07
		Reg	0.370	1.648	0.229	0.000	05:12
		HMM	0.580	1.530	0.242	0.000	443:09
		WHMM	0.471	1.638	0.247	0.000	1193:26
	Text-aware	KNN	0.512	1.439	0.290	0.000	67:40
		WKNN	0.482	1.544	0.269	0.000	36:52
Global	Text-ign.	GPrior	<u>0.598</u>	1.552	0.200	<u>0.001</u>	00:05
	Text-aware	MNB	0.652	1.184	0.409	0.224	294:06
		MNBF	**0.653**	**1.180**	**0.409**	**0.227**	12:48
		GKNN	0.484	1.580	0.278	0.000	244:20

#e: 33989	Prior	WPrior	SMA	Reg	HMM	WHMM	KNN	WKNN	GPrior	MNB	MNBF	GKNN
Prior		.22	.37	.44	.20	.40	.35	.37	.01	.18	.18	.42
WPrior	.01		.26	.33	.09	.29	.24	.26	.01	.13	.13	.32
SMA	.01	.01		.12	.03	.10	.08	.08	.01	.06	.06	.13
Reg	.01	.01	.02		.01	.02	.03	.03	.01	.02	.02	.06
HMM	.03	.17	.31	.37		.33	.28	.31	.02	.15	.15	.36
WHMM	.02	.02	.13	.18	.03		.10	.11	.01	.05	.05	.19
KNN	.04	.08	.19	.24	.07	.19		.15	.04	.08	.08	.23
WKNN	.03	.05	.15	.20	.05	.14	.08		.03	.06	.06	.19
GPrior	.09	.28	.42	.48	.26	.45	.39	.42		.20	.20	.46
MNB	.37	.43	.53	.58	.42	.54	.50	.52	.35		.04	.55
MNBF	.38	.44	.54	.58	.42	.54	.51	.53	.35	.06		.56
GKNN	.05	.07	.11	.13	.07	.10	.09	.10	.04	.04	.04	

Table 1: The upper subtable shows the performance and execution time of all predictors on Tools with $w = 5$. Higher values are better for (balanced) accuracy; smaller values are better for RMSE. Best runs are indicated in boldface, and best runs among the text-ignorant predictors are underlined. The matrix in the lower subtable: shows the percentage of entities where the κ^+ value of the predictor in the row is higher than for the predictor in the column.

5.2 Results and Discussion

The upper part of Table 1 summarizes our results on accuracy, balanced accuracy, RMSE, κ^+ statistics and execution time. The lower part shows the pairwise comparisons on κ^+. We discuss the two subtables thereafter.

5.2.1 Prediction Quality. The upper subtable of Table 1 shows prediction quality and execution time of the algorithms. The entity-centered models are treated as one *conglomerated* global model (cf. subsection 4.3). As expected, the text-aware global models outperform the conglomerated ones, since the former exploit both the past texts and the past labels.

Among the entity-centered methods, the conglomerated model of the text-ignorant *SMA* exhibits best performance on RMSE, outperforming also the text-aware predictors *KNN, WKNN* and even the global one *GKNN*. The execution time advantage of *SMA* over other methods is remarkable (cf. last column), so it is a recommendable choice when the labels are numerical.

For categorical labels, the conglomerated model of *WPrior* shows best performance among the entity-centered text-ignorant methods, but cannot outperform the text-aware ones. The difference in predictor quality among the entity-centered methods when we

[2] http://jmcauley.ucsd.edu/data/amazon/links.html
[3] Figures and tables on Watches can be found at http://www.kmd.ovgu.de/research/oscar/2018_sac_supplementary.html

juxtapose the RMSE and the accuracy / balanced accuracy also indicates that RMSE is more appropriate when the class distribution is skewed towards one of the extreme values (5 stars in our datasets).

The prediction quality of the conglomerated models of the entity-centered *HMM, WHMM* is mediocre and their execution time is prohibitively high. Hence, text-ignorant entity-centered learning with HMMs does not seem promising.

The κ^+ of the entity-centered conglomerated models is zero. This is an obvious consequence of the fact that this κ^+ is computed over the whole stream, while the entity-centered algorithms see individual entities. When observing Fig. 1, we see that the performance of an entity-centered method for a given entity may be superior to the performance of a global method. Hence, we study the κ^+ per entity, as described hereafter.

5.2.2 Pairwise Comparison of Prediction Quality for each Entity. In the lower subtable of Table 1, we juxtapose the performance of the predictor in each row with that of the other predictors (in the columns) by computing the percentage of entities for which the predictor in the row has positive and higher κ^+ values than the predictor in the column. For example, consider the entity-centered predictor *Prior* in the first row, in juxtaposition to the entity-centered *Regression*, shortened as *Reg*, in the 4th column: the *Prior* outperforms *Reg* for 44% of the entities (the width of the green bar corresponds to 44% of the cell's width), while *Reg* outperforms *Prior* for 1% of the entities (see 4th row, 1st column). For the remaining 55% of the 33,989 entities, the two predictors perform equally.

We can see that *MNB* and *MNBF* show superior performance for up to 58% of the entities, i.e. their performance is zero or equal to that of another predictor's κ^+ for more than 40% of the entities. For 6% of the entities they are outperformed by the *SMA*, for 18% of the entities by the *Prior*, for 13% by the *WPrior.* Since all three algorithms are text-ignorant and entity-centered, this indicates that concentrating on the entities is beneficial for some entities at least. Understanding the characteristics of those entities is a matter for future work.

6 CONCLUSION AND OUTLOOK

In this study, we investigated to what extend the exploitation of the entity to which an opinion refers is predictive of the opinion's label. We proposed a framework for *entity-centered* polarity prediction, in which one model is built for each entity. This prediction model is based on the *trajectory* of past opinions on the entity. Some of our predictors are text-aware, i.e. they identify entities with similar opinions, while others exploit only past labels.

Our first results indicate that entity-centered learning is competitive for some entities, although for the majority of the entities it is better to build a model over the whole opinionated stream. Our experiments show that there is a small but not negligible percentage of entities, for which simple text-ignorant predictors are superior to a method that exploits the whole stream of opinionated documents but ignores the association between opinion and entity. A preliminary data inspection shows that entities with large changes in the polarity of opinions benefit from these simple predictors.

Our framework encompasses simple entity-based predictors that assess the next opinion label by approximating past label values with a regression line or by selecting the most frequently seen label for the entity. Our framework contains also elaborate entity-based predictors based on HMMs. However, our experiments show that the processing time needed to build one sophisticated model per entity is not accompanied by outstanding prediction quality: simpler predictors are more competitive.

Our experimental findings have some restrictions. We used two datasets (for one of them the results are described in the supplementary material) in which the class distribution is biased towards the most positive label. A less skewed distribution may have led to differences in relative performance between entity-centered and conventional stream classifiers. A further limitation emanates from the evaluation scenario itself, in which we measured overall prediction quality for the whole stream instead of performing prequential evaluation for each entity separately: this evaluation is likely to favor methods that exploit the whole stream. Moreover, differences in the number of opinions and the class distribution per entity have not been taken into account.

Our entity-centered predictors require at least a small number of opinions for an entity, before they can make predictions for further opinions. In contrast, a conventional opinion stream classifier can predict a label of an opinion even if no other opinions are there for the same entity. Hence, our approach is not an alternative but rather a low-resource complementary approach to a conventional stream learner – to be used for some entities but not for others. In our future work, we will therefore investigate hybrid approaches that combine entity-centered and entity-ignorant, text-aware and text-ignorant learners. To design such hybrids in a well-informed way, we plan to study how class distribution for each entity, number of opinions per entity and other entity properties affect the performance of entity-centered predictors.

7 ACKNOWLEDGMENTS

This work is partially funded by the German Research Foundation, project OSCAR "Opinion Stream Classification with Ensembles and Active Learners". Additionally, the first author is also partially funded by a PhD grant from the federal state of Saxony-Anhalt.

REFERENCES

[1] Joao Gama, Indre Zliobaite, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A Survey on Concept Drift Adaptation. *ACM Comput. Surv.* 46, 4 (2014).

[2] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. 507–517.

[3] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics* 37, 1 (2011), 9–27.

[4] Svitlana Vakulenko, Albert Weichselbraun, and Arno Scharl. 2016. Detection of Valid Sentiment-Target Pairs in Online Product Reviews and News Media Articles. In *2016 IEEE/WIC/ACM Int. Conf. on Web Intelligence.*

[5] Sebastian Wagner, Max Zimmermann, Eirini Ntoutsi, and Myra Spiliopoulou. 2015. Ageing-based Multinomial Naive Bayes Classifiers over Opinionated Data Streams. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECMLPKDD'15*, Vol. 9284. 401–416.

[6] Lei Zhang and Bing Liu. 2014. Aspect and Entity Extraction for Opinion Mining. In *Data Mining and Knowledge Discovery for Big Data: Methodologies, Challenge and Opportunities*, Wesley W. Chu (Ed.). 1–40.

[7] Indrė Žliobaitė, Albert Bifet, Jesse Read, Bernhard Pfahringer, and Geoff Holmes. 2015. Evaluation methods and decision theory for classification of streaming data with temporal dependence. *Machine Learning* 98, 3 (2015), 455–482.