A Semi-supervised Self-Adaptive Classifier over Opinionated Streams

Max Zimmermann[a]
Faculty of Computer Science
Otto-von-Guericke-University
of Magdeburg, Germany
max.zimmermann@iti.cs.uni-magdeburg.de

Eirini Ntoutsi
Institute for Informatics
Ludwig-Maximilians-University
of Munich, Germany
ntoutsi@dbs.ifi.lmu.de

Myra Spiliopoulou
Faculty of Computer Science
Otto-von-Guericke-University
of Magdeburg Germany
myra@iti.cs.uni-magdeburg.de

***Abstract*— We investigate the problem of polarity learning over a stream of opinionated documents. We deal with two challenges. First, if the opinions are not labeled, then we cannot assume that a human expert will be regularly and frequently available to assess the sentiment of arriving documents for learning and model adaption. Further, the vocabulary of the stream, and thus the feature space used for learning, changes over time: people use an abundancy of words, and sometimes even invent new ones to express their feelings. We propose a semi-supervised opinion stream classification algorithm that uses only an initial training set of labeled documents for polarity learning and gradually adapts to changes in the vocabulary. In particular, our algorithm *S*3Learner*starts with the vocabulary of opinionated words that are in the documents of the initial training set, and then expands it with new words, as soon as there is enough evidence for estimating their polarity. We study the performance of *S*3Learner*on opinionated streams under the natural order of document arrival and under a modified ordering that allows us to simulate vocabulary evolution.**

I. Introduction

With the omnipresence of Web 2.0 technologies, which promotes people to commit thoughts, feelings, sentiments and opinions rather than solely receive information, a huge amount of user written content in form of product reviews and microblogging entries is published online. Ordered by the time being published, the opinion-rich content sparks opinionated streams of user-generated content that are augmented everyday by tons of reviews referring to products, services and people at websites such as Amazon, CNet or Twitter. Classifying the arriving reviews as either positive or negative is the objective of opinion stream classification. It deals with streams that underly a changing environment and thus evolve over time. For example, the service of a hotel can degrade over time, products may improve their functionalities with upgrades, a competitor product with state-of-the-art features may be released and therefore it may change the impression of old products as being not up to date. Hence, the content of documents and thereby the words, used to express opinions, change over time: completely new words appear and known words might disappear.

Traditional opinion stream classification assumes that there is an arriving stream of *labeled* documents and that the feature space (the set of words) is fixed. This allows to model a document as a fixed-size vector of words, and to adapt the probability of each word given class, as the stream continues. It is necessary to depart from these simplifying assumptions. First, if a website does not enforce the people to assign polarities to their opinions (as TripAdvisor does, and as Twitter does not), then there is no stream of labeled documents, so that the classifier must be adapted without getting new labels. Further, people do use new words to express their sentiments, and they also give up ones that are used out - for example, when "cool" was not cool enough any more, "supercool" emerged.

In [1], Zimmermann et al. propose a semi-supervised stream classifier for opinionated documents, which alleviates the first problem but addresses the second one only partially. Their method *ADASTREAM* assumes that only a static set of labeled documents is available for learning: they add new documents to this training set by classifying each arriving document d and then deciding whether d would be a beneficial addition to the training set. We claim that the information needed to adapt a semi-supervised classifier is encapsulated in the words, not in the documents. Accordingly, we propose a semi-supervised stream classifier that adapts itself by assessing the polarity of newly seen words and adding those to the vocabulary, for which the polarity has been assessed to an adequately reliable extend. Following our example, "supercool" would become part of the vocabulary and used for labeling only after acquiring enough evidence that this word is positive.

Our approach has its roots in self-training [2]. Self-training classifiers [3], [4] are especially prone to classification errors as they propagate the errors to the small set of documents with true labels and thus spoiling them which weakens the performance of the classifier over time. Our contribution is an opinion classifier that: (a) uses an initial training set of documents as only evidence of true labels for training; (b) predict the label for a new arriving unlabeled document by self-trainig from all words contained in the initial seed as well as reliable unknown words seen thus far; (c) adapts while propagating this label to all the unknown words contained in the document; thus, maintaining the unknown words over time and keeping the only evidence of true labels unspoiled while not propagating predicted labels to words for which evidence of true labels is available. Our label prediction utilize only known words and unknown words which are reliable w.r.t. to the class so as to minimize the classification error. Only those words are selected as reliable which do not show any evidence of true labels but which reveal a pure class distribution and a

[a] Max Zimmermann was partially funded by the German Research Foundation project SP 572/11-1 "IMPRINT: Incremental Mining for Perennial Objects".

high frequency qualifying them as class-informative.

II. Related Work

Due to the abundance of opinionated texts nowadays, there is a lot of research recently on sentiment analysis and opinion mining. Cambria et al. [5] provide a detailed overview of the evolution of sentiment analysis research and propose a categorization of the existing approaches into: keyword spotting, lexical affinity, statistical methods and concept-based approaches. Moreover, they discuss the new trends in the area, namely multimodal sentiment analysis, based on new available sources like audio and video and contextual polarity learning for being specific to each user preferences and needs. Using the different approaches in a complementary way has been recently proposed [6]; the idea is to build meta-features from these methods and train a meta-feature classifier for boosting the sentiment analysis task. However, they focus on static data. Our approach belongs to the statistical methods category, most the related works in this category also focus on static datasets and work in a fully-supervised manner, cf. the pioneering work by Pang et al. [7]. Facing a stream of opinionated documents expects learning with adaptation to concept changes which is investigated in stream classification literature. For instance, Gama et al. [8] detect changes by monitoring the labels of new arriving instances and adjusts the model towards the most recent data. A further approach for adjusting classifiers involves re-weighting, in particular if the model is learned by a classifier ensemble, as e.g. in [9]. Bifet et al [10], [11] study sentiment classifiers dedicated to opinionated streams in a fully supervised setting, i.e. they rely on the prompt arrival of new *labeled* documents, and thus cannot be applied in a semi-supervised environment. Also, there exist methods that exploit special application characteristic like wide spread usage of emoticons in Twitter [12]. Our work is not limited to a specific application though, rather, it can be applied to any opinionated stream with an initial limited labeled set of instances. *Semi supervised learning* is ideal for cases where there is scarcity of labeled data but abundance of unlabeled ones. The idea of using unlabeled data to boost the performance of a learner when only a small number of labeled data exists is quite established and can be achieved either by considering whole examples or just some of their features. One of the first approaches by Blum and Mitchel [13] starts with a small set of labeled data with two sets of statistical independent features and uses co-training to also exploit the unlabeled data. Most of these works though focus on static scenarios. Zimmermann et al. [1] propose ADASTREAM, an adaptive semi-supervised opinion stream classifier that encompasses a forward adaptation component that expands the training set by incorporating whole unlabeled useful documents rather than single words and that therefore is more prone to errors. On the contrary, we expand the training set by adding words for which enough sentiment-evidence has been accumulated from the stream. As our experiments show, the word-based expansion of the initial seed set is more effective than the document-based expansion of ADASTREAM. Yerva et al. [14] propose an active stream learning based classifier for classifying tweets into relevant or irrelevant for a given company. Their idea is to built and maintain a company profile of positive and negative evidence words and test the tweet against the profile to decide on its class. Initially a small set of words is included but the seed set is expanded by also including words that co-occur often in the stream with words in the seed set. We also expand in a word-basis, however our approach is not topic specific but broader. Wang et al. [15] introduce a self-training approach which adapts the seed set, learned by a lexicon-based method, by adding iteratively such instances to the seed set which show a high confidence regarding their learned labels. They employ the classifier to distinguish subjective from objective documents within a static environment. We, in contrast, consider a stream of opinionated documents and utilize our classifier to differentiate among positive and negative documents. Moreover, our classifier adapts not by all words from the documents rather it augments the classifier by only class-informative words which promotes a finer-grained adaptation mechanism. Another way of knowledge expansion is by considering prior knowledge such as an ontology of positive and negative words. These approaches though do not seem appealing for streams because they encompass drifts and shifts over time. For example, Melville et al. [16] propose a sentiment analysis framework that combines prior lexical knowledge with text classification.

III. Adaptive Learning with only an Initial Seed of Polarized Words

We observe a textual stream $\mathcal{D}$ of documents arriving at distinct timepoints; each document is represented by the bag-of-words model. The incoming documents are unlabeled w.r.t. to their sentiment, i.e. there is no class information. The mining goal is to assess the labels of arriving documents.

A. *Overview of Our Approach* S*3Learner

Unlike in typical stream classification and similarly to semi-supervised classification, we assume that the only training set available is a handcrafted collection $\mathcal{S}$ of documents, to which an expert has assigned a polarity label (positive or negative). From the work of [1] on *ADASTREAM* we borrow: the notation, terming $\mathcal{S}$ as the *initial seed set* and the use of Multinomial Naive Bayes as base learner of our semi-supervised classifier. We focus on the list of unknown words and expand it gradually, adding new words as soon as enough information has been collected on their polarity.

1) Using Known and Unknown Words Vocabulary: The words in the initial seed set $\mathcal{S}$ constitute the *initial vocabulary of known words* V. As the stream progresses, the vocabulary must change: people use additional, previously unknown words to express their positive or negative opinion about some subject. This evolution can have several origins. First, people may use new words to describe a known subject. For example, a person may criticize a hotel's breakfast as "bad" (a known negative word), while another may state that "the breakfast is pathetic", associating a word of yet unknown polarity with the same subject. Second, new subjects may show up, e.g. "an overcomplicated manual", "a hairy plan" or "an overworked receptionist". These new words, on whose polarity is little known at first, constitute the *vocabulary of unknown words* U, which grows with time in contrast to the static V.

To assess the polarity of a word in U, we compute and maintain an estimate of its class distribution, using the labels predicted by the classifier for arriving documents. This

estimation is volatile, since it might change as more documents are accumulated from the stream. That is, we revise the polarity of the words in U as new evidence arrives, except that the evidence is not delivered by an expert, but by the classifier.

To learn the polarity of documents, we use Multinomial Naive Bayes (MNB) [17] and train a classifier over the initial seed set $\mathcal{S}$, denoted hereafter by $\Delta(\mathcal{S})$. Since we use the bag-of-words model, the words constitute the feature space, and the label of an opinionated document d is computed as $P(c|d) = P(c)\prod_{i=1}^{|d|} P(w_i|c)$, where $P(c)$ is the prior probability of class c and $P(w_i|c)$ is the conditional probability of word w_i belonging to c.

The estimates of the word probabilities (we use a "hat" as in $\hat{P}$, to denote estimates hereafter) are computed using the initial labeled seed set $\mathcal{S}$, i.e.

$$\hat{P}(w_i|c) = \frac{n_{ic} + 1}{\sum_{j=1}^{|V|} n_{jc} + |V|} \quad (1)$$

where n_{ic} is the number of documents in $\mathcal{S}$ belonging to c and containing w_i, $n_{ic} = |\{d \in \mathcal{S} : w_i \in d \wedge class(d) = c\}|$, applying the Laplace estimator to alleviate the zero frequency problem for words that have not been seen under a given class.

The Laplace correction is sufficient for learning over a static vocabulary. In our scenario, the extension of the initial vocabulary with new words and the estimation of their polarities requires an extension of the basis MNB learner, which we describe in subsection III-B.

2) The S*3Learner *Algorithm:* Briefly, our learning and adaption method *S*3Learner* works as follows: The initial classifier is trained upon the initial seed set $\mathcal{S}$. For each new document d from the stream, the class label for d is predicted by the current version of the classifier - this is the MNB extension we describe in subsection III-B. Based on the class prediction of d, the unknown words of d are chosen to adapt the existing classifier, i.e. the class counts of unknown words that appear in the document are updated while increasing class counts for existing unknown words. Additionally, new initial class counts and entries in the unknown vocabulary are established of such unknown words which appear for the first time. Finally, also based on the class prediction of d, the document class count is updated.

B. Semi-supervised Adaptable Multinomial Naive Bayes

Let d be a new arriving document from the stream at timepoint t. For each word $w_i \in d$ at timepoint t there are three cases: (i) $w_i \in V$, i.e. w_i might be part of the initial vocabulary V and its class distribution counts $n_{ic}, c \in C$ are known, (ii) $w_i \notin V$, $w_i \notin U$, i.e. w_i occurs for the first time in the stream and there is no information on its class distribution counts, (iii) $w_i \notin V$, $w_i \in U$, i.e. w_i does not appear in the seed set but it has appeared in the stream before and therefore belongs to the vocabulary of unknown words U and its class distribution counts $m^t_{ic}, c \in C$ are estimated from the stream. These estimations are temporary and not final since the stream is still progressing, therefore we employ the ^{t} symbol in the above notation.

In case (i), we take over the class distribution from the initial seed set $\mathcal{S}$ to compute the class conditional probabilities $\hat{P}(w_i|c), c \in C$. This is already described in Section III-A1, Equation 1. In case (ii), we use the Laplace correction to initialize the conditional probabilities in order to avoid the zero frequency problem (cf.Section III-B1). In case (iii), we estimate the conditional probabilities from the stream $\mathcal{D}$. Since the stream progresses over time, these estimations might also change over time; their maintenance is described in Section III-B2.

In order to enrich the classifier and make it adaptable over the course of the stream, we propose a combination of the vocabulary of known words V and that of unknown words U (Section III-B3).

1) Initializing the probabilities of unknown word: For an unknown word $w_i \notin V$ in a new arriving document d at timepoint t, which is not in U^t we make an initial estimate of its class probability by employing Laplace correction (similarly to cf.Section III-A1): $\hat{P}(w_i|c) = \frac{1}{|V|}$. We opt to divide by $|V|$ and not by e.g. $|U|$ because the U is ever growing and therefore, the initial probability for unknown words gets lower and lower over time. Relying on $|U|$ for the regularization would mean that words appearing later in the stream would be penalized.

2) Maintaining class distribution for unknown words: For an unknown word w_i, we maintain its class distribution based on the predicted class labels of the incoming documents by the existing classifier Δ^t. As already mentioned, this is an informed guess since it relies on the predicted and not the true class labels. Moreover, this informed guess is based on an estimation of the class distribution, which is itself temporary.

Given the current timepoint t, for each word $w_i \in U$ there is an entry $m^t_{ic}, c \in C$ keeping track of the number of times w_i was predicted in class c in the stream. The update of the above counts is as follows: For each incoming document d from the stream at timepoint t, we predict its class label $class(d) \in C$ based on the classifier Δ^t (the classifier is explained in Section III-B3). The predicted label is propagated to the document's words $w_i \in d$ and all related entries in the vocabulary of unknown words are increased by 1.

3) Updatable Multinomial Naive Bayes: As the stream evolves, the initial classifier $\Delta(\mathcal{S})$ (which was based solely on the vocabulary of known words, V) evolves through the concurrent consideration of unknown words $w_i \in U$.

The updated classifier at timepoint t, Δ^t, relies upon the known words distribution $\mathcal{N}$ and the unknown ones $\mathcal{M}^t$. The estimation of class conditional distribution for known and unknown words is different. In case of known words, the estimates $\mathcal{N}$ come from the seed set which is assumed to be reliable in terms of the class labels. In case of unknown words though, the estimates $\mathcal{M}^t$ come from the prediction of the classifier and therefore any errors in the classifier are reflected in these predictions. Moreover, there might be not enough observations for these words and therefore the class distribution estimation might be biased. For example, if an unknown word was observed just once as positive, it will affect the classification decision towards the positive class. However, that prediction might not be correct as the probability estimation is based on just one observation.

To deal with the issue of few document observations per word, we introduce the so-called *min word occurrence threshold MinFreq*. Unknown words that will be considered for classifier's update should occur in at least *MinFreq* documents. This threshold solves the poor observation issue, however except for enough word observations we are also interested in words with pure class distributions, i.e. words which have a clear sentiment. Words that equally occur to both positive and negative classes do not contribute in the classification decision and therefore are not informative for the task per se. To capture this requirement we introduce the so-called *max word entropy threshold MaxEntr*. Recall that the higher the entropy of a set, the less pure in terms of classes the result is. The entropy threshold solves the non-informative words problem and only words with a low entropy w.r.t. the *MaxEntr* threshold are allowed to adapt the classifier.

We introduce the observed entropy of a word w_i at a given time t for words belonging to the unknown vocabulary U.

$$ObservEntr(w_i)^t = \begin{cases} -\sum_{c \in C} H(w_i, c)^t, & \text{if } m_i^t \geq \textit{MinFreq} \\ 1, & otherwise \end{cases} \quad (2)$$

where $H(w_i, c)^t = \hat{p}_{ic}^t \log_2 \hat{p}_{ic}^t$ according to the unknown word observations, i.e. $\hat{p}_{ic}^t = \frac{m_{ic}^t}{m_j^t}$ and $m_i^t = |\{d : d \in \mathcal{D}^t \wedge w_i \in d\}|$. That is, the observed entropy is equal to the *Shannon entropy*, which is based on the probabilities of observing w_i in classes $c \in C$, if there are more than *MinFreq* observations of this word in the incoming documents. Otherwise the entropy is 1, i.e. the maximum value for a 2-class classification problem.

Only words whose entropy is less than the entropy threshold *MaxEntr* are considered for classifier's update. Therefore the number of documents containing the word w_i and belonging to class $c \in C$ is filtered according to this threshold. The filtered number $\hat{m}_{ic}^t$ is given as follows:

$$\hat{m}_{ic}^t = \begin{cases} m_{ic}^t & \text{if } ObservEntr(w_i)^t < \textit{MaxEntr} \\ 0 & otherwise \end{cases}$$

The value 0 for words that violate the entropy threshold means actually that these words contribute to the classifier no more than initialzed unknown words, i.e. $1/|V|$.

Similarly, we define the filtered number of documents from the stream $\mathcal{D}$ belonging to class c till time t:

$\hat{m}_c^t = |\{d : d \in D^t \wedge class(d) = c \wedge \exists w_i \in d : ObservEntr(w_i)^t \leq \textit{MaxEntr}\}|$

These are the documents which were predicted to belong to class c and contain at least one word for which the $ObservEntr$ is below or equal to *MaxEntr*.

The new classification model that makes use of both, the known vocabulary V and the unknown vocabulary U, is defined by Equation 3:

Definition 1 (Updatable Multinomial Naive Bayes): The class label of a new document d arriving from the stream at timepoint t is the one maximizing the posterior probability of the document being generated by the class. The class prior estimations and the word class conditional estimations make use of both the vocabulary of known words V and of unknown words U. In the first case, the probabilities are derived from the seed set of true class labels whereas in the second one the estimates come from the observed word-class occurrences in the stream where the class information is derived from the classifier.

$$class(d) = \operatorname*{argmax}_c \frac{\hat{m}_c^t + n_c}{\sum_{c \in C} \hat{m}_c^t + n_c} * \prod_{w_i \in d} \hat{P}(w_i|c)_{filtered} \quad (3)$$

where,

$$\hat{P}(w_i|c)_{filtered} = \begin{cases} \frac{n_{ic}+1}{\sum_{w_j \in V \cup U} (\hat{m}_{jc}^t + n_{jc}) + |V|} & \text{if } w_i \in V \\ \frac{\hat{m}_{ic}^t + 1}{\sum_{w_j \in V \cup U} (\hat{m}_{jc}^t + n_{jc}) + |V|} & otherwise \end{cases}$$

IV. Experiments

To evaluate *S*3Learner*, we experiment with two real world datasets of opinionated documents (product reviews and tweets, Section IV-A). The original datasets come with natural ordering. In order though to show the effect of unknown words in the performance of our method, we re-ordered the datasets so that the ratio of unknown words increases gradually over time (cf.Section IV-A1). We experiment with both versions.

A. Datasets

Stream Review comes from a dataset first introduced by Yu et al. [18]. The true labels of the reviews were made by the authors themselves derived from star-ratings. The reviews cover mostly products and their properties such as "phone", "firmware" and "price". We use only reviews describing single product features, after removing very short reviews containing less than 2 adjectives. Stream Review contains 13.650 product reviews and was partitioned into 273 batches of 50 reviews. The dataset is skewed towards the negative class, at each batch both classes are present though. The dataset is available. [2]

Stream Twitter, first introduced in [19] [3], comes from [1] as explained there, the stream was collected by querying the (non-streaming) Twitter API for messages between April 2009 and June 25, 2009. The stream is very heterogeneous regarding the content as it captures several domains such as products, movies, locations etc. The true labels (ground truth) of the tweets were acquired through the Maximum Entropy classifier using emoticons as class labels. The original stream contains 1.600.000 tweets, where the class distribution in the first 1.450.000 tweets is skewed towards the positive class while the last 250.000 tweets are only from the negative class. Since we are interested in investigating our approach according to drifts within the class distribution, we take a snippet of the original stream which captures the drift by maintaining the original order. In particular, the shortened stream contains the tweets 1.235.000 - 1.485.000, i.e. 250.000 tweets, which were partitioned into 500 batches of 500 tweets.

In Review we focused only on adjectives and adverbs for sentiment analysis since these words bear the actual opinion of the author [20], [21], while stream Twitter comes with nouns and verbs as stated in [19], cf.Section IV-F for the effect of

[2] https://www.dropbox.com/s/8d0z8v6j3qoxk4j/datasetReviewJI.zip

[3] Available at: http://help.sentiment140.com/for-students

nouns and verbs. The Twitter dataset contains almost twice as much opinion bearing words (6 per document) in comparison to the review dataset Review (3.5 per document) albeit Twitter allows only 140 characters per tweet. Hence, the authors of tweets use in average a higher variety of sentimental words.

1) The effect of unknown words over the stream: To show the effect of unknown words, we re-ordered the original streams in such a way that the number of known words decreases over time whereas the number of unknown ones increases. For each original stream we "designed" its re-ordered counterpart as follows: the stream begins with documents that contain only words from $\mathcal{S}$; the number of unknown words increases as the stream progresses, i.e. the ratio of words from $\mathcal{S}$ to all words in the documents drops gradually. The percentage of known and unknown words per document over time for the re-ordered and natural ordered version of the streams is drawn in Figure 1. For the unknown words, we distinguish between first-time observed unknown words (in gray) and already monitored unknown words (in blue).

For the *natural order* of the streams, we receive a high number of known and an increasing number of unknown words. The firs-time unknown words are more in the beginning of the stream but over time the number of already monitored unknown words gets higher. First time appearing unknown words exist at all timepoints, showing that new content is added over time from the stream. Their number is higher for stream Twitter compared to stream Review (gray area dominates blue area in the second right and very right picture of Figure 1), because the first one covers a wide variety of topics whereas the second dataset refers to product reviews only. In the *re-ordered* versions the number of unknown words is increasing over time and after some point the stream bears merely unknown words. However, the number of first-time observed words is rather static over time showing a continuously increasing variety of words. The reason for re-ordering is to show the performance in extreme/ hard cases.

2) The class distribution over the stream: The class distribution of stream Review for the natural order is almost stationary over time and skewed towards the negative class. Whereas the distribution of Twitter is slightly skewed towards the positive class until the end of the observation period where it consumes only negative documents. The re-ordered stream depicts a more fluctuating behavior: it is also slightly skewed towards the positive class but shows up several sudden changes where it consumes, for a while, only negative tweets. Hence, with stream Review we capture the case of facing an almost stationary class distribution while dealing on the one hand with many known words (natural order) and on the other hand with only few or no known words (re-ordered). With stream Twitter on the contrary, we evaluate how our method performs on a shifting class distribution.

B. Learning methods and quality measures

Below we outline the approaches we used to compare to *S*3Learner*. They are all based on Naive Bayes but do not employ word filtering. In particular, they differ w.r.t. whether i) the classifier is adapted based on new documents from the stream, ii) the adaptation is done on the basis of the true or the predicted class labels and iii), which part of the vocabulary is adapted, V, U or both? They are decribed as follows.

V:static: The vocabulary of known words V is static and the unknown words U are not considered. This is the static case, where no errors in the class label predictions of incoming documents are propagated to the classifier but neither the classifier is updated by new content.

V:adaptedByPredictedLabels: V and existing word-class counts $\mathcal{N}$ are updated by propagating the predicted class labels of incoming documents. However, unknown words are not considered. Errors in the class label predictions are propagated.

V&U:adaptedByTrueLabels: There is no differentiation among known and unknown words, both are updated gradually based on the true class labels (full supervised case). There is *no* word filtering on unknown words, all words contribute to the classifier. No errors in the label predictions are propagated.

V:static&U:adaptedByTrueLabels: V is static, we only adapt U and $\mathcal{M}$ by true class labels (fully supervised). No errors in the class label predictions are propagated.

Moreover, we evaluate *S*3Learner* to the document-based approach of [1], namely ADASTREAM. Based on the so-called *usefulness* threshold it expands the initial seed set by considering whole documents rather than single words. The greater the value of the *usefulness* threshold, the more documents expand the seed and thus more adaptation is taken into account. The results are discussed in Section IV-C.

For evaluating the classifier's quality, we use *kappa statistic*. The kappa statistic [10] normalizes accuracy by that of a chance classifier: $k = \frac{p_0 - p_c}{1 - p_c}$ p_0 is the accuracy of a classifier and p_c is the probability of making a correct prediction by a chance classifier that assigns the same number of examples to each class as the classifier under consideration. The kappa varies among -1 and 1: a value ≤ 0 indicates that the classifier's predictions coincide with, or are worse, than the predictions of the chance classifier. A value > 0 implies that the classifier's predictions overcome these of a chance classifier. The higher the value, the more often the predictions match with the true labels. Kappa is preferred to accuracy for data streams as it is not prone to unbalanced class distributions.

C. Comparing the baselines

In this section, we compare *S*3Learner* against the two supervised baselines, the two semi-supervised baselines and the approach of [1], cf.Section IV-B, based on the performance of kappa over time. We use 0.0 as value for *usefulness* threshold as it shows the best performance for ADASTREAM. We examine the performance of *S*3Learner* on the natural order and the re-ordered version of the two datasets.

1) Results on stream Review: Figure 2 depicts kappa for the compared approaches and *S*3Learner* on stream Review natural order (first picture) and re-ordered (second picture). We utilize a seed set of 140 documents to show how *S*3Learner* performs on a large amount of unknown words, i.e. less influence of true labels. The results on the natural order of stream Review show how *S*3Learner* performs on a large but rather static amount of unknown words over time, while the re-ordered version of stream Review exposes how *S*3Learner* performs on a large and also increasing amount of unknown words.

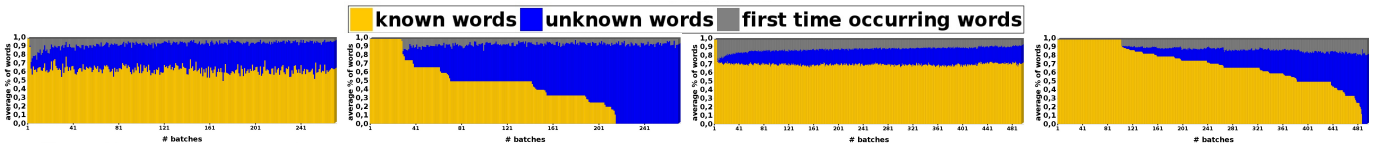

Fig. 1. % of known, unknown and first appearing words over time (avg per batch) for sream Reviewnatural order (left) and re-ordered (second left), $|\mathcal{S}|$=140 and for stream Twitter natural order (second right), $|\mathcal{S}|$=2.500, and re-ordered (right), $|\mathcal{S}|$=10.000.

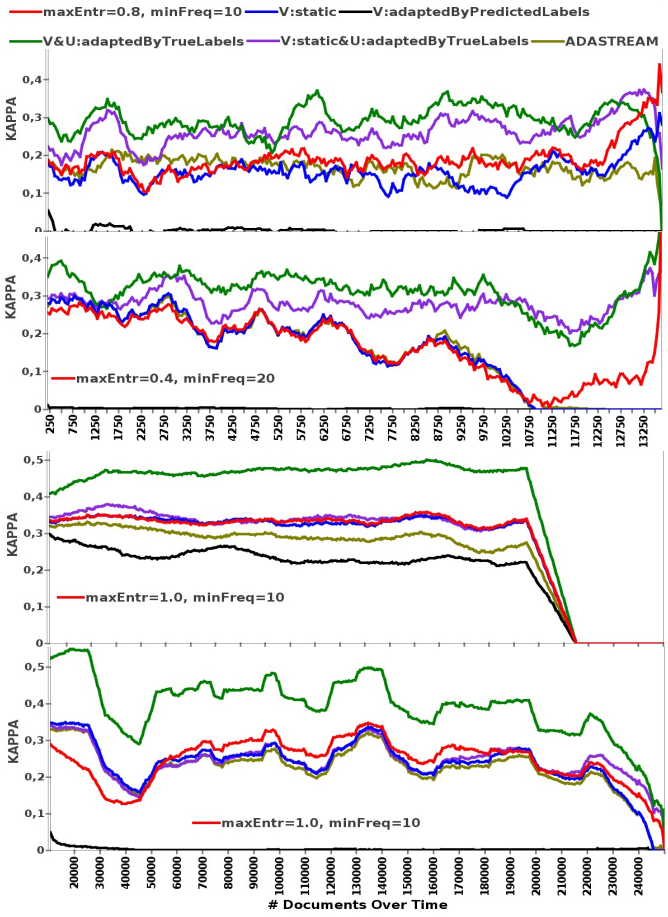

Fig. 2. Kappa for the four baselines plus ADASTREAM and one setting of *MaxEntr* and *MinFreq* for *S*3Learner* of stream Review and Twitter natural order (first,third), re-ordered (second,fourth), drawn as (*MaxEntr*,*MinFreq*), $|\mathcal{S}|$=140;10.000

*S*3Learner* reveals the best kappa across the semi-supervised approaches for a large but static amount of unknown words depicted in the first picture of Figure 2; while the two supervised approaches show the best kappa values among all approaches. However, the adaptation mechanism of *S*3Learner* based on the filtering by the two thresholds works well since the kappa increases as the stream progresses and even overcomes the supervised baselines at the end of the stream. We use 0.8 and 10 as values for threshold *MaxEntr* resp. *MinFreq* as this setting shows the best performance.

Facing a large and increasing amount unknown words exposes that the supervised methods show a kappa being rather constant over time whereas the semi-supervised approaches, including ours, draw a decreasing kappa over time, cf. second from abve picture of Figure 2. This is the case because the amount of unknown words increases over time till only unknown words arrive, cf. very left picture of Figure 1, and so the influence of documents with true labels for the class prediction of unlabeled documents decreases. Across the semi-supervised approaches, our method carries out though the best performance while showing an increasing kappa when there are only unknown words arriving. Hence, the adaptation mechanism of *S*3Learner* works very well so that even documents carrying only unknown words, i.e. having no impact of true labels, can be correctly classified.

2) Results on stream Twitter: The results on stream Twitter natural order (third) and re-ordered (fourth) for the compared approaches and *S*3Learner* are shown by Figure 2. In case of the natural order of stream Twitter we use a seed size of 2500 tweets while we evaluate our experiments on the re-ordered version of Twitter with a seed size of 10.000 tweets. In both streams, we use *MaxEntr*=1 and *MinFreq*=10 showing the best kappa values over time for our method. In both streams, the fully supervised approach which adapts the seed set as well as the unknown words U by true labels, draws the best kappa over time. The fully supervised but only adaptive on U baseline, i.e. the seed set is kept static, does not perform well on stream Twitter though. This might indicate that the seed set captures most of that part of the stream which is affected by changes over time, cf. Section IV-E for more information on impact of the seed size.

The experiments on the natural order of stream Twitter exhibit a constant kappa for the compared approaches and *S*3Learner*. Whereas there are obvious differences of kappa among the approaches: the lowest kappa is drawn by the baseline which adapts U by predicted labels followed by the document filtering approach proposed by [1]. *S*3Learner*performs rather similar to the supervised and fully static baseline. Hence, our methods perform well when the unknown part of the stream does not capture much changes, i.e. the most changing content is captured by the seed.

The re-ordered version of stream Twitter shows much fluctuation towards the document class distribution, i.e. there are sudden changes in the distribution receiving only negative documents. *S*3Learner* deals with these changes better than all compared approaches apart from the fully supervised and adaptive baseline: it draws a small kappa at the beginning but soon, as the stream progresses, the kappa increases and overcomes the baselines and maintains that advantage till the end of the recorded stream.

D. *Impact of* MaxEntr *and* MinFreq *thresholds*

This section discusses the entropy *MaxEntr* and word frequency threshold *MinFreq* filtering words from the stream which are not class-informative, i.e. words which have a word

class distribution that is too mixed to be considered or they occurred too less times in order to be class-informative. We examine the effects of drastic filtering, i.e. only words with a pure class distribution and a high frequency are maintained, and of calm filtering, i.e. also words with a mixed class distribution and a small frequency are maintained.

We did experiments on various settings of *MaxEntr* and *MinFreq*, observing that drastic filtering caused by a small *MaxEntr* together with a big value of *MinFreq* does not expose a good performance by *S*3Learner*. The reason is that many words are filtered and thus being treated similarly by *S*3Learner* as their conditional probabilities are equal to the laplace correction. This is considered as the zero frequency problem mentioned in Section III-A1. Moreover, the experiments reveal that calm filtering, caused by a low value of *MinFreq* and a high value of *MaxEntr*, influences *S*3Learner* negatively. Showing up a bad performance that becomes worse over time as all unknown words are considered. Such unknown words are maintained which are not very class-informative neither they have a pure class distribution impairing a clear decision on the class label. Hence, classification errors are heavily promoted and propagated through the stream dropping the performance over time.

*S*3Learner* performs best when applying calm filtering by one threshold while having a drastic filtering on the other threshold. The decision across the thresholds depends on the structure of the stream. Considering Twitter which has huge variation of words through the stream, depicted by the gray colored bars in the second right and very right pictures of Figure 1, comes up with many words being observed only few times and thus having a biased class distribution. There, *S*3Learner* performs best when using *MinFreq* as drastic word filter (relatively small value) and *MaxEntr* as calm word filter (a value close to 1). This can be seen by the upper picture of Figure 3 where we fixed a small value for *MinFreq*while changing the values for the *MaxEntr* threshold from 0.1-1.0. The picture exposes the best performance for *MaxEntr*=1.0 and drops in performance when decreasing *MaxEntr*. So, our algorithm filters many words that bias the classifier and regulates the amount of unknown words while allowing words to contribute which might have a mixed class distribution. Hence, the zero frequency problem can be avoided.

The word distribution for the Review dataset in the left pictures of Figure 1 shows only a slight variety of words, depicted by a small gray bar through the stream, so the influence of first-time observed words is less than for stream Twitter. However, there is a large amount of unknown words, shown by the blue bar left pictures of Figure 1, which promotes many mixed class distributions for the unknown words. To show the less influence of first-time observed words when selecting a satisfying *MaxEntr* threshold that does not allow the contribution of to many words with mixed class distribution, we employed *S*3Learner* with a small value for *MaxEntr* while varying the word frequency threshold *MinFreq* from 1-100. Bottom picture of Figure 3 shows a stable performance of *S*3Learner* along different settings of *MinFreq* while keeping a static value of 0.4 for *MaxEntr*. Hence, our algorithm is not affected by first-time observed words if there amount is small and if the value of the *MaxEntr* threshold is selected carefully so that words with a mixed class distribution are prevented to

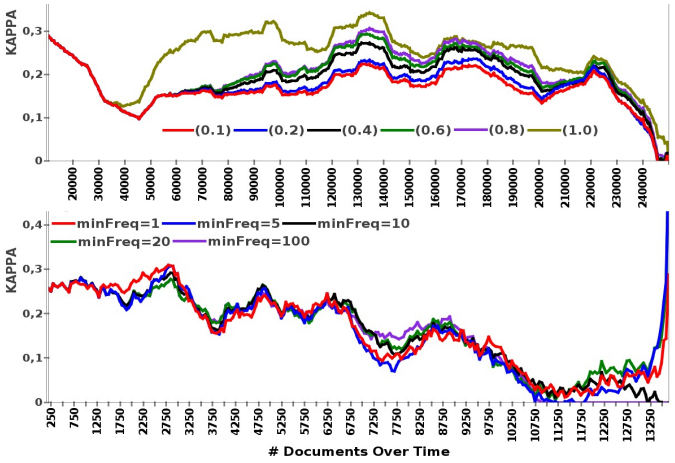

Fig. 3. Twitter: Kappa over time on re-ordered for different settings of *MaxEntr* and a fixed *MinFreq*=10, $|\mathcal{S}|$=10.000.

contribute.

E. Impact of the volume of known words V

We examine how the performance of *S*3Learner* is affected by the size of the vocabulary of known words V. Recall that the documents in S reflect the only evidence of true class lavels. Therefore, we experiment with different sizes of S. For both streams, we select values corresponding roughly to 1%, 2%, 4%, 6%, 8%, 10% and 12% of the related stream. As datasets, we used the natural ordered streams in these experiments, since the ordering of the stream remains the same in this case allowing us to compare across different seed sizes.

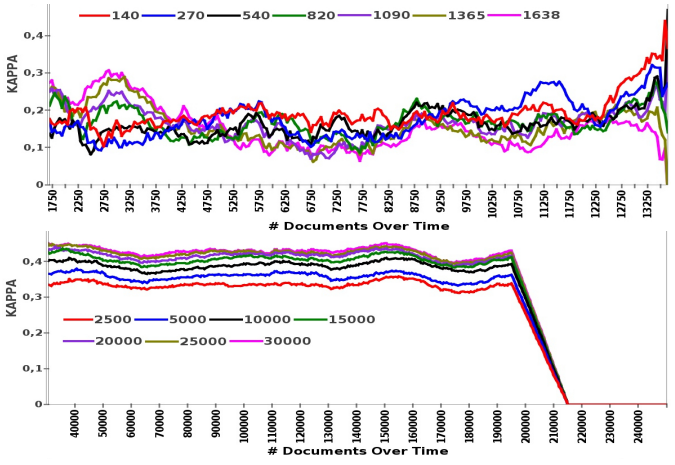

Fig. 4. Kappa for various seed sizes $|\mathcal{S}|$ for Review (above 0.8;10) and Twitter (below 1.0;10).

The kappa on streams Review, Twitter for different $|S|$ is depicted in Figure 4. For stream Twitter, the bigger $|S|$ is and therefore the larger the amount of known words, the better the resulting kappa is in general. However as S becomes larger, there is no big difference in kappa: doubling the seed size has a clear benefit in the beginning but after $|S| = 10.000$ the performance improvement is getting lower. Our experiments on stream Review reveal similar results: showing for a large amount of known words a high kappa

at the beginning of the stream while, as the stream progresses, smaller seed sizes perform better. Hence, the large seed sets might capture most of the variety of words so that no more unknown words can occur over time. Since *S*3Learner* adapts only by unknown words, willing not to violate word class distributions obtained from true labels, it is not capable to reflect emerging changes in population induced by known words. In fact, *S*3Learner* works well when the seed set is not to large while capturing the complete variety of words captured by the stream.

F. Impact of nouns and vebs

Finally we study the performance of *S*3Learner* when considering nouns and verbs additionally to adjectives and adverbs as words of the stream. We focus on the naturally ordered stream Review. As stated in Section IV-A and in [20], [21], adjectives and adverbs are preferred over nouns and verbs since they bear the actual opinion of the author. Figure 5 depicts the kappa for *S*3Learner* and ADASTREAM: *S*3Learner* shows a significantly higher kappa over time when using adjectives and adverbs only; *ADASTREAM* exposes a slightly higher kappa ignoring nouns and verbs. Hence, *S*3Learner* and *ADASTREAM* perform better when only adjectives and adverbs are considered as words of the stream.

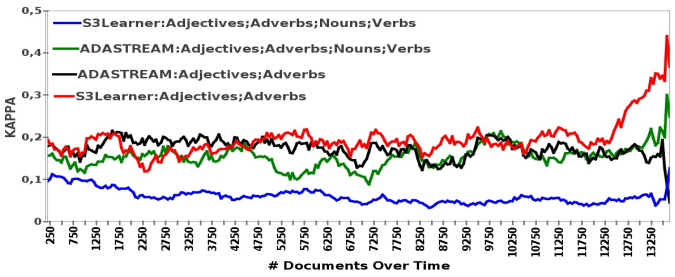

Fig. 5. Kappa for *S*3Learner* (10,0.8) and ADASTREAM (usefulness=0.0) on stream Review natural order, $|\mathcal{S}|$=140

V. Conclusion

We study the problem of opinion stream classification, with only a small initial seed of labeled documents, when the vector of words evolves over time, i.e. new words appear and old words disappear. We cope with the challenge of adapting the model by class predictions rather than true labels as they are not available for new arriving documents also we deal with changes of the used words to express opinions. We propose an opinion stream classifier that utilizes the only evidence of true labels (the seed) most effectively while not allowing classification errors being propagated to the seed set. But adapts by maintaining the class distributions of the unknown words, i.e. words not part of the seed, w.r.t. to the class label predictions of related documents. We, however, adapt only by reliable unknown words. In particular we quantify the reliability of unknown words using entropy and word frequency as basis. Our experiments on real-word streams show that our method suits very well to the changing evironment of opinion stream mining where only few labeled documents are available and words expressing opinions change rather often. The experiments reveal that our method overcomes the fully-supervised approaches when the selected seed is small, i.e. the stream captures many unknown words. Observing a seed with few unknown words appearing through the stream, we are competitive with fully-supervised approaches and overcome the compared semi-supervised methods. Future work include more elaborated mechanisms to find reliable words. We further want to investigate how our method performs when the concept of words changes, i.e. words which express positive sentiments change as being used to express negative opinions.

References

[1] M. Zimmermann, E. Ntoutsi, and M. Spiliopoulou, "Adaptive semi supervised opinion classifier with forgetting mechanism," in *SAC*, 2014.

[2] S. Fralick, "Learning to recognize patterns without a teacher," *IEEE Trans. Inf. Theor.*, vol. 13, no. 1, pp. 57–64, Jan. 1967.

[3] B. Drury, L. Torgo, and J. J. Almeida, "Classifying news stories with a constrained learning strategy to estimate the direction of a market index," *IJCSA*, vol. 9, no. 1, pp. 1–22, 2012.

[4] Y. He and D. Zhou, "Self-training from labeled features for sentiment analysis," *Inf. Process. Manage.*, vol. 47, no. 4, pp. 606–616, Jul. 2011.

[5] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *Intelligent Systems, IEEE*, vol. 28, no. 2, pp. 15–21, March 2013.

[6] F. Bravo-Marquez, M. Mendoza, and B. Poblete, "Meta-level sentiment models for big social data analysis," *Knowl.-Based Syst.*, vol. 69, pp. 86–99, 2014.

[7] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *EMNLP*, 2002.

[8] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence–SBIA 2004*, 2004.

[9] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *KDD*, 2009.

[10] A. Bifet and E. Frank, "Sentiment knowledge discovery in twitter streaming data," in *Discovery Science*, 2010.

[11] A. Bifet, G. Holmes, and B. Pfahringer, "Moa-tweetreader: real-time analysis in twitter streaming data," in *Discovery Science*, 2011.

[12] F. Bravo-Marquez, M. Mendoza, and B. Poblete, "Combining strengths, emotions and polarities for boosting twitter sentiment analysis," in *Workshop on Issues of Sentiment Discovery and Opinion Mining*, 2013.

[13] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, 1998.

[14] S. R. Yerva, Z. Miklós, and K. Aberer, "Entity-based classification of twitter messages," *IJCSA*, vol. 9, no. 1, pp. 88–115, 2012.

[15] D. Wang and Y. Liu, "A cross-corpus study of unsupervised subjectivity identification based on calibrated em," in *WASSA*, 2011.

[16] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," in *KDD*, 2009.

[17] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI-98 Workshop*, 1998.

[18] J. Yu, Z.-J. Zha, M. Wang, K. Wang, and T.-S. Chua, "Domain-assisted product aspect hierarchy generation: Towards hierarchical organization of unstructured consumer reviews," in *EMNLP*, 2011.

[19] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Proc.*, pp. 1–6, 2009.

[20] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *ACL*, 2002.

[21] H. Yu and V. Hatzivassiloglou, "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences," in *EMNLP*, 2003.