Towards an Active Learning System for Company Name Disambiguation in Microblog Streams*

Maria-Hendrike Peetz[1], Damiano Spina[2], Julio Gonzalo[2], Maarten de Rijke[1]
{M.H.Peetz,deRijke}@uva.nl,{damiano,julio}@lsi.uned.es

[1] ISLA, University of Amsterdam
[2] UNED NLP & IR Group

Abstract. In this paper we describe the collaborative participation of UvA & UNED at RepLab 2013. We propose an active learning approach for the filtering subtask, using features based on the detected semantics in the tweet (using Entity Linking with Wikipedia), as well as tweet-inherent features such as hashtags and usernames. The tweets manually inspected during the active learning process is at most 1% of the test data. While our baseline does not perform well, we can see that active learning does improve the results.

1 Introduction

With increasing volumes of social media data, social media monitoring and analysis is a vital part of the marketing strategy of businesses. Manual, and increasingly also automatic, extraction of topics, reputation, and trends around a brand allows analysts to understand and manage a brand's reputation. Twitter, in particular, has been used as such a proxy.

Efficient manual and automatic extraction requires filtering and disambiguation of tweets. Currently, for manual analysis, many non-relevant tweets have to be discarded. This has an impact on the costs of the analysis. For automatic analysis, non-relevant tweets might distort the results and decrease reliability.

* This research was partially supported by the Spanish Ministry of Education (FPU grant nr AP2009-0507), the Spanish Ministry of Science and Innovation (Holopedia Project, TIN2010-21128-C02), the Regional Government of Madrid and the ESF under MA2VICMR (S2009/TIC-1542) the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 640.004.802, 727.011.005, 612.001.116, HOR-11-10, the Center for Creation, Content and Technology (CCCT), the BILAND project funded by the CLARIN-nl program, 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 number 027.012.105, and the Yahoo! Faculty Research and Engagement Program.

Automatic reliable named-entity disambiguation on social media is therefore an active field of research. Typically, filtering systems are static (once trained, the model does not change) and fully automatic (there is no interaction with the analysts). However, both language and topics around an entity may change over time and the disambiguation performance is therefore likely to decay. Additionally, assuming the improvement in performance are worth it, the time to annotate a handful of tweets a day can easily be spent by analysts. We therefore propose an active learning approach to company name disambiguation. In particular, we analyze whether the annotation of a small number of tweets (at most 1% of the test data) per company improves significantly the results.

The paper is organized as follows. We continue with an introduction of the proposed approach in Section 2. We proceed with an explanation of the runs in the experimental setup (Section 3) and analyse the results in Section 4. We conclude in Section 5.

2 Proposed Approach

Our proposed approach is based on active learning, a semi-automatic machine learning process that interacts with the user for updating the classification model. It selects those instances that may maximize the classification performance with minimal effort. Figure 1 illustrates the pipeline of the system. First, the instances are represented as feature vectors. Second, the instances from the training dataset are used for building the initial classification model. Third, the test instances are automatically classified using the initial model. Fourth, the system guesses the candidate to be prompted to the user. This step is performed by uncertainty sampling: the instance with least certain as to be correctly classified is selected. Fifth, the user manually inspects the instance and labels it. The labeled instance is then considered to update the model. The active learning process is repeated until a termination condition is satisfied (e.g., the number n of iterations performed).

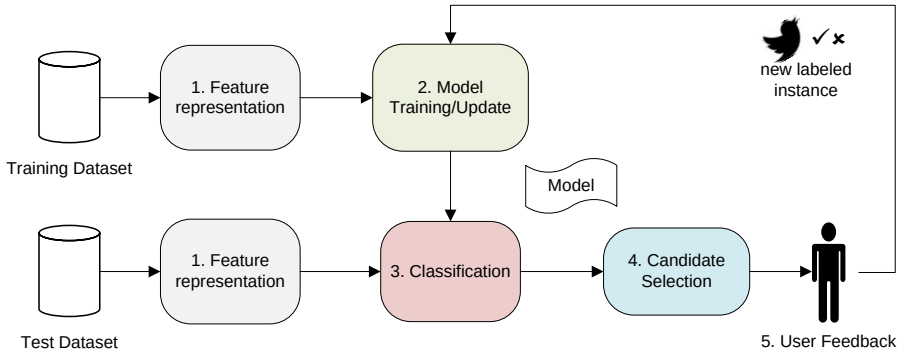

Fig. 1. Overview of the proposed active learning process.

2.1 Feature representation

We tested two feature representations:

Bag of Entities + Twitter metadata (`BoE`). First, an entity linking system is used to identify relevant Wikipedia articles to a given tweet. The COMMONNESS probability [5], based on the intra-Wikipedia hyperlinks, is used to select the most probable entity for each of the longest n-grams that were linked to Wikipedia articles from corpora related to the specific language. Spanish Wikipedia articles are finally translated to the corresponding English Wikipedia article by following the interlingual links, using the Wikimedia API.[3] Besides the entities linked to the tweet, special Twitter metadata—hashtags, usernames and author of the tweet—is also considered as features.

`BoE` + Bag of Words (`BoE+BoW`). This second feature representation simply adds the tokenized text of the tweet to the features in `BoE`.

Features are then weighted by two different weighting functions:[4]

Presence Each term is weighted with binary occurrence in the tweet: 1 if present, 0 otherwise.

Pseudo-document TF.IDF As in [10], we consider a pseudo-document D built from all the tweets given for an entity in the RepLab 2013 training/test dataset and a background corpus C containing all the D_i in the RepLab 2013 collection. Then, the weight w given to the term t is

$$w(t, D, C) = tf(t, D) \cdot \log \frac{N}{df(t)}$$

where $tf(t, D)$ denotes the term frequency of term t in pseudo-document D and $df(t)$ denotes the total number of pseudo-documents $D_i \in C$ in which the term t occurs at least once.

2.2 Learning model

We use Naïve Bayes[5] (NB) as a classifier and build an initial model. 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:

– Select the best candidate x from the test set T;

[3] http://www.mediawiki.org/wiki/API:Properties
[4] Each linked entity, hashtag, named user and author is treated as a term.
[5] http://nltk.org

– Annotate the candidate x;
– Update the model.

The annotations are selected from the test set. The test set depends on the experimental setup: in the cross-validation scenario, we could use the available annotations, while in the actual testing scenario, we annotated the candidates manually.

Candidate Selection. Following [8] the candidate selection can be based on *uncertainty sampling*, *margin sampling* (in particular for support vector machines [11]) or *entropy sampling*. Selecting candidates close to the margin is motivated by selecting candidates where the classification is less confident. Extending this motivation to NB, we choose to look at the probabilities $P(C_1|F)$ and $P(C_2|F_x)$) that a candidate x feature vector F_x generates the classes C_1 and C_2. The candidate x to be annotated from the test set T is:

$$x = \arg\min_{i \in T} |P(C_1 \mid F_x) - P(C_2 \mid F_x)|. \quad (1)$$

This candidate x is then being annotated and used to update the model.

Model updating Due to the speed of the training of the model, we decided to *retrain* NB with every new instance. We assigned all newly annotated instances a higher weight than the instances in the original training set.

3 Experimental Setup

In the following we describe how we used the training set to select the best feature groups. Based on this, we describe the runs we submitted. Unlike previous company name disambiguation datasets, such as the WePS-3 ORM dataset [1,12,9] and the RepLab 2012 dataset [2], the RepLab 2013 collection shares the same set of entities in training and test datasets. As reputation seems to be entity-specific [7], we build models *per entity*.

3.1 Training and parameter selection

Section 2.1 lists two feature representations: *bag of entities* (BoE) and *BoE + bag of words* (BoE+BoW). The BoW representation was generated by tokenizing the text of the tweet using a Twitter-specific tokenizer [6] and removing stopwords (using both Spanish and English stopword lists). Additionally, the feature values could be *presence* or *TF.IDF*.

We used 10 fold cross-validation (10CV) and iterative time-based splitting (ITS) [4] to evaluate the performance of the features. ITS ensures that classification of past tweets cannot be learnt from future tweets. Thus, we sort the tweets according to their time stamps and train the classifier on the first K

tweets and evaluate on the next K tweets. We then train on the first $2K$ tweets and evaluate on the next K tweets, etc. The total accuracy is the mean accuracy over all splits. We set $K = 10$. For both 10CV and ITS we used accuracy as our evaluation metric.

3.2 Submitted runs

The research questions that motivate our selection of submitted runs are:

RQ1 Does annotating a small number (15) of tweets from the test set improve the results?
RQ2 Do language-dependent models perform better?

We submitted four runs based the research questions, and two additional runs based on our observation that the data is imbalanced. We submitted two baseline runs without applying active learning: `UvA_UNED_filtering_1` and `UvA_UNED_filtering_2`. The first run is language-dependent, i.e., it uses a different NB model per language. The second run is language-independent, i.e., it uses a combined NB model for both languages. In order to answer **RQ1**, we submitted the two active learning runs `UvA_UNED_filtering_3` and `UvA_UNED_filtering_4`. The initial models are based on `UvA_UNED_filtering_1` and `UvA_UNED_filtering_2`, respectively. For the language-dependent case, we annotated 10 tweets per entity from the test set for English and 5 tweets per entity for Spanish. In the language-independent case, per entity, we annotated candidate 15 tweets from the test set. The runs `UvA_UNED_filtering_5` and `UvA_UNED_filtering_6` are `UvA_UNED_ filtering_3` and `UvA_UNED_filtering_4`, but when for an entity the training set related ratio[6] was < 0.1 or > 0.9, we used a *winner-takes-all* strategy. The *winner-takes-all* strategy classifies all the tweets as related or unrelated depending on which class is dominant in the training set.

Table 1 provides an overview over the runs. The official results are evaluated based on accuracy, reliability (R), sensitivity (S) and F(R,S), the F_1-measure of R and S [3].

Table 1. Overview over the runs submitted to RepLab 2013.

	language	
	dependent	independent
initial models	`UvA_UNED_filtering_1`	`UvA_UNED_filtering_2`
active learning (AL)	`UvA_UNED_filtering_3`	`UvA_UNED_filtering_4`
AL + winner-takes-all	`UvA_UNED_filtering_5`	`UvA_UNED_filtering_6`

[6] related ratio $= \frac{|\text{related}|}{|\text{related}|+|\text{unrelated}|}$.

4 Results

In the following we analyze the results on the training set in Section 4.1. We then elaborate on the official results in Section 4.2.

4.1 Preliminary experiments

Table 2 shows the accuracy for the two representation methods `BoW+BoE` and `BoE`. It shows results for coding the presence of the feature or the `TF.IDF` value. We can see that using the `TF.IDF` coding of the features works better than the `Presence` encoding. Additionally, we can see that using the `BoE` alone works better than the combination of both, `BoE+BoW`.

Table 2. Accuracy for the different representation methods tested on the initial model, not split by language.

Representation	10CV	ITS
`BoE+BoW - Presence`	0.42	0.46
`BoE+BoW - TF.IDF`	0.68	0.72
`BoE - Presence`	0.66	0.70
`BoE - TF.IDF`	0.76	0.79

Table 3 shows the accuracy for the two representation methods `BoW+BoE` and `BoE` for models built on split languages. It shows results for coding the presence of the feature or the TF.IDF value. Again, we can see that only using the BoE representation performs better in both test settings, 10CV and ITS. Additionally, we can see that here that `TF.IDF` outperforms performs better than `Presence`. We therefore choose to use the BoE representation coded with `TF.IDF` for our submitted runs.

Table 3. Accuracy for the different represenation methods tested on the initial model, split by language.

	English		Spanish	
Representation method	10CV	ITS	10CV	ITS
`BoE + BoW - Presence`	0.60	0.60	0.65	0.46
`BoE + BoW - TF.IDF`	0.71	0.72	0.71	0.52
`BoE - Presence`	0.67	0.69	0.74	0.54
`BoE - TF.IDF`	0.78	0.78	0.78	0.57

4.2 Submitted runs

Table 4 shows the results of our official runs with respect to accuracy, reliability (R), sensitivity (S), and F(R,S), the F_1 Measure of Reliability and Sensitivity. We can see that our baselines as well as the best performing system performs worse than a simple baseline. The provided baseline selects the class of an instance based on the class of the closest (using Jaccard similarity) instance in the training set.

We can, however, see some interesting improvements. For a start, active learning helps. We can see that the use of 1% annotation improves the results for all four metrics. Secondly, building a language-independent model performs better than building two language-dependent models per entity. Finally, we can see that the class imbalance also holds in the test set, as assigning the majority class for strongly skewed data performs much better than using active learning alone.

Table 4. Results of the official runs.

run id	accuracy	R	S	F(R,S)
`baseline`	0.8714	0.4902	0.3200	0.3255
`UvA_UNED_filtering_1`	0.2785	0.1635	0.1258	0.0730
`UvA_UNED_filtering_2`	0.2847	0.2050	0.1441	0.0928
`UvA_UNED_filtering_3`	0.5657	0.2040	0.2369	0.1449
`UvA_UNED_filtering_4`	0.6360	0.2386	0.2782	0.1857
`UvA_UNED_filtering_5`	0.7745	0.6486	0.1833	0.1737
`UvA_UNED_filtering_6`	0.8155	0.6780	0.2187	0.2083

5 Conclusions

We have presented an active learning approach to company name disambiguation in tweets. For this classification task, we found that active learning does indeed improve the results in terms of accuracy, reliability, and sensitivity. Since our initial models perform significantly lower than an instance-based learning baseline (probably due to bugs in the implementation), future work will include the analysis of the impact of active learning on stronger baselines.

References

1. Amigó, E., Artiles, J., Gonzalo, J., Spina, D., Liu, B., Corujo, A.: WePS-3 Evaluation Campaign: Overview of the Online Reputation Management Task. In: CLEF 2010 Labs and Workshops Notebook Papers (2010)
2. Amigó, E., Corujo, A., Gonzalo, J., Meij, E., de Rijke, M.: Overview of RepLab 2012: Evaluating Online Reputation Management Systems. In: CLEF 2012 Labs and Workshop Notebook Papers (2012)

3. Amigó, E., Gonzalo, J., Verdejo, F.: A General Evaluation Measure for Document Organization Tasks. In: Proceedings SIGIR 2013 (Jul.)
4. Bekkerman, R., Mccallum, A., Huang, G., Others: Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora. Center for Intelligent Information Retrieval, Technical Report IR (2004)
5. Meij, E., Weerkamp, W., de Rijke, M.: Adding semantics to microblog posts. In: Proceedings of the fifth ACM international conference on Web search and data mining (2012)
6. O'Connor, B., Krieger, M., Ahn, D.: Tweetmotif: Exploratory search and topic summarization for Twitter. Proceedings of ICWSM pp. 2–3 (2010)
7. Peetz, M.H., de Rijke, M., Schuth, A.: From sentiment to reputation. In: Forner, P., Karlgren, J., Womser-Hacker, C. (eds.) CLEF (Online Working Notes/Labs/Workshop) (2012)
8. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
9. Spina, D., Gonzalo, J., Amigó, E.: Discovering filter keywords for company name disambiguation in Twitter. Expert Systems with Applications 40(12), 4986–5003 (2013)
10. Spina, D., Meij, E., de Rijke, M., Oghina, A., Bui, M.T., Breuss, M.: Identifying entity aspects in microblog posts. In: SIGIR. pp. 1089–1090 (2012)
11. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. J. Mach. Learn. Res. 2, 45–66 (Mar 2002)
12. Tsagkias, M., Balog, K.: The University of Amsterdam at WePS3. In: CLEF 2010 Labs and Workshops Notebook Papers (2010)