Published in Proceedings of Conference and Labs of the Evaluation Forum (CLEF), Sheffield, United Kingdom, 15-18 september 2014, p.1491-1499 which should be cited to refer to this work

INSTANCE-BASED LEARNING FOR TWEET CATEGORIZATION IN CLEF REPLAB 2014

Julien Gobeill[1,2], Arnaud Gaudinat[1], Patrick Ruch[1,2]

[1], BiTeM group, HEG / HES-SO, University of Applied Sciences, 7 rte de Drize, 1227 Carouge, Switzerland
[2], SIBtex group, SIB Swiss Institute of Bioinformatics, 1 rue Michel-Servet, 1206 Genève, Switzerland
{julien.gobeill, arnaud.gaudinat, patrick.ruch}@hesge.ch

Abstract. BiTeM/SIBtex is a university research group with a strong background in Text Mining and Bibliomics, and a long tradition of participating in large evaluation campaigns. The CLEF RepLab 2014 Track was the occasion to integrate several local tools into a complete system for tweet monitoring and categorization based on instance-based learning. The algorithm we implemented was a *k* Nearest Neighbors. Dealing with the domain (automotive or banking) and the language (English or Spanish), the experiments showed that the categorizer was not affected by the choice of representation: even with all data merged into one single Knowledge Base (KB), the observed performances were close to those with dedicated KBs. Furthermore, English training data in addition to the sparse Spanish data were useful for Spanish categorization (+14% for accuracy for automotive, +26% for banking). Finally, our best official run was in top five. Yet, performances suffered from an overprediction of the most prevalent category, while we were not able to address this issue of unbalanced labels within the competition time. The algorithm showed the defects of its virtues: it was very robust, but not easy to improve. BiTeM/SIBtex tools for tweet monitoring are available within the DrugsListener Project page of the BiTeM website (http://bitem.hesge.ch/).

1 Introduction

BiTeM/SIBtex is a university research group with a strong background in Text Mining and Bibliomics, and a particular focus on clinical and biological data. Occasionally, the group is involved in studies with data from the intellectual property (granted patents) or the social media (tweets and reviews) domains. Finally, the group has a long tradition of participating in large evaluation campaigns, such as TREC, NTCIR or CLEF [1-4]. The CLEF RepLab 2014 Track was the occasion to integrate several local tools into a complete system, and to evaluate a simple and robust statistical approach for tweet classification in competition.

BiTeM/SIBtex only took part in the first task: Reputation Dimensions. The goal of the task was to perform text categorization on Twitter, i.e. to design a system able to assign a predefined category to a tweet. This category was one out of eight related to companies' reputations. All tweets dealt with entities from the automotive (20 entities) or the banking (11 entities) domain, and were in English (93%) or in Spanish (7%). For training and/or learning purposes, participants were provided with approximately 15,000 tweets labeled by human experts (the training set). Additionally, participants were allowed to use provided sets of tweets related to the mentioned companies for incorporating domain knowledge. Then, the systems had to predict the good categories for 32,000 unlabeled tweets (the test set).

In this task, the main difficulty was to efficiently preprocess the text, as standard Natural Language Processing strategies can fail to deal with the short, noisy, and strongly contextualised nature of the tweets. Another difficulty was to efficiently learn from unbalanced classes: indeed, the "Products & Services" category was assigned to 44% of the training tweets, versus only 1% for the "Innovation" category. Finally, this was a multilingual task, but the language distribution also was unbalanced, with less than 10% Spanish learning instances. We applied a simple and robust statistical approach in order to design our system, based on instance-based learning for categorization purposes. Instance-based learning is a kind of machine learning that compares unseen instances with labelled instances contained in a Knowledge Base (KB). The instance-based learning algorithm we chose to implement is *k* Nearest Neighbors (*k*-NN).

Three particular questions were investigated during this study:

- Q_1 : is it better to build one KB for each domain, or to merge automotive and banking into the same KB ?
- Q_2 : is it better to build one KB for each language, or to merge English and Spanish into the same KB ?
- Q_3 : as the labels are unbalanced, is it efficient to use weighting strategies for categorization ?

2 Methods

2.1 Overall architecture of the system

Figure 1 illustrates the overall architecture of our system. The workflow is divided into two steps: the training phase (offline), and the test phase (online). Three independent components act cooperatively to preprocess data (component 1), building the knowledge base (component 2) and classifying tweets (component 3).

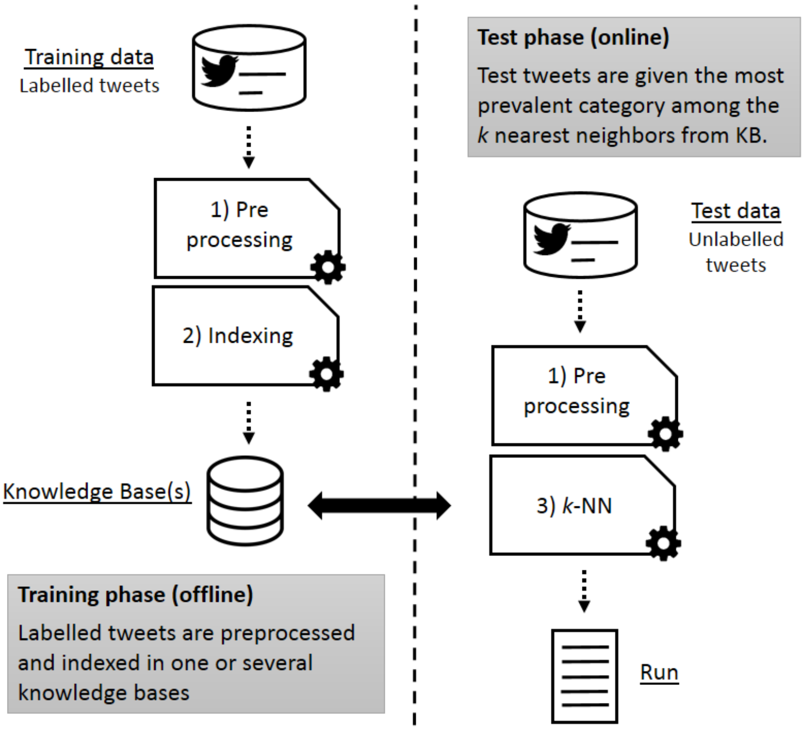

Figure 1: Overall architecture of the system

During the training phase, all tweets belonging to the training set were preprocessed by component 1. Component 1 is composed of several standard Natural Language Processing treatments, along with a language detector. Then, they were indexed in one or several indexes by component 2, in order to make the KB. Component 2 is an Information Retrieval platform, which builds indexes for related documents retrieval.

During the test phase, all tweets belonging to the test set also were preprocessed by component 1. Then, for a given test tweet, the component 3 (*k*-NN) exploited the KB in order to retrieve the most similar tweets seen in the training data, and to infer a predicted category. Official runs were computed with the whole test set.

2.2 Data

A training set of approximately 15,000 labelled tweets was provided by the organizers. There as an average of 511 tweets for an automotive entity, versus 485 for a banking entity. Table 1 shows the average distribution of each category for a given entity.

Table 1: Average distribution of reputation labels in training entities.

Category	*Automotive (20 entities)*	*Banking (11 entities)*
Citizenship	53	104
Governance	2	114
Innovation	8	4
Leadership	4	19
Performance	20	49
Products & Services	338	104
Undefined	75	66
Workplace	10	25
TOTAL	*511*	*485*

The first observation from Table 1 is that classes are unbalanced. For the automotive domain, 66% of training tweets deal with *Products & Services*, while only 0.8% deal with *Leadership*. The second observation is that distributions are different for the banking domain (e.g. only 21.4% for *Products & Services*). The distribution observed in test set (not reported) were consistent with those observed in the training set.

Here is a representative example of a tweet:

208844584137134080: Me and a sexy BMW M3 at last nights shoot <a href="http://t.co/ibW6sdXW" class="twitter-timeline-link" data-pre-embedded="true" dir="ltr" >pic.twitter.com/ibW6sdXW</a>

Tweets often contain metadata within tags, the most frequent being hyperlinks (<a>) and emphasis (<b>). Moreover, they often don't have proper punctuation.

2.3 Preprocessing

The goal of the component 1 was to preprocess the tweets in order to have proper and efficient instances to index (for the training phase) or search (for the test phase). For this purpose, a set of basic rules was applied. Tags were first discarded. Contents within an emphasis tag (<b>) were repeated in order to be overweighted. Contents within a hyperlink tag (<a>) also were repeated, and were preceded by the "HREF" mention.

For language detection purposes, we performed simple N-Gram-Based Text Categorization, based on the Cavnar and Trenkle works [5]. This approach aims at comparing n-grams frequency profiles in a given text, with profiles observed in large English and Spanish corpus. This simple approach is reported to have an accuracy in the range of 92% to 99%. N-grams profiles were taken from [6].

2.4 Indexing

The goal of the component 2 was to build one or several indexes from the training data, in order to obtain a related documents search engine. For this purpose, we used the Terrier platform [7]. We used default stemming, stop words and a Poisson weighting scheme (PL2).

Dealing with Q_1 and Q_2, we investigated several strategies and built several indexes:

- *all*: a unique index with all the training tweets;
- *cars*: an index with all tweets from the automotive domain;
- *banks*: an index with all tweets from the banking domain;
- *cars_en*: an index with all English tweets from the automotive domain;
- *banks_en*: an index with all English tweets from the banking domain;
- *cars_es*: an index with all Spanish tweets from the automotive domain;
- *banks_es*: an index with all Spanish tweets from the banking domain.

2.5 *k*-NN

The goal of the component 3 was to categorize tweets from the test set. For this purpose, we used a *k*-NN, a remarkably simple algorithm which assigns to a new text the categories that are the most prevalent among the *k* most similar tweets contained in the KB [8]. Similar tweets were retrieved thanks to component 2. Then, a score computer inferred the category from the *k* most similar instances, following this formula:

$$predcat = \underset{c \in \{c_1, c_2 \dots c_m\}}{\arg\max} \sum\nolimits_{x_i \in K} E(x_i, c) \times RSV(x_i)$$

where *predcat* is the predicted category for a test tweet, $c_1, c_2 \dots c_m$ are the possible categories, K is the set of the k nearest neighbors of the test tweet, $RSV(x_i)$ is the retrieval status value given by the component 2 (i.e. the similarity score) of the neighbor x_i, and $E(x_i,c)$ is 1 when x_i is of category c, 0 otherwise.

Dealing with Q_3, an additional score computing was tested for handling the issue of unbalanced labels when using a *k*-NN. Several studies were conducted for such an issue [9-12]. Solutions varies from rebalancing the training data to injecting weights in the score computing. The conclusions about how the *k*-NN really suffers from unbalanced data are not always concrete. Due to a lack of time, we investigated only one solution and chose to compute a weight associated to the local distribution of training tweets. The formula thus evolved into:

$$predcat = \underset{c \in \{c_1, c_2 \dots c_m\}}{\arg\max} \sum\nolimits_{x_i \in K} E(x_i, c) \times RSV(x_i) \times W(x_i, k + d, c)$$

where d is a parameter and $W(x_i,k+d,c)$ is the frequency of training tweets from category c in the set of the k nearest neighbors of x_i.

3 Results and Discussions

The Q_1, Q_2 and Q_3 issues were addressed with the training data, thanks to a ten-fold cross validation strategy.

3.1 *Q_1: is it better to build one KB for each domain, or to merge automotive and banking into the same KB ?*

First, we investigated Q_1, by exploiting the *all*, *cars*, and *banks* indexes. Both languages were merged into the same indexes. Figures 2a and 2b show the performances of the system for different values of k.

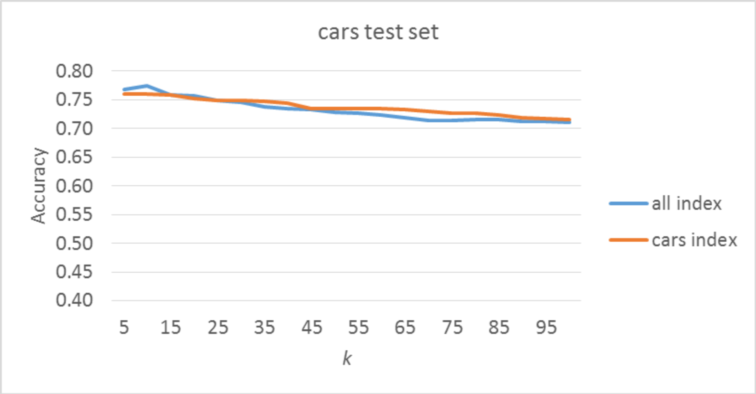

Figure 2a: Performances for the cars test set, using the all index (all training data merged) or the specific cars index (only cars training data), for different values of *k*.

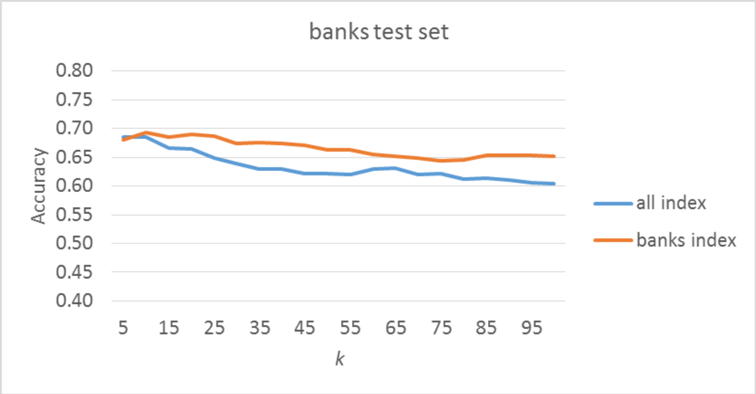

Figure 2b: Performances for the banks test set, using the *all* index (all training data merged) or the specific *banks* index (only banks training data), for different values of *k*.

Experiments showed that the optimal k for these data was around 10. They also showed that throughout the curves, it was better to use specific indexes (orange curves) versus a unique merged index (blue curves). Yet, the difference between best performances is not significant, with an accuracy of 0.69 for the *all* and the *banks* indexes for banks tweets (at *k*=10), and accuracies of 0.77 versus 0.76 for the *cars* index and the *all* index. We can say that, for categorizing tweets from a given domain, data from the other domain do not provide useful information, but do not degrade the optimal performances, thanks to the *k*-NN robustness.

3.2 *Q_2: is it better to build one KB for each language, or to merge English and Spanish into the same KB ?*

Second, we investigated Q_2, especially for the Spanish language that represented less than 7% of the training data. We exploited the *cars*, *banks, cars_es and banks_es* indexes. Figures 3a and 3b show the performances of the system for different values of *k*.

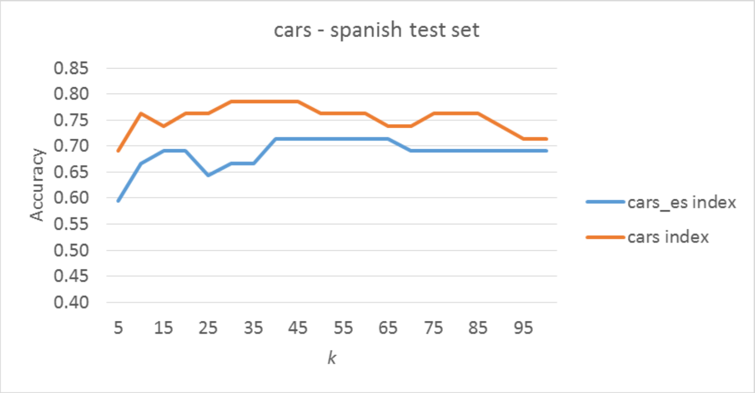

Figure 3a: Performances for the cars - Spanish test set, using the cars index (English and Spanish merged) or the specific cars - Spanish index (only Spanish data), for different values of *k*.

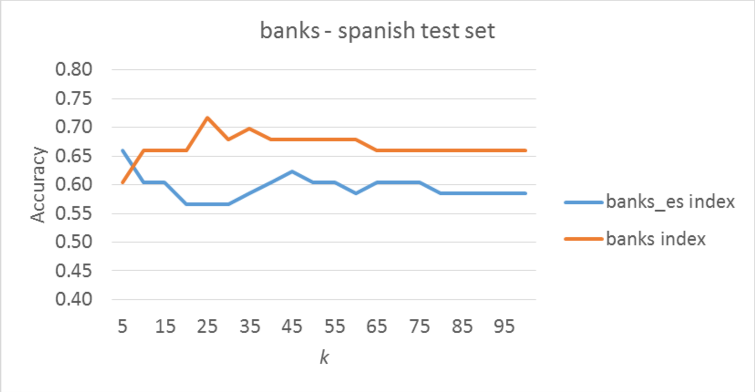

Figure 3b: Performances for the banks - Spanish test set, using the banks index (English and Spanish merged) or the specific banks - Spanish index (only Spanish data), for different values of *k*.

Experiments showed that the optimal *k* for Spanish data was around 30, significantly higher than the general case. This could be explained by the smaller set of Spanish instances. They also showed that it was better to use both languages indexes (orange curves) versus a Spanish-specific index (blue curves). We can say that, for categorizing tweets from Spanish, an additional amount of English data provides useful information and increases the top accuracy (from 0.69 to 0.79 for cars, from 0.57 to 0.72 for banks).

The same experiments with the English language (not reported) showed no significant differences between the merged and the English-specific indexes.

3.3 *Q_3: as the labels are unbalanced, is it efficient to use weighting strategies for categorization ?*

The last experiments aimed at tuning the *k*-NN for dealing with unbalanced labels. Results with different values of *d* (not reported) showed no improvements from the unweighted *k*-NN. Other strategies need to be investigated fur this issue.

3.4 Official submissions and results

We finally submitted two runs. For both runs, the automotive and banking training tweets were in separate Knowledge Bases. For run 1 (SIBtex_RD_1), we used a merged index for both languages. For run 2 (SIBtex_RD_2), we used specific languages. The best accuracy in the competition was 0.731. SIBtex_RD_1 had an official accuracy of 0.707 and was ranked #4. SIBtex_RD_2 had an official accuracy of 0.704 and was ranked #6. Interestingly, performances were better with the test set.

Official statistics also showed that, in our run, the “Products & Services” category was overrepresented (68% instead of 49% in the gold standard). Although we failed to design an efficient strategy for dealing with unbalanced data, this distribution shows that our *k*-NN probably suffered from this issue.

4 Conclusion

We designed a complete system for tweet categorization according to predefined reputational categories. Dealing with the domain (automotive or banking) and the language (English or Spanish), we explored a range of representations and wanted to know if it was better to use separate or merged Knowledge Bases. The experiments showed that the *k*-NN was not very affected by the kind of representations: even with all data merged into one single KB, the observed performances are close to those observed with dedicated KB. Moreover, English training data were useful for Spanish categorization (+14% for accuracy for automotive, +26% for banking). Yet, the unbalanced labels make the k-NN to predict the most prevalent category (“Products & Services”) more often than necessary (68% instead of 49%); this issue needs to be investigated in future works. The *k*-NN showed the defects of its virtues: it was robust, but not easy to improve. BiTeM/SIBtex tools for tweet monitoring are available within the DrugsListener Project page of the BiTeM website [13].

5 References

1. Gobeill J., Teodoro D., Pasche E. and Ruch P., Report on the trec 2009 experiments: Chemical IR track. the Eighteenth Text REtrieval Conference (TREC-18), 2009.
2. Gobeill J., Pasche E., Teodoro D. and Ruch P., Simple Pre and Post Processing Strategies for Patent Searching in CLEF Intellectual Property Track, 2009.
3. Teodoro D., Gobeill J., Pasche E., Ruch P., Vishnyakova D. and Lovis C., Automatic IPC encoding and novelty tracking for effective patent mining. In: The 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies, Tokyo, Japan, pp 309-317, 2010.
4. Vishnyakova D., Pasche E., Ruch P., Selection of relevant articles for curation for the Comparative Toxicogenomic Database. BioCreative Workshop [Internet], pp 31-38, 2012.
5. Cavnar W. and Trenkle J., N-gram-based Text Categorization. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994.
6. http://practicalcryptography.com/
7. Ounis I., Amati G., Plachouras V., He B., Macdonald C. and Lioma C., Terrier: A High Performance and Scalable Information Retrieval Platform. In Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval, 2006.
8. Manning C. and Schütze H., Foundations of Statistical Natural Language Processing. Cambridge: MIT Press, 1999.
9. Tan S., Neighbor-weighted K-nearest neighbor for unbalanced text corpus. Expert Syst. Appl. 28 (4), 2005.
10. Yang Y., An Evaluation of Statistical Approaches to Text Categorization. Inf. Retr. 1, pp 69-90, 1999.
11. Yang Y. and Liu X., A re-examination of text categorization methods. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99). ACM, New York, NY, USA, pp 42-49, 1999.
12. Qiao X. and Liu Y., Adaptive weighted learning for unbalanced multicategory classification. Biometrics, Mar;65(1), pp 159-68, 2008.
13. http://bitem.hesge.ch/