Vote/Veto Classification, Ensemble Clustering and Sequence Classification for Author Identification

Notebook for PAN at CLEF 2012

Roman Kern[1,2], Stefan Klampfl[2], and Mario Zechner[2]

[1] Institute of Knowledge Management
Graz University of Technology
rkern@tugraz.at
[2] Know-Center GmbH
Graz, Austria
{rkern,sklampfl,mzechner}@know-center.at

Abstract The Author Identification task for PAN 2012 consisted of three different sub-tasks: traditional authorship attribution, authorship clustering and sexual predator identification. We developed three machine learning approaches for these tasks. For the two authorship related tasks we created various sets of feature spaces, where individual differences in writing styles are assumed to surface in just a subset of these spaces. The challenge there was to combine these feature spaces to enable the machine learning algorithms to detect these differences across multiple feature spaces. In the case of authorship attribution we combined the results of multiple base classifiers by following a supervised vote/veto meta classifier approach. For the intrinsic plagiarism/authorship clustering subtask we used an unsupervised ensemble clustering approach in order to combine information from several feature spaces. In the sexual predator identification task we applied a supervised sequence classification approach to uncover temporal patterns within chat conversations by categorizing not only the offending messages, but also the reactions to these offending messages.

1 Introduction

In the following we provide a detailed description of our approaches to solve the three subtasks of the Author Identification track of PAN 2012. This lab report is structured as follows: In section 2 we present our supervised vote/veto classification approach to solving the traditional authorship attribution subtask. These meta classifiers collect information from various heterogeneous feature spaces, a method we extended in section 3 for the unsupervised authorship clustering task by employing an ensemble clustering approach. Finally, in section 4 we describe a sequence classification approach for identifying offending messages by potential sexual predators.

2 Vote/Veto Classification for supervised authorship attribution

The task of supervised authorship attribution is to assign a previously unseen text to an author. For this author there may already exist a set of reference text documents. In this

case the problem is labelled closed class. If the author of the text is possibly not in the set of known authors, the problem is labelled as open class. For the open class problem we simply added texts from different data sets to the training data set and labelled all new authors as "other".

For the PAN 2012 competition we re-applied our system from the previous year [5] with only minor modifications. This allowed to compare the two different data sets, instead of comparing algorithmic approaches. In 2011 the test corpora had consisted of emails and the task had been to identify the original sender of these mails. Due to specific characteristics found in such texts, our system contained features specifically engineered for such a scenario.

Meta-Classifiers & Feature Sets We decided to use techniques from the field of supervised machine learning as a base for the authorship attribution task. In order to utilise a classification algorithm one has to transform the input data into a representation suitable for such an algorithm. Therefore the input data needed to be transformed into features, organised within feature spaces. Thereby these features were further organised in feature sets, which combined multiple features into coherent sets of similar features.

We defined ten different feature spaces (for a more detailed description please refer to the original paper [5]): *Basic Statistics*, *Token Statistics*, *Grammar Statistics*, *Stop-Word Terms*, *Pronoun Terms*, *Slang Terms*, *Intro-Outro Terms*, *Bigram Terms*, *Unigram Terms*, and *Terms*. Some of these feature spaces encode statistical properties of the text, others are sensitive to the topic. Stop-words and pronouns are expected to serve a function within the sentence, but not to be specific to certain topics. Neither are slang terms, intro-outro terms, or grammar, which rather reflect the writing style of the author. On the other hand, terms, unigrams, and bigrams should be indicative for specific topics.

As most of these feature sets were not compatible with each other, the individual features could not simply be combined to build a single feature space. Therefore we developed a meta-classifier, which took the output of individual base classifiers to reach a final classification decision. Each of these base classifiers operated in a single feature space.

During the training phase the performance of each base classifiers was assessed using a ten-fold cross-validation approach. The precision and recall for each class was recorded, where each author in the training set was represented as a single class. If the precision in the test phase exceeded a given threshold t_p for a class, the base classifier became eligible to vote for this class. If the recall exceeded another threshold t_r for a class, the base classifier could veto against this class. In the training phase the meta-classifier was just responsible to record the individual performance of the base classifiers for each class.

In the classification phase, where a previously unseen text document needed to be assigned to one of the authors (or an author not being present in the training data set for the open-class problem), the role of the meta-classifier was to combine the results of the base-classifiers. One of the base classifiers was treated differently than the others, its output was taken directly without taking into account its performance in the training phase, and the individual probabilities for the classes were taken as initialisation. For all the others, their *a posteriori* classification results were taken into consideration, as well as their assessed training performance. If a base classifier $\mathfrak{C}$ assigned a probability

$p_i^{\mathfrak{C}}$ to a specific class i it was compared to another set of two thresholds. In the case that the base classifier might vote for this class and the probability exceeded the threshold p_p, the probability was multipled by w_p and added to the class probability. When the base classifier was eligible for a veto and the probability was smaller than p_r, then the product of $w_r(1 - p_i^{\mathfrak{C}})$ was deducted from the class probability. Thus the final score for each author was then the combination of the individual classification results of the base classifiers.

Configurations The modular nature of our system allowed us to assess the influence of the different feature spaces on the overall performance. We were especially interested on the influence of the content based feature spaces in relation to the pure statistical feature spaces. Generally, statistical features can be considered to be less dependent on the actual topic or domain of the text than content based features. Thus features like terms will rather allow to detect a change in topic instead of actually identifying individual writing styles.

We defined three different configurations for our system, each of them is a combination of base-classifiers:

terms In this configuration all feature spaces were combined and the results of the terms feature space was an initialisation for the author scores. One would expect that this configuration should work best in cases where changes in authorship are directly coupled to changes in topic and each author used different content words.

stylo The second configuration was a combination of statical features and term features, which should carry little semantics. All three statistical feature spaces were used in combination with the stop-word and the pronoun feature spaces. The token statistics feature space was used as initialisation for the author scores.

stats The final configuration consisted only of the statistical based feature spaces, again using the token statistics for initialisation. This configuration was expected to provide the worst performance in cases where terms are indicative for authorship. In cases where different authors produce texts which are topically related this configuration should not be affected.

Comparison of the Data Sets To compare the data sets from the PAN 2011 with the PAN 2012 workshops we conducted a feature analysis on the individual base classifiers. The list of features for each feature space was ranked according to their information gain. The results are shown in Table 1.

In the case of the basic statistics feature space the differences between the two data sets are obvious. For the authorship of emails the layout appears to have played an important role, while for the PAN 2012 data set the length of the text was the most relevant factor. For the token statistics feature space the most discriminative features appear to be more similar. A closer look reveals that the histogram of token lengths between the two data sets appear to be different, at least in regard to their ability to distinguish between individual authors. The grammar statistics between the two data set varied by great degree, especially with regard to the depth of the sentence parse tree, which had been a good indicator for the PAN 2011 data set, but only ranked as the 11th most discriminant feature for the PAN 2012 data set.

Table 1. Comparison of the most discriminative features for the basic statistics feature space, the token statistics feature space, and the grammar statistics feature space of the PAN 2011 and PAN 2012 data set.

basic statistics:

	PAN 2011	PAN 2012
1	Paragraph to lines ratio	Number of characters
2	Text to lines ratio	Number of words
3	Number of lines	Number of lines
4	Empty lines ratio	Number of stop-words
5	Number of paragraphs	Number of tokens

token statistics:

	PAN 2011	PAN 2012
1	Likelihood of proper nouns	Number of tokens
2	Number of tokens	Likelihood of proper nouns
3	Average token length	Average verb length
4	Likelihood of infrequent word groups	Average token length
5	Likelihood of tokens of length 9	Likelihood of pronouns

grammar statistics:

	PAN 2011	PAN 2012
1	Number of phrases per sentence	Likelihood of dependency type poss
2	Average depth of the sentence parse tree	Likelihood of dependency type nsubj
3	Likelihood of the phrase FRAG	Likelihood of phrase NP
4	Likelihood of dependency type appos	Likelihood of dependency type conj
5	Number of phrase types	Likelihood of dependency type possessive

Table 2. Comparison of the performance of our system (classification accuracy in %) for the three configurations explained in the text.

Configuration	A	B	C	D	I	J
terms	83.3	50	62.5	35.3	64.3	50
stylo	33.3	40	25	11.8	35.8	37.5
stats	66.7	40	25	23.5	35.8	50

Performance on the Test Data Set The official numbers from the organisers allowed us to compare the performance of our system for the three configurations (see Table 2). The configuration which performed best is the one which incorporated the terms feature spaces. This can be seen as an indicator that the test set contained authors with disjunct topics, thus there was little overlap in their content words. The performance of the stylo configuration is expected to lie between the terms and stats configuration. The most plausible reason why this configuration performed worst is that for such a scenario the thresholds and weighting factors were not properly optimised. This can be seen as an indicator that for authorship attribution the domain does has an influence and should be taken into account when tuning an algorithm for optimal performance.

3 Ensemble clustering for unsupervised authorship identification

In the intrinsic plagiarism/authorship clustering task of the author identification track we were given a number of texts, each of which was written by at least two different authors. The task was to recover the author of individual paragraphs in an unsupervised manner, yielding a clustering or partition of the paragraphs. (For simplicity each text was segmented into paragraphs such that each paragraph was written by exactly one author.)

Table 3. Stylometric features and used in the authorship clustering task.

feature name	description
alpha-chars-ratio	the fraction of total characters in the paragraph which are letters
digit-chars-ratio	the fraction of total characters in the paragraph which are digits
upper-chars-ratio	the fraction of total characters in the paragraph which are upper-case
white-chars-ratio	the fraction of total characters in the paragraph which are whitespace characters
type-token-ratio	ratio between the size of the vocabulary (i.e., the number of *different* words) and the total number of words [13]
hapax-legomena	the number of words occurring once [13]
hapax-dislegomena	the number of words occurring twice [13]
yules-k	a vocabulary richness measure defined by Yule [13]
simpsons-d	a vocabulary richness measure defined by Simpson [13]
brunets-w	a vocabulary richness measure defined by Brunet [13]
sichels-s	a vocabluary richness measure defined by Sichel [13]
honores-h	a vocabulary richness measure defined by Honore [13]
average-word-length	average length of words in characters
average-sentence-char-length	average length of sentences in characters
average-sentence-word-length	avarage length of sentences in words

To solve these unsupervised authorship attribution problems we followed the hypothesis that individual differences in the writing style of different authors may emerge only in non-trivial combinations of heterogeneous features. We therefore extended the idea in the previous section of combining information from multiple feature spaces to the more challenging unsupervised case, and employed an ensemble clustering approach. Ensemble clustering (also known as consensus clustering or clustering aggregation) deals with the problem of finding a single clustering that agrees as much as possible with a given set of input partitions of the same data, see, e.g., [11,4]. For the problem at hand, we found an ensemble clustering approach a suitable choice, since it is able to collect information spread over very heterogeneous feature spaces, whose features are not directly comparable.

Feature spaces In order to extract valuable information from a given input text, we preprocessed the raw paragraphs with an NLP pipeline that performed POS tagging and token normalization. In the next step we extracted features from these individual annotated paragraphs of the documents, yielding one instance per paragraph in each feature space. In particular, the feature spaces we considered were as follows:

1. frequencies of individual characters (a-z), including digits (0-9) and punctuations (e.g., ",", ".", "?", etc.),
2. frequencies of character bigrams within tokens (e.g., the normalized token "word" consists of the bigrams "wo", "or", "rd"),
3. frequencies of character trigrams within tokens (e.g., "wor", "ord"),
4. frequencies of stopwords and pronouns (e.g., "they", "for", "until"),
5. frequencies of stem-suffices, i.e., endings of words that were removed in a stemming procedure [6] (e.g., "ible", "ized", "ness"),
6. a variety of stylometric features, as in Table 3,
7. a number of basic statistical features, as in [5] and in the supervised authorship attribution subtask desribed in section 2.

These feature spaces were chosen to reflect the style of the author, rather than the topic, which typically does not change within a plagiarized document. Hence we did

Table 4. Relative weighting of the individual feature spaces obtained by an exhaustive search in the space $\{1, 2, 3, 4, 5\}^7$.

feature space	characters	bigrams	trigrams	stopwords	stem suffix	stylometry	basic stats
weight	5	5	4	5	2	4	3

not consider the frequency of unigrams or other terms carrying semantic information as features. Similar features have been used in previous studies on authorship attribution and intrinsic plagiarism detection, see, e.g., [15,9,14,10]. While each feature space grouped together similar features, the feature spaces themselves were quite different from each other. Some feature spaces measured frequencies of characters or character groups, others measured statistical properties. Since many of these features are not directly comparable, we used an ensemble clustering approach to combine information distributed over these feature spaces.

Ensemble clustering For each of the feature spaces 1-7 we used the standard k-means clustering with k-means++ seed selection [2] algorithm to obtain individual clusterings. Depending on the feature space we also used different similarity functions to measure the distance between two instances/paragraphs: for the frequency based feature spaces 1-5 we employed cosine similarity and for the statistical feature spaces 6 and 7 we chose the standard Euclidean distance. In the latter case we additionally scaled individual features to zero mean and unit variance before feeding them into the clustering algorithm. In order to merge these individual clusterings obtained in each feature space into a final single clustering we then used the median partition approach presented in [12]. This approach performs another k-means clustering in a meta-space spanned by the individual clusterings, while not using any information from the underlying feature spaces. In this meta-space, two instances have a large distance to each other if they are mostly assigned to different clusters by the individual partitions, and a small distance if the majority of individual clusterings puts them into the same cluster.

This clustering aggregation method also allowed the relative weighting of the individual clusterings, enabling us to enforce feature spaces with "good" clusterings and weaken the influence of feature spaces where no discriminative clusterings with respect to the author could be obtained. In order to determine the best weights we performed an exhaustive search: we varied each of the 7 weights from a minimum value of 1 to a maximum value of 5 in steps of size 1. The performance measure we optimized was the average classification accuracy over 10 trial runs of this clustering pipeline on labelled training data for which we used documents from the traditional authorship training corpus (see next subsection for details). We arrived at the weights shown in Table 4.

Results The training corpus for the intrinsic plagiarism/authorship attribution subtask consisted of just two files: the first file for the mixed task consisted of 6 paragraphs where even and odd paragraphs belonged to two different authors, and the second file for the intrusive task spanned 17 paragraphs where most paragraphs were written by one author, except for two particular consecutive paragraphs belonging to a different

(intrusive) author. We found these provided datasets too limited for a reliable evaluation of our methods, and furthermore they were also quite unrealistic for a plagiarism scenario: the combined original documents varied considerably in topic and type (e.g., political theory vs children's fiction). For the development and evaluation of our ensemble clustering approach we therefore used different datasets. More precisely, we generated artificial input data by randomly mixing all paragraphs from two prespecified authors from the traditional authorship attribution dataset with 8 authors ("Problem C").

Table 5 shows the results of our algorithm on these artificial data. After clustering with $k = 2$ we assigned the found clusters to the known labels (authors) by minimizing the total number of misclassifications. This assignment problem was solved by the well-known Hungarian Algorithm, where the cost of labelling a cluster C_i with a label L_j was the number of samples of class L_j outside cluster C_i [1]. We then interpreted clustering performance as classification accuracy, i.e., the percentage of correctly classified instances. The number of instances/paragraphs for each author pair ranged from 328 to 830, and for particular pairs performance values up to 80% were achieved. The average performance over all datasets was 67.15%; this average performance measure was also the optimization objective for the weights in Table 4. These accuracy values might seem not particularly high, but one has to keep in mind that many paragraphs consisted of just a few words, e.g., in a dialogue, and are therefore hard to separate. Table 5 also compares the performance values for selected author pairs obtained by single clusterings in individual feature spaces with the performance of the combined clustering. It can be seen that the combined performance is greater than any performance obtained in a single feature space, demonstrating the power of the clustering ensemble approach. For the sake of completeness we also report the performance on the provided training files in Table 6. In the mixed task there was one misclassified paragraph, whereas in the intrusive task the algorithm correctly recalled both paragraphs from the other author, but incorrectly identified four other paragraphs from the main author.

We would also like to report the performance of our ensemble clustering method on the provided test corpus. The test corpus consisted of 3 texts of 30 paragraphs each for the mixed task, and 4 texts of 20 paragraphs each for the intrinsic task. In the mixed task each text was written by 2-4 authors; however, this twist in the final challenge assignment violated the original problem description, which had asked to return exactly two clusters. We therefore had limited time to develop methods for estimating also the correct number of clusters, and therefore submitted separate runs with $k = 2$, $k = 3$, and $k = 4$. Among the approaches we tried was to optimize a clustering objective across different k, such as the residual sum of squares or the stability across multiple runs with perturbed input. However, the results are typically biased towards small k, and due to the high-dimensionality of the feature spaces the results were numerically not very stable. As far as the intrusive task is concerned, we did not employ an additional method for finding a consecutive range of paragraphs by the intrusive author, we only returned the result of the clustering for three different runs.

According to the results published on the PAN website we achieved performances of 70%, 70%, and 65.56% for the runs in the mixed task with $k = 2$, $k = 3$, and $k = 4$, respectively. Among the 14 runs submitted, these runs ranked 6th and 11th. The three runs of the intrusive task yielded performances of 73.25% and 66.25% (twice),

Table 5. Performance evaluation of our ensemble clustering approach on artificial training data. We selected two authors from the traditional authorship attribution dataset with 8 authors ("Problem C") and reported clustering performance as classification accuracy. The shown accuracy is the average over 10 runs; the numbers in brackets denote the number of total instances/paragraphs for the particular author pair. For the bold entries the performances obtained in individual feature spaces are expanded below. These numbers demonstrate that combining feature spaces achieves a better performance than any clustering in a single feature space alone.

authors	**perf.**	**authors**	**perf.**	**authors**	**perf.**	**authors**	**perf.**
A vs B (328)	**66.10%**	B vs C (576)	64.49%	C vs E (814)	75.27%	D vs H (599)	68.15%
A vs C (624)	67.60%	B vs D (534)	69.16%	C vs F (774)	63.48%	**E vs F** (716)	**78.44%**
A vs D (582)	69.80%	B vs E (518)	66.94%	C vs G (729)	67.93%	E vs G (671)	62.18%
A vs E (566)	58.53%	B vs F (478)	62.11%	C vs H (641)	69.04%	E vs H (583)	61.19%
A vs F (526)	69.18%	B vs G (433)	58.24%	D vs E (772)	82.19%	F vs G (631)	66.66%
A vs G (481)	63.61%	B vs H (375)	59.74%	D vs F (732)	80.53%	F vs H (543)	66.35%
A vs H (393)	58.57%	**C vs D** (830)	**80.34%**	D vs G (687)	65.48%	G vs H (498)	58.80%

feature space	**A vs B**	**C vs D**	**E vs F**
1. characters	51.52%	53.98%	61.87%
2. character bigrams	50.91%	54.46%	56.70%
3. character trigrams	50.91%	51.33%	52.37%
4. stopwords & pronouns	62.20%	50.72%	72.91%
5. stem suffices	65.85%	63.01%	54.61%
6. stylometry	52.74%	59.76%	64.25%
7. basic statistics	57.01%	56.87%	65.22%
combined	**66.10%**	**80.34%**	**78.44%**

Table 6. Performance of our ensemble clustering approach on the provided training corpus. Shown are the confusion matrices for both the mixed and intrusive task: C1 and C2 denote the obtained clusters, and E1/E2 (F1/F2) are the labels/authors of the mixed (intrusive) task. The classification accuracy was 83.33% for the mixed task and 76.47% for the intrusive task.

mixed (83.33%)

	C1	**C2**	**Total**
E1	3	0	3
E2	1	2	3
Total	4	2	6

intrusive (76.47%)

	C1	**C2**	**Total**
F1	0	2	2
F2	11	4	15
Total	11	6	17

corresponding to rank 10 and 11. These are reasonable results given that we did not check for a consecutive range. If one only counts the best run per group, we rank 6th of 8 for the mixed task and 7th of 8 for the intrusive task.

4 Sequence classification for sexual predator identification

To identify sexual predators within chats we transformed the problem into a sequence classification task. Each chat was seen as a sequence of individual messages, and each message was represented as a single instance to be classified. In the first pass the classification of messages was done independently from the other messages present within a chat conversation. An important aspect of this classification is that the *a posteriori* probabilities need to be present together with alternative classification results, again

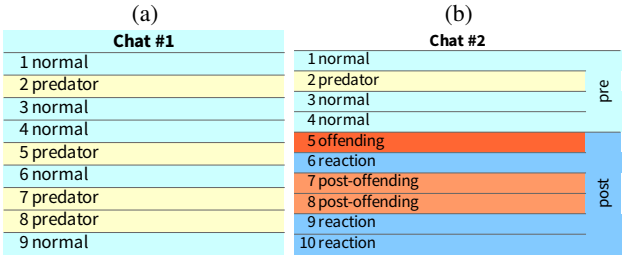

Figure 1. (a) Classes of the individual messages for a single chat conversation, which contains a sexual predator, but no message can be considered to be offending. (b) Messages and their class labels for a single chat conversation, which contains a sexual predator, but this time a message has been identified as offending. All messages which succeed this message are labelled differently.

coupled with their confidence values. In the second pass the classification results of adjacent messages were combined given by the examples seen in the training set.

Message Classes For the classification of individual messages within chats we defined five different classes.

normal An ordinary message not being written by a sexual predator. This should be the most common class of messages.

predator A message written by a predator, which is not seen as offending. We did not expect the classification algorithm to be reliably able to distinguish between normal messages and predator messages.

offending The first offending message from an predator within a chat conversation. After an offending message only the remaining two labels can occur.

reaction All messages written by normal chat participants after an offending message were labelled as reaction. This is based on the intuition that in some circumstances the reactions to an offending message is easier to detect than the original messages by the predator.

post-offending Any messages which follow an offending message by the predator are labelled as post-offending. This is motivated by the assumption that the behaviour of the predator might change once an offending message has been posted.

Two exemplary chats should illustrate the labelling sequence for the messages. In the first example a sexual predator takes part in a chat conversation, but behaves like an normal chat participant (see Figure 1a). Ideally the classification algorithm would still be able to detect the individual messages by the predator. In such a scenario it would be sufficient to identify only one of the predator messages, as this decision could then be propagated to the other messages from the same author as a post-processing step.

In the second example the chat does not only contain messages from a predator, but at least one of the messages can be attributed to be offending (see Figure 1b). Here the first of such messages is labelled as offending. All consecutive messages by the same author are marked as post-offending. The remaining messages in the chat are labelled as reaction. A single detected reaction would allow to attribute the other participant in a

chat (assuming there are only two participants) as a predator. Alternatively such a case could also be filed for later manual inspection.

Processing the data set Our proposed approach operated on the level of individual messages and therefore one of the prerequisites was a training corpus containing chats and messages labelled according to our message classes. Unfortunately the training data set supplied by the organisers only contained the information whether an author is a known sexual predator or not. In order to test our approach we had to develop a data set on our own.

Therefore we manually annotated the supplied training data set to follow our message classes. Due to time and resource constraints we were not able to annotate all the training data, but just a small subset of it. In the first iteration we created a development data set for the initial tests and creating the sequence labelling classification system.

To boost our productivity in annotating the data, we developed a web-based system to to ease the process of identifying offending messages. Unfortunately we did not have an expert on sexual predator behaviour at our disposal therefore a single member of our team spent a day to annotate a subset of the training data set. The result of this effort was used to train the classifiers for the submitted runs.

Classification Algorithm & Feature Sets There are a number of classification algorithms which do support the integration of sequence information. Examples for this type of algorithms are Hidden Markov Models and Conditional Random Fields. An alternative approach is to use a base classification algorithm, which does not provide native support of sequence information, but to postprocess the output of the base classifiers. We followed the second approach and used the Apache OpenNLP[3] project, which is an open-source implementation of the Maximum Entropy [3] classification algorithm in combination with a Beam Search strategy [8].

We also used the OpenNLP library to tokenise the message into separate tokens. We did not apply any stop-word processing or further analysis of the text. The tokens were just processed with the Double Metaphone algorithm [7]. Based on these tokens we built the features for the classification algorithms. For each message we added the individual tokens not only of the message itself, but also all the terms from all the other messages by the same author.

To incorporate the sequence information we added the classification result of the four preceding messages as features. We also added additional binary features: i) *isInitialAuthor*, if the author of the message initiated the chat, ii) *isLastAuthor*, if the author has posted the final message, iii) *isMostVerboseAuthor*, for the messages from the author with the most messages, iv) *isFewerAuthors*, if the message caused other authors to stop writing, and v) *hasTermFromPrevious*, if the messages shares at least a single term from the directly preceding message.

Once the classification result has been processed by OpenNLP we apply a set of simple post-processing rules: i) if there are multiple messages classified as offending, we only keep the label for the first one, all others are labelled as post-offending; ii) if there are messages classified as post-offending, but none as offending, we assign

[3] http://opennlp.apache.org/

Table 7. Sexual predator identification performance on the development data set and the test data set. The performance for the indirect identification of an offending message is (post-offending, reaction) is far better than the direct identification (offending).

development data set:

Class	Count	Precision	Recall
normal	3,117	0.955	0.995
predator	29	0.3	0.103
offending	52	0	0
post-offending	216	0.871	0.847
reaction	275	0.959	0.764
Identify predators	2	0.667	1

test data set:

Task	Precision	Recall
Identify predators	0.1476	0.6920
Identify predator line	0.0855	0.2074

the message that directly precedes the first post-offending message as offending; iii) if there is at least a single message classified as reaction and a single message classified as predator, the predator message which precedes the reaction is assigned the label offending.

Results on the Development and Test Data Sets We used the development data set to asses the initial performance of our system. Then we split the already small data set into a training part and into a test part, where we made sure that the test did not contain any authors which are used for training. We report the performance figures for our system based on the results on the test part of our development data set (see Table 7). The test set contained a total of three predators, where our classification system together with the postprocessing rule achieved a precision of 0.667 and a recall of 1. Due to the limited size of the development data set these figures cannot be regarded as conclusive. Nevertheless one can conclude from the measured performance that the indirect identification via the reaction and the post-offending behaviour appear to be easier to detect that the offending message itself.

To train our system for the official submitted runs, we assembled a training data set out of our manually assigned chat conversations. We took all conversations which contained a message identified as offending, added all chats from the same predator to the training set. We then added about the same amount of chats, which did not contain a participant marked as sexual predator (about 600 chats). The results are shown in Table 7.

5 Conclusions

We presented our systems for three tasks of the PAN 2012 challenge: We employed supervised vote/veto ensemble classification for the classical authorship attribution, unsupervised ensemble clustering for intrinsic plagiarism detection and sequence classification for the identification of sexual predators within chat logs. The source code for our systems can be used under the AGPL license and is available at https://knowminer.know-center.tugraz.at/svn/opensource/projects/pan2012.

Acknowledgments

This work has been funded by the European Commission as part of the TEAM IAPP project (grant no. 251514) within the FP7 People Programme (Marie Curie). The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Albalate, A., Suchindranath, A., Suendermann, D., Minker, W.: A semi-supervised cluster-and-label approach for utterance classification. In: Workshop Proceedings of the 6th International Conference on Intelligent Environments. pp. 61–70 (2010)
2. Arthur, D., Vassilvitskii, S.: k-means ++ : The Advantages of Careful Seeding. Proceedings of the eighteenth annual ACMSIAM symposium on Discrete algorithms 8(2006-13), 1027–1035 (2007)
3. Berger, A.L.: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics pp. 1–36 (1996)
4. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. ACM Transactions on Knowledge Discovery from Data 1(1), 4–es (2007)
5. Kern, R., Seifert, C., Zechner, M., Granitzer, M.: Vote/Veto Meta-Classifier for Authorship Identification. In: In 3nd International Competition on Plagiarism Detection (2011)
6. Muhr, M., Kern, R., Zechner, M., Granitzer, M.: External and Intrinsic Plagiarism Detection using a Cross-Lingual Retrieval and Segmentation System Lab Report for PAN at CLEF 2010. 2nd International Competition on Plagiarism Detection (2010)
7. Philips, L.: The double metaphone search algorithm. CC PLUS PLUS USERS JOURNAL 18(6), 38–43 (2000)
8. Ratnaparkhi, A.: Maximum Entropy Models for Natural Langual Ambiguity Resolution. Ph.D. thesis (1998)
9. Stamatatos, E.: Intrinsic plagiarism detection using character n-gram profiles. In: 3rd PAN Workshop: Uncovering Plagiarism, Authorship and Social Software Misuse. pp. 38–46 (2009)
10. Stein, B., Lipka, N., Prettenhofer, P.: Intrinsic plagiarism analysis. Language Resources and Evaluation 45(1), 63–82 (Jan 2010)
11. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3(3), 583–617 (2002)
12. Topchy, A., Jain, A.K., Punch, W.: Combining Multiple Weak Clusterings. Proceedings IEEE International Conference on Data Mining pp. 331–338 (2003)
13. Tweedie, F., Baayen, H.: How variable may a constant be? Measures of lexical richness in perspective. In: Computers and the Humanities. pp. 323–352 (1998)
14. Zechner, M., Muhr, M., Kern, R., Granitzer, M.: External and intrinsic plagiarism detection using vector space models. In: PAN09 - 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse and 1st International Competition on Plagiarism Detection. pp. 47–55 (2009)
15. Zheng, R., Li, J., Chen, H., Huang, Z.: A Framework for Authorship Identification of Online Messages: Writing-Style Features and Classification Techniques. Journal of the American Society for Information Science 57(3), 378–393 (2006)