Active Learning for Entity Filtering in Microblog Streams

Damiano Spina[†]
damiano.spina@rmit.edu.au

Maria-Hendrike Peetz[‡,*]
mariahendrike.peetz@gmail.com

Maarten de Rijke[‡]
derijke@uva.nl

[†] RMIT University, Melbourne, Australia
[‡] University of Amsterdam, Amsterdam, The Netherlands

ABSTRACT

Monitoring the reputation of entities such as companies or brands in microblog streams (e.g., Twitter) starts by selecting mentions that are related to the entity of interest. Entities are often ambiguous (e.g., "Jaguar" or "Ford") and effective methods for selectively removing non-relevant mentions often use background knowledge obtained from domain experts. Manual annotations by experts, however, are costly. We therefore approach the problem of entity filtering with active learning, thereby reducing the annotation load for experts. To this end, we use a strong passive baseline and analyze different sampling methods for selecting samples for annotation. We find that margin sampling—an informative type of sampling that considers the distance to the hyperplane used for class separation—can effectively be used for entity filtering and can significantly reduce the cost of annotating initial training data.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

Keywords

Text classification; Entity filtering; Active learning; Twitter

1. INTRODUCTION

With increasing volumes of social media data, monitoring and analyzing this data is a vital part of the marketing strategy of businesses. The extraction of topics, conversations, and trends around an entity (such as a company, organization, celebrity) allows analysts to understand and manage the entity's reputation. It is infeasible to manually process every single tweet or blogpost that may have been written about an entity. Since entity names are often ambiguous, filtering social media for relevant information—that is, Entity Filtering (EF)—saves tedious work and is a vital preprocessing step for further automation of Online Reputation Monitoring (ORM) [8, 12, 13]. If the performance of the EF module decreases, the performance of all subsequent modules is harmed [13]. EF on social media is therefore an active field of research and has previously been considered in various settings: at the WePS-3 evaluation effort [1] and as part of the RepLab 2012 and 2013 challenges [2, 3]. Missing important tweets and news items about an entity of interest can be disastrous and expensive [9]. An entity may need to react immediately to avoid long-lasting harmful publicity. The ORM industry therefore seeks to find a balance between manual and automatic filtering.

Our approach to EF is based on active learning [11], a semi-automatic machine learning process interacting with the user for updating the classification model. It selects instances that are meant to maximize the classification performance with minimal annotation effort. Active learning is especially attractive in the setting of EF for ORM as it promises to (a) use the analysts' background knowledge and understanding to improve performance, and (b) capture new topics and problems without exhaustive annotation effort. Active learning has been widely used in information access tasks [10, 16] and text categorization [7]. Below, we present an active learning framework for EF.[1] We start from simple but competitive passive baselines and then examine alternative strategies for sampling examples for learning and their applicability for EF.

2. APPROACH

In this section we elaborate our active learning approach to EF. Briefly, (1) instances are represented as feature vectors; (2) instances from the training dataset are used for building the initial classification model; (3) test instances are automatically classified using the initial model; (4) we sample candidates for additional labeling; this step is performed by margin sampling: the instance closest to the class separation is selected; and (5) the user manually inspects the instance and labels it; the labeled instance is then considered when updating the model. The active learning process is repeated until a termination condition is satisfied.

We use a Support Vector Machine[2] (SVM) classifier. Our active learning approach can be split into the *selection of candidates* for active annotations, *annotation of the candidates* and *updating the model*. Therefore, one iteration of our learning model follows the following three steps: (1) select the best candidate x from the test set T; (2) annotate the candidate x; and (3) update the model.

If the resources are available, the training data used to initialize the model can be a large manually annotated (bulk) set of tweets published before the test set. Below we detail the candidate selection, candidate annotation, and model updating steps.

2.1 Candidate selection and annotation

Candidate selection is the process of sampling candidates that are used for annotation. A successful selection approach selects candi-

*Now at Google Switzerland.

SIGIR '15, August 09–13, 2015, Santiago, Chile.

ACM 978-1-4503-3621-5/15/08 ... $15.00.
DOI: http://dx.doi.org/10.1145/2766462.2767839

[1] The code is available at http://damiano.github.io/al-ef
[2] http://scikit-learn.org/stable/modules/svm.html

dates that, when annotated, improve the model. Standard baseline approaches are: *passive learning* without sampling, which is identical to non-active learning, *random sampling*, which samples randomly from the pool, and *margin sampling*, which samples close to the margin of the classification boundary. We also propose two further approaches to improve margin sampling. For *reranking*, we rerank the list of samples based on density.

Passive learning. Passive learning does not use any active learning at all. We only initialize the model without retraining it, therefore skipping the candidate selection, training, and updating phase.

Random sampling. Here, the candidate instance is sampled without replacement from the training set. There is no informed prior on the instances. Random sampling has proven to be effective for other tasks, e.g., building dependency treebanks [5], or clinical text classification [6].

Margin sampling. The most commonly used sampling method in binary classification problems is uncertainty sampling [11]. We consider a specific uncertainty sampling method especially suitable for support vector machines [15]: *margin sampling*. We measure the uncertainty of a candidate x based on the distance to the margin:

$$\text{Uncertainty}(x) = 1 - \left|P(C_1 \mid F_x) - P(C_2 \mid F_x)\right|, \quad (1)$$

where $P(C_1 \mid F_x)$ and $P(C_2 \mid F_x)$ are the probabilities that the candidate x, as represented by the feature vector F_x, generates the classes C_1 and C_2, respectively.

Candidates are sampled based on the classification difficulty, thereby selecting candidates where the classifier is less confident. Following this, the candidate x to be annotated from the test set T is selected as follows:

$$x = \arg\max_{x_i \in T} \text{Uncertainty}(x_i). \quad (2)$$

This candidate x is then annotated and used to update the model. For a linear kernel of the SVM this means: instances (tweets here) that are closest to the class separation are selected.

*Margin*Density.* Following [17], we incorporate the density of a candidate into the maximization criterion of a candidate ranker. Intuitively, while outliers may be difficult to predict, they are also not very helpful in improving the classifier:

$$\text{K-Density}(x) = \sum_{x_i \in \text{KNN}(x,T,K)} \frac{\text{sim}(x,x_i)}{K}, \quad (3)$$

where $\text{KNN}(x, T, K)$ are the K most similar instances to x in the test set T. The similarity $\text{sim}(x, x_i)$ between two instances x, x_i is based on the Jaccard similarity. We say that the $\text{K-Density}(x)$ is the $\text{Density}(x)$, if K is dynamically set to $|T|$.

In the Margin*Density setting, the candidate x to be annotated from the test set T is selected as follows:

$$x = \arg\max_{x_i \in T} \text{Uncertainty}(x_i) \cdot D(x_i), \quad (4)$$

where $D(x_i)$ is a placeholder for $\text{K-Density}(x)$ or $\text{Density}(x)$.

Once the candidates are selected, the algorithm collects annotations from the user. Section 3.2 details how we simulate the user input.

2.2 Model updating

The training of the model is fast.[3] We therefore decided to *retrain* the model with *every* freshly annotated instance. The instance and its annotation are added to the training set and the model is retrained. As commonly done, the weights for both training and new instances are uniform.

3. EXPERIMENTAL SETUP

We aim to analyze the effectiveness of active learning for EF. Below, we describe the dataset, the feedback scenario, and how we evaluate.

3.1 Data

We use the RepLab2013 [3] dataset, which is, to our knowledge, the largest dataset available for the EF task in microblog posts.[4] The dataset comprises a total of $142,527$ tweets in two languages: English and Spanish. The dataset consists of 61 entities in four domains: automotive, banking, universities and music. For every company, 750 $(1,500)$ tweets were used as training (test) set, on average, with the beginning of the training and test set being six months apart. Crawling was performed from June 1, 2012 to December 31, 2012 using each entity's canonical name as query (e.g., "stanford" for Stanford University).

Tweets are represented as set-of-words: bag-of-words with binary occurrence (1 if the word is present in the tweet, 0 if not). We removed punctuation, lowercasing, tokenizing by white spaces, reducing multiple repetitions of characters (from n to 2), and stopwords.

3.2 Scenario

Without direct users, the usual approach to model an active learning setting is to take the annotations from the test set. This simulates the user feedback; this is also what we do. We therefore train on the dedicated training set and sample from the entire test set.

For our experiments we use Support Vector Machines, using a linear kernel.[5] The penalty parameter C is automatically adjusted by weights inversely proportional to class frequencies. We use the default values for the rest of parameters.

We compare the effectiveness using different N_{test} of sampled tweets with the effectiveness of two passive supervised learning approaches: the initial model and the best approach at RepLab2013. We compare random sampling, margin sampling, and the diverse instantiations of density sampling methods listed in the previous section. Table 1 provides an overview over the acronyms used for the runs that we consider. The *passive* run is the underlying baseline for active learning; it is based on the training set. The *best* run is the score for the best performing system at RepLab2013. *RS* and *MS* are active learning runs, using random and margin sampling,

Table 1: Runs used in our experiments. MSD and MS-RRD can be combined with a K for K-Density.

Acronym	Active	Description
passive	no	Passive learning, lower bound
best	no	Best RepLab2013 system
RS	yes	Random sampling
MS	yes	Margin sampling
MSD	yes	Margin*Density sampling
MS-RRD	yes	Reranking MS based on density

[3] In our experiments, a few dozen seconds on a workstation with 16 cores, 2.6GHz, and 96GB RAM workstation. In the ORM scenario, the training set is entity-oriented and would typically not include more than a thousand tweets.

[4] http://nlp.uned.es/replab2013

[5] We tested different algorithms (Naïve Bayes, Decision Trees) and this is the one that obtained the best results in terms of the initial (passive learning) model.

respectively. *MSD* combines margin with density sampling, based on the candidate set. Finally, *MS-RRD* reranks margin sampling based on density.

3.3 Evaluation

Unless stated otherwise, we use the official evaluation metrics from the RepLab2013 Filtering Subtask: accuracy and the harmonic mean of reliability and sensitivity ($F_1(R,S)$) [4]. We use the Student's t-test to evaluate the significance of observed differences, using Bonferroni normalization where appropriate. We denote significant improvements with $^{\blacktriangle}$ and $^{\triangle}$ ($p < 0.01$ and $p < 0.05$, respectively). Likewise, $^{\blacktriangledown}$ and $^{\triangledown}$ denote declines.

4. RESULTS AND ANALYSIS

We analyze the passive baseline in Section 4.1. We then discuss the performance of margin sampling and random sampling in Section 4.2 and the impact of training in Section 4.3. Section 4.4 analyses the performance of density sampling.

4.1 Passive baseline

In order to establish our passive baseline as state of the art for EF, we compare it with the best performing system at RepLab.

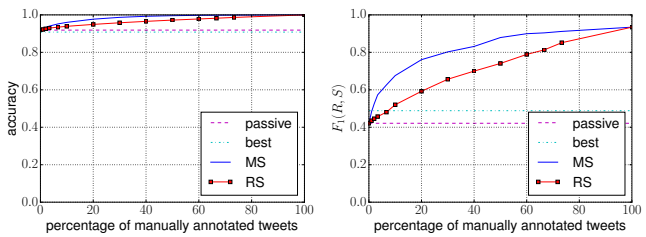
Table 2: Performance of *passive* baseline and *best* RepLab2013 system.

Run	Accuracy	$\mathbf{F_1(R,S)}$
best	0.91	0.49
passive	0.92	0.42

Table 2 compares the effectiveness of *best* and *passive* runs in terms of accuracy and $F_1(R,S)$. A number of observations are worth making. First, in terms of accuracy, the effectiveness of all the runs is above 0.9, leaving little room for improvement. The *passive* run outperforms the *best* run, but the difference is not statistically significant. In contrast, $F_1(R,S)$ reveals more differences between the runs. Taking into account that our *passive* approach that relies on sets-of-words is more efficient than the approach used in *best*—which makes use of external knowledge bases—, we conclude that *passive* is a suitable starting point to use for active learning of EF.

4.2 Margin sampling vs. random sampling

We compare the effectiveness of two candidate selection methods for active learning, random and margin sampling. We also compare our active learning approach to EF against the state-of-the-art in the EF task and against our *passive* learning baseline.



(a) Accuracy (b) $F_1(R,S)$

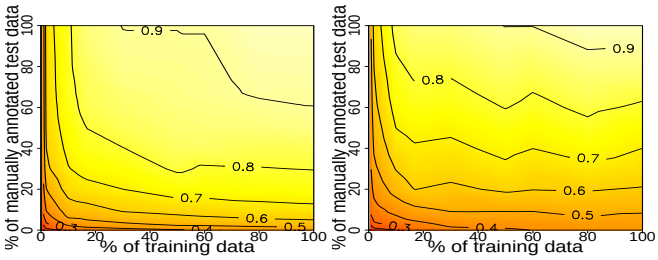
Figure 1: Accuracy (1a) and $\mathbf{F_1(R,S)}$ (1b) vs. percentage of manually annotated tweets $\mathbf{N_{test}}$.

Fig. 1 compares the *MS* and *RS* runs with the passive runs in terms of accuracy (1a) and $F_1(R,S)$ (1b). *MS* outperforms *passive* and *best* after inspecting only 2% of the test data (which, on average, corresponds to 30 tweets per entity), obtaining a $F_1(R,S)$-score of 0.52 (vs. 0.49). Using $N_{\text{test}} = 5\%$, *MS* significantly outperforms *best*, obtaining an $F_1(R,S)$-score of $0.63^{\blacktriangle}$. On the other hand, *RS* needs more feedback to be able to reach *best*. Using $N_{\text{test}} = 5\%$ it achieves an $F_1(R,S)$-score of 0.48, while using 10% of sampled tweets achieves a score of 0.52. The graphs also show *MS* outperforming *RS* consistently. Here, differences begin to be statistically significant from 3%–5%, with $F_1(R,S)$-scores of $0.57^{\triangle}$, $0.63^{\blacktriangle}$. Interestingly, while *RS* shows a linear behavior, *MS* starts with an exponential gain of effectiveness in terms of $F_1(R,S)$. In terms of $F_1(R,S)$, the effectiveness reached by *RS* after inspecting 10% of the test data can be achieved by *MS* considering only 2%. This amounts to an 80% reduction in cost.

In sum, our active learning approach requires small amounts of feedback to outperform state-of-the-art passive EF systems. Additionally, margin sampling significantly outperforms random sampling.

4.3 Initial training reduction

We examine to which degree active learning can reduce the cost of the initial training phase. Initializing any supervised approach—whether passive or active—to EF has a cost derived from annotating the initial training data. We look at different percentages of training data used to initialize the model.



(a) Margin Sampling (b) Random Sampling

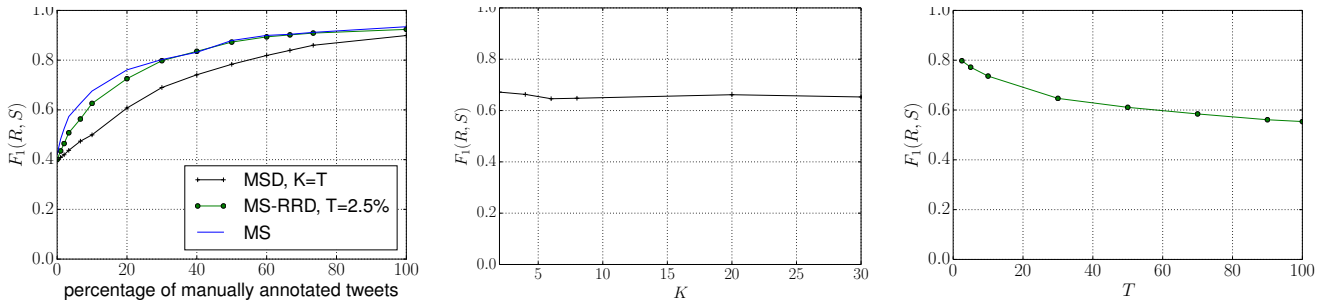
Figure 2: $\mathbf{F_1(R,S)}$-scores with different percentages of training data for the initial model (x-axis) and different percentages of test data for manually inspection during the active learning process (y-axis), using margin (2a) or random (2b) sampling. Red/dark and yellow/light correspond to lower and higher $\mathbf{F_1(R,S)}$ values, respectively.

Fig. 2 shows heat maps representing the evolution of $F_1(R,S)$ for different percentages of training data (x-axis) and sampled test data (y-axis), for *MS* (2a) and *RS* (2b). Red/dark and yellow/light correspond to lower and higher $F_1(R,S)$ values, respectively. *MS* needs less training data to obtain competitive $F_1(R,S)$-scores than *RS*. For instance, initializing the model with 10% and inspecting 10% of test data using *MS* achieves an $F_1(R,S)$-score of 0.55, while *RS* achieves only 0.44. Considering only 10% (i.e., 75 tweets) as the initial training set, the effectiveness of *best* can be reached after 100 tweets ($N_{\text{test}} < 7\%$) using *MS*. In terms of annotation cost, this corresponds to a 75% reduction.

In sum, the cost of training the initial model can be substantially reduced by using active learning, especially with margin sampling.

4.4 Dealing with outliers

We compare the effectiveness of favoring samples that are close to the margin and similar to other, unknown, samples in the candidate set. Fig. 3a compares margin*density sampling (*MSD*) for



(a) *MSD* and *MS-RRD* vs. *MS*. (b) K for *MSD*, considering $N_{\text{test}} = 28\%$. (c) T for *MS-RRD*, considering $N_{\text{test}} = 28\%$.

Figure 3: $\mathbf{F_1(R,S)}$-scores for the two approaches of density sampling, *MSD* and *MS-RRD*. Fig. 3a shows the development for different percentages of annotated tweets, Fig. 3b and Fig. 3c show the difference in $\mathbf{F_1(R,S)}$-scores for 28% of manually annotated tweets over different K and K, for *MSD* and *MS-RRD*, respectively.

$K = T$ and the best performing MSD reranking (*MS-RRD*, $T = 3$) in terms of $F_1(R,S)$ with margin sampling (*MS*). Both approaches to *MSD* performs worse than *MS*. Experiments that consider different numbers of instances K to compute the density (Fig. 3b) show that the performance is quite constant over K.

The performance of reranking based on density highly depends on the quantity of items of the initial ranking that are being considered (Fig. 3c). The performance drops significantly with the number of candidates used to rerank. While the performance of *MS-RRD* is closer to *MS* than *MSD*, vanilla *MS* performs significantly better than all density approaches.

One entity, however, stands out: *Chrysler* reaches an $F_1(R,S)$ of 1 using *MSD* and an $F_1(R,S)$ of 0 for normal margin sampling.[6] This entity has very clear topical clusters, about *winning prizes*, *chrysler 300*, *chrysler building*, and *union rights*. Manually classifying one of the elements in a cluster automatically classifies all the other elements. The other entity, *Bankia*, where *MSD* performs much better than *MS*, features a similar topical distribution.

We can conclude that unlike what was found in previous work, for EF density usually does not provide complementary information to margin sampling on this dataset.

5. CONCLUSION

We have examined the feasibility of using active learning for entity filtering on tweets. We have shown that much less annotation is needed when annotation is done on the fly, i.e., using active learning with 10% of the initial training set can lead to a 75% reduction of costs. We have contrasted several state-of-the-art sampling methods and have shown that margin sampling works best.

Future work should focus on entity filtering specific sampling algorithms and exploring the task in streaming scenarios. Since entity filtering is a daily task, an active learning streaming scenario simulates the ORM process even better.

Acknowledgments. This research was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 312827 (VOX-Pol), the Netherlands Organisation for Scientific Research (NWO) under project nrs 727.011.005, 612.001.116, HOR-11-10, 640.006.013, 612.066.930, CI-14-25, SH-322-15, Amsterdam Data Science, the Dutch national program COMMIT, the ESF Research Network Program ELIAS, the Elite Network Shifts project funded by the Royal Dutch Academy of Sciences (KNAW), the Netherlands eScience Center under project nr 027.012.105, the Yahoo! Faculty Research and Engagement Program, the Microsoft Research PhD program, and the HPC Fund.

[6]Note that $F_1(R,S)$ tends to zero when the system displays a non-informative behavior [4, 14].

6. REFERENCES

[1] E. Amigó, J. Artiles, J. Gonzalo, D. Spina, B. Liu, and A. Corujo. WePS-3 evaluation campaign: Overview of the online reputation management task. In *CLEF '10 (Online Working Notes/Labs/Workshop)*, 2010.

[2] E. Amigó, A. Corujo, J. Gonzalo, E. Meij, and M. de Rijke. Overview of RepLab 2012: Evaluating online reputation management systems. In *CLEF '12 (Online Working Notes/Labs/Workshop)*, 2012.

[3] E. Amigó, J. Carrillo de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín, E. Meij, M. de Rijke, and D. Spina. Overview of RepLab 2013: Evaluating online reputation monitoring systems. In *CLEF '13 (Online Working Notes/Labs/Workshop)*, pages 333–352, 2013.

[4] E. Amigó, J. Gonzalo, and F. Verdejo. A general evaluation measure for document organization tasks. In *SIGIR '13*, pages 643–652, 2013.

[5] J. Atserias, G. Attardi, M. Simi, and H. Zaragoza. Active learning for building a corpus of questions for parsing. In *LREC '10*, 2010.

[6] R. L. Figueroa, Q. Zeng-Treitler, L. H. Ngo, S. Goryachev, and E. P. Wiechmann. Active learning for clinical text classification: is it better than random sampling? *Journal of the American Medical Informatics Association*, 19(5):809–816, 2012.

[7] R. Hu. *Active Learning for Text Classification*. PhD thesis, Dublin Institute of Technology, 2011.

[8] M.-H. Peetz. *Time-Aware Online Reputation Analysis*. PhD thesis, University of Amsterdam, 2015.

[9] E. Pilkington. Unsold H&M clothes found in rubbish bags as homeless face winter chill. http://bit.ly/theguardian2010HM, January 2010.

[10] M. Sassano. An empirical study of active learning with support vector machines for Japanese word segmentation. In *ACL '02*, 2002.

[11] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[12] D. Spina. *Entity-Based Filtering and Topic Detection for Online Reputation Monitoring in Twitter*. PhD thesis, UNED, 2014.

[13] D. Spina, J. Carrillo de Albornoz, T. Martín, E. Amigó, J. Gonzalo, and F. Giner. UNED Online Reputation Monitoring Team at RepLab 2013. In *CLEF '13 (Online Working Notes/Labs/Workshop)*, 2013.

[14] D. Spina, J. Gonzalo, and E. Amigó. Discovering filter keywords for company name disambiguation in Twitter. *Expert Systems with Applications*, 40(12):4986–5003, 2013.

[15] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, Mar. 2002.

[16] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *ECIR '07*, 2007.

[17] J. Zhu, H. Wang, and B. Tsou. A density-based re-ranking technique for active learning for data annotations. In *ICCPOL '09*, 2009.