On the Use of PU Learning for Quality Flaw Prediction in Wikipedia

Notebook for PAN at CLEF 2012

Edgardo Ferretti[1], Donato Hernández Fusilier[2,4], Rafael Guzmán Cabrera[2],
Manuel Montes y Gómez[3], Marcelo Errecalde[1], and Paolo Rosso[4]

[1] Departamento de Informática, Universidad Nacional de San Luis (UNSL).
San Luis, Argentina.
{ferretti,merreca}@unsl.edu.ar
[2] División de Ingenierías, Campus Irapuato-Salamanca, Universidad de Guanajuato.
Salamanca, Guanajuato, Mexico.
{donato,guzmanc}@ugto.mx
[3] Departamento de Ciencias Computacionales, Instituto Nacional de Astrofísica,
Óptica y Electrónica (INAOE). Puebla, Mexico.
mmontesg@inaoep.mx
[4] NLE Lab - ELiRF, Universidad Politècnica de València (UPV). Spain.
prosso@dsic.upv.es

Abstract. In this article we describe a new approach to assess Quality Flaw Prediction in Wikipedia. The partially supervised method studied, called PU Learning, has been successfully applied in classifications tasks with traditional corpora like Reuters-21578 or 20-Newsgroups. To the best of our knowledge, this is the first time that it is applied in this domain. Throughout this paper, we describe how the original PU Learning approach was evaluated for assessing quality flaws and the modifications introduced to get a quality flaws predictor which obtained the best F1 scores in the task "Quality Flaw Prediction in Wikipedia" of the PAN challenge.

1 Introduction

Given the daily increase in the amount of data on the Web, machine-based assessment of Information Quality (IQ) is becoming a topic of enormous interest. This fact is rooted, among others, in the increasing popularity of user-generated Web content and the unavoidable divergence of the delivered content's quality [5]. In this respect, Wikipedia is a paradigmatic undertaking. This free-access encyclopedia generated from among the content contributed by millions of users, has this characteristic as main strength regarding its increased popularity. Nonetheless, this feature is probably, the main challenge that Wikipedia faces on how to systematically improve the quality of its articles.

According to our literature review, there are three main research lines related to IQ in Wikipedia, namely: (a) Featured articles identification [10, 12]; (b) Development of quality measurement metrics [11, 16]; and (c) Quality flaws detection [2–4]. It is clear that all the efforts made in improving IQ in Wikipedia should be enhanced, nevertheless, as indicated by Anderka *et al.* in [2, 3], a first step towards automatic quality assurance in Wikipedia is *detecting* quality flaws.

In [1], it has been presented the first complete breakdown of Wikipedia's quality flaw structure, which reveals the quality flaws that actually exist, the distribution of flaws in Wikipedia, and the extent of flawed content. It is important to notice that the majority of quality flaws are not caused due to malicious intentions but stem from edits by inexperienced authors.

In previous editions of the PAN challenge, assessing quality issues in Wikipedia has been addressed in the form of vandalism detection. Given the context above, in PAN@CLEF 2012,[5] the vandalism detection task has been generalized in focussing on the prediction of quality flaws in Wikipedia articles. In particular, the quality flaws to be predicted are the ten most frequent quality flaws of the English Wikipedia articles, namely: *Advert*, *Empty Section* (*Empty*), *No footnotes* (*No-foot*), *Notability* (*Notab*), *Original research* (*OR*), *Orphan* (*Orph*), *Primary sources* (*PS*), *Refimprove* (*Ref*), *Unreferenced* (*Unref*) and *Wikify* (*Wiki*). Besides, the task is formally defined as follows: "Given a set of Wikipedia articles that are tagged with a particular quality flaw, decide whether an untagged article suffers from this flaw". That is to say, that detection of text quality flaws is cast as a one-class classification, as proposed in [2].

In our view, the most notable proposals to quality flaw predictions in Wikipedia have been made by Anderka *et al.* [2–4]. In [3], it is reported on the exploratory analysis performed to target IQ flaws, and also a one-class classification technology for their identification is devised. The proposed method combines density estimation with class probability estimation. The experimental results show that certain flaws can be detected with a nearly perfect precision, while for others precision deteriorates significantly. In [2] it is performed a more in-depth experimental analysis, where two settings are considered in deriving outlier examples: an *optimistic setting* which uses featured articles[6] as outliers, and a *pessimistic setting* that uses a random sample of documents not tagged as containing the flaw. Finally, in [4], this idea is pushed further and previous work is extended with: (a) a comprehensive breakdown of prior work on quality assessment, (b) an in-depth discussion of the clean-up tag mining approach, (c) a description of the quality flaw model, and (d) a detailed analysis of the one-class problem.

As mentioned above, all the work done in literature with respect to quality flaw prediction in Wikipedia, has been carried out following supervised approaches. Despite the fact that very good results have been achieved in [2, 4] in the so-called *optimistic setting*, when using untagged articles as outliers (*pessimistic setting*) the effectiveness of flaws predictions notably decrease. In this way, motivated by [13], where several partially supervised learning techniques are discussed and it is also shown their good performances in Web mining applications, we decided to assess this task by means of a semi-supervised method. After considering several alternatives we came to the decision of following the

[5] `http://pan.webis.de/`

[6] `http://en.wikipedia.org/wiki/Wikipedia:Featured_article_criteria`.

approach proposed by Liu *et al.* [14, 15]. This approach, called *PU Learning* is explained next in Sect. 2. The key feature of this method is that it uses as input a small labelled set of the positive class to be predicted and a large unlabelled set to help learning. To the best of our knowledge, this is the first time that this method is used to predict information quality flaws in Wikipedia.

In Sect. 3, it is described in detail the research questions which guided the development of our proposal to participate in PAN@CLEF 2012. Besides, in this section it is also described a more intuitive rule-based approach to assess certain quality flaws. Then, Sect. 4 reports and discusses the results obtained in the competition with our PU Learning approach and with the rule-based approach as well. Finally, in Sect. 5 some general conclusions are drawn.

2 PU Learning

Text classification is an important problem which has numerous applications. As pointed out by Liu *et al.*:[7] "Although this classic model is important,[8] in practice one also encounters another problem. That is, one has a set of documents of a particular topic or class P (positive class), and is given a large set U of mixed (unlabelled) documents that contains documents from class P and also other types of documents (negative documents). One wants to classify the documents in U into documents from P and documents not from P. The key feature of this problem is that there is no labeled negative training data, which makes the traditional text classification techniques inapplicable. This problem is termed, partially supervised classification (PSC). We also call it PU-learning (Learning from Positive and Unlabelled examples)."

In particular, given its simplicity and robust performance we decided to implement the two-step strategy proposed in [14], which addresses the problem of building two-class classifiers with only positive and unlabelled examples, but no negative examples. This strategy is briefly described below and for extra details, the interested reader should refer to [14, 15].

Step 1: Identifying a set of reliable negative documents from the unlabelled set.
Step 2: Building a set of classifiers by iteratively applying a classification algorithm and then selecting a good classifier from the set.

Figure 1 depicts the above-mentioned two-step strategy when classifier in the second stage is applied only once. We describe this variant since it was the one implemented, and in Sect. 3.2 it is explained why it was chosen. As it can be observed in this figure, the first stage classifier is trained with an unbalanced training set composed by positive documents (P) and untagged documents (U). Then, this classifier is tested with the untagged documents used for training. From this test, all the documents predicted as negatives compose the set of reliable negatives (RNs). In turn, the RNs set together with the positive documents are used for training the second stage classifier. Finally, the model generated by the second classifier is the one used in the classification task.

[7] `http://www.cs.uic.edu/~liub/NSF/PSC-IIS-0307239.html`

[8] Here, "this" refers to the supervised approach.

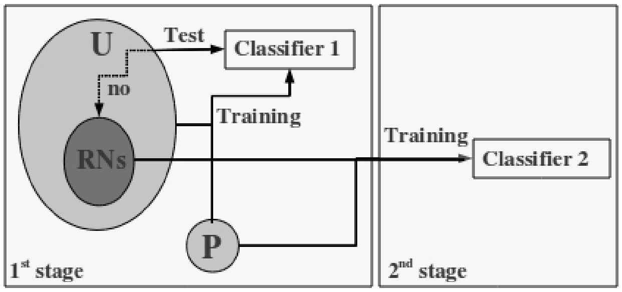

Fig. 1. Two-step strategy to PU Learning

3 Experimental Setting and Preliminary Results

It is well-known in Machine Learning research community that documents' representation is a key issue. However, given that the team expertise is stronger in the research field of algorithms, we decided to use features already proposed in the literature for modelling the documents and focussing our efforts in exploiting as much as possible the characteristics of the PU learning approach described in Sect. 2. There are four research questions which guided our experiments, namely:

1. What is the best classifier in each stage?
2. How to determine the sets of untagged documents for training the first stage classifier to improve its performance in selecting RNs?
3. After determining the RNs set, what documents should be used for training the second stage classifier?
4. Which parameters setting of the second classifier improves its performance?

These four questions are discussed next in below subsections. Regarding the documents' representation, we used as a guide the work performed in [4, 8], where they explore a significant number of quality features to assess the quality of Wikipedia articles. These features are detailed in the Sect. 3.1. Given the characteristics of some flaws like *Empty*, *No-foot*, *Ref* and *Unref*, there is no need to generate a complex document model to predict them. This is why, we also devised a simpler rule-based approach based on parsing the articles' wikitexts to find particular patterns indicating the presence of these flaws. This approach is briefly described in Sect. 3.6.

3.1 Documents Model

In [8], several features are conceptually grouped in three classes: Text Features (those extracted from articles' textual content), Review features (those extracted from articles' review history) and Network features (those extracted from the social network inherent to the collection). Similarly, in [4], a four dimension classification of features is devised. Our document model is composed by the

Table 1. List of Features Composing our Document Model

Text Features	Length: character count, information-to-noise ratio, sentence count, syllables count, one-syllable word count, word count; Structure: average sentence length, average word length, average section length, average subsection length, average subsubsection length, average sections nesting, average subsections nesting, category count, external link count, file count, heading count, image count, longest section length, longest subsection length, longest subsubsection length, mandatory sections count, section count, subsection count, subsubsection count, tables count, templates count, trivia sections count, passive sentences rate, citation count, reference sections count, shortest section length, shortest sentence length, shortest subsection length, shortest subsubsection length; Style: Complex word rate, Conjunction rate, Difficult word rate, Doubt word rate, Easy word rate, stop words rate, longest sentence length, long sentences rate, long words rate, average word syllables, one-syllable word rate, short sentences rate, Peacock words rate, prepositions rate, pronouns rate, questions rate, "To be" verb rate, Auxiliary verb rate, Weasel word rate, rate of sentences beginning with: article coordinating conjunction, interrogative pronoun, preposition, pronoun, subordinating conjunction; Readability: ARI, Bormuth, Coleman-Liau, Dale-Chall, Flesch, Gunning-Fog, Kincaid, Lix, Miyazaki, SMOG-Grading
Network Features	In-link count, Internal link count, Inter-language link count

features mentioned in Table 1, which are a subset of the ones used in [4]. The results reported in [8] show that textual features perform best and this is why almost all of our features belong to this category. It is worth noticing that all the features shown in Table 1 have been proposed by different authors [4, 6, 8, 12, 16] and for a better understanding they have been organized as suggested in [8].

3.2 What Classifier in Each Stage?

As mentioned above, in [14] a benchmark system is proposed where a comprehensive evaluation of sixteen combinations of classifiers for both steps is performed. From this study it is shown that Support Vector Machines (SVM) variants perform best as classifiers for the second step. Also it is corroborated that Naïve Bayes (NB) performs very well as first stage classifier. In this way, based on this evidence we decided to evaluate this combination first. Moreover, given the results reported in [17] where KNN is proposed as first stage classifier, we decided to study this technique as well. Consequently, several experiments were carried out using NB, SVM and KNN as first and second stage classifiers, respectively. These experiments involved different corpora created from the PAN training release. Similarly to the findings in [14], using NB and SVM as first and second stage classifiers, respectively, achieved very good results. Besides, NB + SVM also presented a very good trade-off between running times and good results. Thus, this combination was used in the remaining experimental setting.

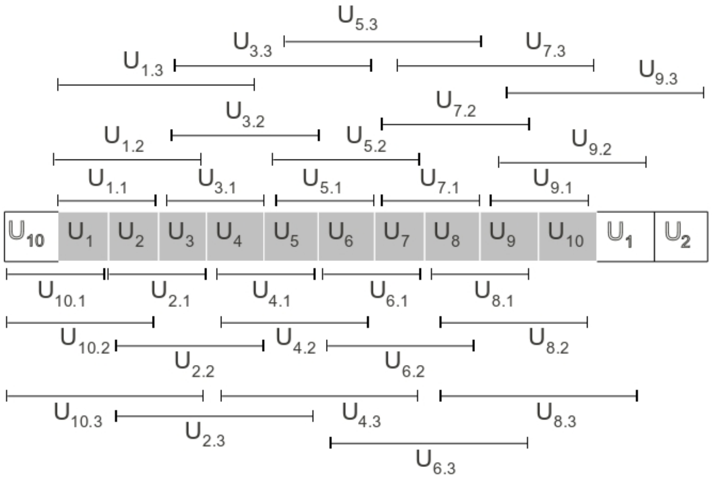

Fig. 2. Untagged Training Sets

3.3 Sampling Strategy of Untagged Documents

As indicated in Sect. 3.2, several corpora were built from the PAN training release. The main concern in building these corpora was studying how the sampling strategy of untagged documents ($\mathbf{U}$) could influence the results obtained by our approach. Instead of using a tenfold cross-validation approach as usual, splitting the documents in $\mathbf{U}$ by our own gave us the possibility of having much more control of the experimental setting, mainly on the issues related with determining the proportions of positive vs. untagged documents in the training sets.

To avoid a bias in how $U(\subseteq \mathbf{U})$ was determined, 40 different samples were selected to cover all the 50000 untagged documents in $\mathbf{U}$. Figure 2 shows how these 40 different samples were obtained. Originally, $\mathbf{U}$ was split in 10 sub-sets U_i such that $|U_i| = 5000$, for $i = 1 \ldots 10$. Then, subsets $U_{i.1}$, were built such that: $U_{i.1} = U_i + U_{(i \bmod 10)+1}$. Hence, $|U_{i.1}| = 10000$ for all $i = 1 \ldots 10$. Similarly, subsets $U_{i.2}$, were obtained as: $U_{i.2} = U_{i.1} + U_{((i+1) \bmod 10)+1}$. Thus, $|U_{i.2}| = 15000$ for all $i = 1 \ldots 10$. Finally, subsets $U_{i.3}$, were built as $U_{i.3} = U_{i.2} + U_{((i+2) \bmod 10)+1}$. In this way, for all $i = 1 \ldots 10$, $|U_{i.3}| = 20000$. The idea of building these untagged sets in an incremental way aims at analysing the effect of increasing the proportion of untagged documents versus positive documents in the training sets. Larger proportions up to $|U| = 50000$ were tried but no improvements in the results were achieved. Moreover, increasing the size of U also increases running times, so 20000 was set as the upper amount of untagged samples to be used.

Given that the number of positive sample documents for each flaw was highly unbalanced, for each flaw it was also determined the minimum amount of positive documents required to get the best results. For eight of the ten flaws, the number of positive documents in the training sets was set to 1000. Besides, several proportions for the respective test sets were analysed. From these experiments we decided to use for testing only 110 positive documents, since having more of them in average resulted in similar performance rates. Flaws *Advert* and *OR* contain 1109 and 507 documents, respectively. Hence, in order to have a unified experimental test setting, it was also set to 110 the number of positive documents to be used in the test sets of these flaws. Hence, the number of positive documents for training were 999 for *Advert* and 397 for *OR*, respectively.

In this way, for each flaw f it was fixed a positive set P_f which was combined with each of the 40 different subsets of **U** depicted in Fig. 2, thus yielding in 40 different training sets for the first stage classifier. As explained in Sect. 2, set P_f also comprise the positive sample of the training set of the second stage classifier. In the following section, the different approaches used to determine the negative training set of the second stage classifier, are explained.

3.4 Strategies for Selecting Reliable Negatives

Four strategies were used for selecting the reliable negative documents (RNs) to compose the training set of the second stage classifier, namely:

1. Selecting all RNs as negative set.
2. Selecting $|P_f|$ documents by random from RNs set.
3. Selecting the $|P_f|$ best RNs (those assigned the highest confidence prediction values by the first stage classifier).
4. Selecting the $|P_f|$ worst RNs (those assigned the lowest confidence prediction values by the first stage classifier).

Strategy 1 is the original one proposed in [14] and was the first one used in our experiments. Testing our approach with positive samples only, we realised that this strategy produces in average more false negatives (fn) than strategies $2-4$. Table 2 reports the average, median, minimum and maximum fn values for these strategies. Since the performance of the second stage classifier can only be measured by considering its recall values, the average recall values over the ten flaws are also presented. As it can be observed, the maximum number of fn for strategy 1 is 110, the actual number of positive samples in the test set. Besides, the average number of fn for strategy 1 is close to the maximum fn predictions for strategies 2 and 4, and it is higher than the maximum fn value for strategy 3. This shows that having a highly unbalanced training set for the second stage classifier affects the performance of our approach.

A statistical analysis (a non-parametric ANOVA) showed that the existing differences in the false negative rates of strategies 2 and 4, were significant when compared against strategies 1 and 3, respectively. For strategy 3, the differences with strategy 1 were found not significant. Similarly, taking into account the median recall values calculated on the ten flaws when trained with the 40 different training sets described in Sect. 3.3, the statistical analysis determined as significant the existing differences between strategies 2 and 4, against 1 and 3. Moreover, the mean rank differences found between strategies 2 vs. 4, and 1 vs. 3, were determined not significant, respectively. Table 3 presents the average recall values obtained per each flaw on the 40 training sets. As it can be observed,

Table 2. Recall and fn values for RNs selection strategies

Strategy	***fn* prediction rates**				**Recall**	
	Average	***Median***	***Minimum***	***Maximum***	***Average***	***Median***
1	22.17	3	0	110	0.80	0.97
2	4.48	1	0	26	0.96	0.99
3	4.00	4	0	10	0.96	0.96
4	4.17	1	0	30	0.96	0.99

Table 3. Average recall values per flaw

Strategy	**Flaws**									
	Advert	**Empty**	**No-foot**	**Notab**	**OR**	**Orphan**	**PS**	**Ref**	**Unref**	**Wiki**
1	0.58	0.98	0.57	0.99	0.30	1.00	0.74	0.61	0.99	0.97
2	0.90	0.99	0.86	0.99	1.00	1.00	0.90	0.99	0.99	0.98
3	0.95	0.98	0.94	0.99	0.97	0.99	0.95	0.96	0.97	0.95
4	0.90	0.99	0.89	0.99	1.00	1.00	0.89	0.99	0.99	0.99

strategy 1 for five out of the ten flaws performs very poorly. Furthermore, when considering the running times for each strategy, strategy 1 was found at least three times slower than the other ones. Based on this evidence, we decided to continue working with strategies $2 - 4$.

When compared against strategies 3 and 4, strategy 2 is conceptually the simplest one, since it just selects at random $|P_f|$ RNs documents to make a balanced training set for the second stage classifier. Conversely, strategy 3 selects those documents assigned the highest confidence prediction values by the first classifier, on the grounds that they are better candidates in representing the real negative documents' features. Finally, strategy 4 aims at selecting those documents that in spite of being predicted as negatives, are still quite similar to the positive ones. The underlying idea of this last strategy, is that selecting these documents could help to build a much more fine-grained borderline between both sets of documents. As shown in Table 2, strategies 2 and 4 perform best.

3.5 SVM: Which Parameters?

In Sect. 3.2, it was mentioned that using a SVM variant as second stage classifier reported the best results in [14] and in our experiments as well. Following the suggestions of Chang and Lin [7],[9] the authors of the SVM implementation we used, we tried all the different combinations of the parameters $\gamma \in \{2^{-15}, 2^{-13}, 2^{-11}, \ldots, 2^{1}, 2^{3}\}$ and $C \in \{2^{-5}, 2^{-3}, 2^{-1}, \ldots, 2^{13}, 2^{15}\}$ of the RBF kernel which is used by default in this software package. We found that combinations reporting the best results were those having a high penalty value (C) for the error term and very low γ values, which allow reproducing in a high-sensitive way decision boundaries. In particular, in our experiments with the training sets described in Sect. 3.3, $C = 2^{15}$ was found as the best penalty value

[9] `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`

Table 4. Best γ values per flaw

Advert	**Empty**	**No-foot**	**Notab**	**OR**	**Orphan**	**PS**	**Ref**	**Unref**	**Wiki**
2^{-7}	2^{-7}	2^{-5}	2^{-11}	2^{-9}	2^{-9}	2^{-5}	2^{-9}	2^{-9}	2^{-9}

for all the flaws, while the best γ values are indicated in Table 4. It is worth mentioning, that values presented in Tables 2 and 3 were obtained with the RBF kernel, accordingly set with the parameters values reported in Table 4.

During the training stage, we studied the performance of our algorithm using the default kernel, *i.e.*, RBF. When the PAN test set was released, the only clue we had about it, was that it was balanced. That is to say, that for all the flaws, there were the same amount of positives and untagged documents. In this way, approximately 50% of the documents was expected to be predicted as positives by our algorithm. When we ran the different variants of our algorithm, in average, for all the flaws, they predicted as positive nearly 75% of the test documents.

With the aim of improving the classifiers expected performance, the same experimental setting carried out for the training set was also run for the test set. Instead of studying the classifiers performance based on recall and fn measures, they were studied with respect to their prediction rates. For each document in the test sets, statistics were gathered considering if a particular classifier predicted it or not as positive. For the flaws *Advert*, *Empty*, *No-foot*, *OR* and *Ref*, most of the classifiers agree on their predictions, while for the remaining flaws the classifiers shown different predicting behaviours. As this phenomenon could be caused by an over-fitting in the models learned from the training sets, therefore, it was tried a more simple approach like a linear kernel instead of RBF. With this kernel, in average, the number of documents predicted as positive was 62.55%, a more balanced percentage than the one obtained for RBF.

Based on these studies on the PAN test set, the linear SVM was also studied as second stage classifier in our PU learning approach for the PAN training set. For this particular kernel, we used the default parameters provided by WEKA [9]. Table 5 reports recall and fn values for RNs selection strategies for the linear kernel. Conversely to the results presented in Table 2, strategy 3 is the best performing for this kernel. We also evaluated the RBF and linear SVM classifiers with positives plus untagged samples to reproduce the experimental conditions of the PAN test set. From these experiments we noticed that strategy 3 tends to predict as positives many untagged documents, while strategies 2 and 4 tend to maintained their positive predictions rates.

In this way, based on all the experiments performed with both kernels and also considering the prediction statistics gathered for each document for the PAN test set, we decided to use strategy 2 for RNs selection in nine out of the ten flaws. Flaw *Orph*, was the only one where strategy 4 was used. Regarding the kernel selection for the second stage classifier, the linear kernel was used in eight out of the ten flaws. Flaws *Advert* and *OR* were the only ones where the RBF kernel was used. In the Sect. 4 the results obtained with the PAN test set are presented.

Table 5. Recall and fn values for RNs selection strategies with linear kernel

Strategy	***fn* prediction rates**				**Recall**	
	Average	***Median***	***Minimum***	***Maximum***	***Average***	***Median***
2	21	21.5	4	49	0.81	0.80
3	6.20	6	0	20	0.94	0.94
4	20	21	1	44	0.82	0.81

3.6 Rule-based Approach

As mentioned above, for some flaws like *Empty*, *No-foot*, *Ref* and *Unref*, generating a complex document model to predict them is not necessary. In this way, according to our experimental study, a document is predicted as having the *Empty* flaw when one of the following three conditions hold: the number of empty sections is greater than zero; the number of subsections without content is greater than zero or when the pattern "< - - - - . . . - - - ->" is present. Similarly, the *No-foot* flaw is predicted when at least one of the following conditions hold: there are no external links ("==External link==" = 0); there are no in-text citations ("http" = 0) or when expression "ref" is found less than 80 times. Moreover, the *Ref* flaw is predicted when: the number of external references is less than 22; the regular expression ("ref >") is found less than 65 times or when reference section is empty. Finally, for *Unref* flaw three conditions are used: expression "ref" is found less than 45 times; the reference section does not exist or there are no in-text citations ("http" = 0).

In [4], a rule-based approach has also been proposed to predict flaws *Empty*, *Orphan* and *Unref*, in what they have called the "intensional modeling". Their results are very accurate. It is worth noticing that their rules are applied on particular features belonging to the document model also used by the supervised approach they work with. In our case, our rule-based approach works with a different document model than the one used by our PU Learning approach.

4 PAN Results

Throughout this paper, we have mainly described the PU Learning approach implemented to participate in the PAN challenge. Despite the fact that we competed with this approach, as indicated in Sect. 3.6, also a rule-based approach was developed to assess the prediction of some quality flaws. As it can be observed in Table 6, the results obtained with our rule-based approach are not very encouraging. Nonetheless, we believe that we have failed in capturing the gist of the wikitext patterns which characterize best these flaws, and as suggested in [4], a rule-based approach can be very useful in detecting these particular flaws. Regarding the results obtained with our PU Learning approach, they show a better performance. In fact, as shown in Table 6, we got an average F1 score of 0.81. Table 7 shows in row *np*, the amount of positive documents predicted per flaw to get these performance values in the challenge. Similarly, in row *tn*, the total amounts of documents composing the test sets, are shown.

Table 6. Official evaluation results

	PU Learning approach			Rule-based approach		
Flaw	***Precision***	***Recall***	***F1***	***Precision***	***Recall***	***F1***
advert	0.736133	0.929000	0.821397			
empty section	0.741546	0.921000	0.821588	0.538670	0.996000	0.699193
no footnotes	0.720446	0.969000	0.826439	0.506842	1.000000	0.672721
notability	0.739655	0.858000	0.794444			
original research	0.647462	0.930966	0.763754			
orphan	0.830365	0.979000	0.898577			
primary sources	0.716615	0.923000	0.806818			
refimprove	0.734848	0.970000	0.836207	0.503528	0.999000	0.669571
unreferenced	0.744731	0.954000	0.836475	0.510475	0.999000	0.675685
wikify	0.742195	0.737000	0.739589			
MEAN	**0.735400**	**0.917097**	**0.814529**	**0.514879**	**0.998500**	**0.679292**

Table 7. Number of documents predicted as positives per flaw

	Advert	**Empty**	**No-foot**	**Notab**	**OR**	**Orphan**	**PS**	**Ref**	**Unref**	**Wiki**
np	1262	1242	1345	1160	729	1179	1288	1320	1281	993
tn	2000	2000	2000	2000	1014	2000	2000	1998	2000	1998

5 Conclusions

The use of a partially supervised method to predict quality flaws in Wikipedia has proven to be effective. In this domain, our PU Learning approach which implements several strategies for selecting RNs has outperformed the original proposal which uses all the RNs found by the first stage classifier. Strategies 2 and 4 achieved the best results. We expected that strategy 4 would perform well. This is due to the fact that its underlying idea consists of building a fine-grained borderline between both classes by selecting those documents that in spite of being predicted as negatives, are still quite similar to the positive ones. Likewise, in our view, strategy 2 achieved also very good results since by choosing the RNs randomly it captures in a better way the implicit heterogeneity of the documents not containing the flaw.

Also, it has been described the exhaustive experimental setting carried out to set up as best as possible all the features of this approach, in order to participate in the PAN challenge, where our proposal obtained the best F1 scores in the task "Quality Flaw Prediction in Wikipedia". As future work, we think that exploring other different semi-supervised techniques is a promising direction to improve quality flaw predictions in this free-access encyclopedia available to the entire world.

Acknowledgements

Edgardo Ferretti and Marcelo Errecalde thank Universidad Nacional de San Luis (PROICO 30310). The collaboration of UNSL, INAOE and UPV has been funded by the European Commission as part of the WIQ-EI project (project no. 269180) within the FP7 People Programme. Manuel Montes is partially supported by CONACYT, No. 134186. The work of Paolo Rosso was carried out also in the framework of the MICINN Text-Enterprise (TIN2009-13391-C04-03) research project and the Microcluster VLC/Campus (International Campus of Excellence) on Multimodal Intelligent Systems.

References

1. Anderka, M., Stein, B.: A breakdown of quality flaws in Wikipedia. In: 2nd Joint WICOW/AIRWeb Workshop on Web quality. pp. 11–18. ACM (2012)
2. Anderka, M., Stein, B., Lipka, N.: Detection of text quality flaws as a one-class classification problem. In: 20th ACM International Conference on Information and Knowledge Management (CIKM'11). pp. 2313–2316. ACM (2011)
3. Anderka, M., Stein, B., Lipka, N.: Towards Automatic Quality Assurance in Wikipedia. In: 20th International Conference on World Wide Web. ACM (2011)
4. Anderka, M., Stein, B., Lipka, N.: Predicting Quality Flaws in User-generated Content: The Case of Wikipedia. In: 35rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM (2012)
5. Baeza-Yates, R.: User generated content: how good is it? In: 3rd Workshop on Information Credibility on the Web (WICOW'09). pp. 1–2. ACM (2009)
6. Blumenstock, J.E.: Size matters: word count as a measure of quality on Wikipedia. In: 17th Int'l Conference on World Wide Web. ACM (2008)
7. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
8. Dalip, D., Gonçalves, M., Cristo, M., Calado, P.: Automatic quality assessment of content created collaboratively by Web communities: a case study of Wikipedia. In: 9th ACM/IEEE-CS Joint Conference on Digital Libraries. ACM (2009)
9. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations 11(1) (2009)
10. Lex, E., Völske, M., Errecalde, M., Ferretti, E., Cagnina, L., Horn, C., Stein, B., Granitzer, M.: Measuring the quality of web content using factual information. In: 2nd joint WICOW/AIRWeb Workshop on Web quality. ACM (2012)
11. Lih, A.: Wikipedia as participatory journalism: reliable sources? Metrics for evaluating collaborative media as a news resource. In: 5th International Symposium on Online Journalism. pp. 16–17 (2004)
12. Lipka, N., Stein, B.: Identifying featured articles in Wikipedia: writing style matters. In: 19th international Conference on World Wide Web. ACM (2010)
13. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Data-Centric Systems and Applications, Springer-Verlag, 2nd edn. (2011)
14. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.: Building text classifiers using positive and unlabeled examples. In: 3rd IEEE Int'l Conference on Data Mining (2003)
15. Liu, B., Lee, W.S., Yu, P., Li, X.: Partially supervised classification of text documents. In: 19th International Conference on Machine Learning (2002)
16. Stvilia, B., Twidale, M., Smith, L., Gasser, L.: Assessing information quality of a community-based encyclopedia. In: 10th International Conference on Information Quality (ICIQ'05). pp. 442–454. MIT (2005)
17. Zhang, B., Zuo, W.: Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples. Journal of Computers 4(1), 94–101 (2009)