Vote/Veto Meta-Classifier for Authorship Identification
Notebook for PAN at CLEF 2011

Roman Kern[1], Christin Seifert[1], Mario Zechner[2], and Michael Granitzer[1,2]

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

Abstract For the PAN 2011 authorship identification challenge we have developed a system based on a meta-classifier which selectively uses the results of multiple base classifiers. In addition we also performed feature engineering based on the given domain of e-mails. We present our system as well as results on the evaluation dataset. Our system performed second and third best in the authorship attribution task on the large data sets, and ranked middle for the small data set in the attribution task and in the verification task.

1 Introduction

The PAN 2011 challenge tackled the problem of authorship identification [5,12]. Two different task were given for this domain. The first task was an authorship attribution task. The goal of this task has been to identify the author of a previously unseen text from a set of candidate authors. The second task was authorship verification. The goal of authorship verification is to verify whether a text of unknown authorship was written by a specific (given) author.

We focused on the authorship attribution task and did not directly address the authorship verification problem setting. Instead, we applied the pipeline developed in the authorship attribution task to the verification task. The source code is open source and available for download[3].

2 Authorship Attribution System Overview

Our authorship identification systems consists of two main stages. In the first stage we analyze each author's documents and generate multiple sets of features for each document. In the second stage we train multiple classification models for each author based on these feature sets. We then combine these models to assign author names to unseen documents. The system was trained on the PAN authorship attribution development dataset. This dataset exclusively consists of e-mails, which allows us to employ some assumptions. The following list gives an overview of the stages and the sections describing the stages in detail:

[3] `https://knowminer.at/svn/opensource/projects/pan2011/trunk`

1. Document analysis and feature set generation
   (a) Preprocessing and feature generation, see section 2.1
   (b) Calculating additional statistics from external resources, see section 2.2
   (c) Weighting feature values, see section 2.3
   (d) Creating feature spaces, see section 2.4
2. Classifier training, see section 2.5

2.1 Preprocessing and Feature Generation

The task of the preprocessing step is to analyze the e-mail content (plain text) and to store the result alongside the document. Out of these so called annotations, the features will be generated in consecutive processing stages. The preprocessing pipeline consists of several components which are executed in a fixed sequence. The pipeline is individually applied to each document. Each annotation class will briefly be covered in the following section.

Text Line and Block Annotations: The first pre-processing of the text splits the content of each document into lines. Line endings are identified via the newline character. Multiple consecutive, non-empty lines are merged into text blocks. The text block annotations try to capture the layout of an e-mail message. All succeeding pre-processing steps operate on individual text-blocks.

Natural Language Annotations: We employ the open-source Java library OpenNLP[4] and the available maximum entropy models to split the text blocks into tokens and sentences. For each token we generated alternative representations. At first the characters of the token are normalized to **lower-case**. Next, the tokens are **stemmed** by applying the Snowball stemmer[5]. Finally all tokens are scanned for **stop-words**. Stop-words are detected by checking tokens against a set of predefined stop words from the Snowball stemmer project. Additionally we mark tokens with more than 50% non-letter characters as stop-words. Finally we annotate the **part-of-speech tag** for each token, again using the respective OpenNLP classifier and model for the English language.

Slang-Word Annotations Based on the intuition that e-mails may contain an Internet specific vocabulary we have developed an annotator tailored towards this kind of writing style. This annotator is based on gazetteer-lists[6] of three different kinds of words expected to be found within a document written in the "Internet" writing style:

- **Smilies:** A list of common smilies[7].
- **Internet Slang:** A list of words that are regularly used within online conversations[8].
- **Swear-Words:** A list of swear words has been collected similar to the other two gazetteer lists[9].

[4] `http://maxent.sourceforge.net`
[5] `http://snowball.tartarus.org`
[6] All resources have been crawled on 2011-05-12
[7] `http://piology.org/smiley.txt`
[8] `http://www.noslang.com/dictionary/full`
[9] `http://www.noswearing.com/dictionary`

The usefulness of such annotations depends on the presence of such words in the document set. Manual inspection of a sample of the documents reveals that smilies and other Internet specific terminology is hardly used by the authors of the training data set.

Grammar Annotations The final set of annotations have been developed motivated by the intuition that the writing style of people varies in the grammatical complexity of the sentences. For example some people prefer short sentences and a simple grammatical structure, while others may prefer to construct complex sentences with many interjections. Grammatical relations have been used in the past to exploit the grammatical structure of sentences [8]. For our authorship identification system we employed the Stanford Parser [9] to produce the sentence parse tree, the sentence phrases, as well as to identify the typed grammatical relationships [10]. These informations are then added as annotations to the document.

2.2 Calculating Statistics from External Resources

For many term weighting approaches the commonness of a word plays an important part. As it is not clear whether the training set is large enough to derive reliable statistics, we incorporated an external resource to generate this information: We parsed the Open American National Corpus (OANC)[10] to gain global statistics of the usage of words. This corpus consists of many different kinds of documents and writing styles. Many of the contained documents are rather long. Therefore we employed a text segmentation algorithm [6] to split long document into smaller, coherent parts.

2.3 Feature Value Weighting

We developed multiple normalization and weighting strategies to allow a customizable representation of the frequency based document features. Finding the best suited feature weighting function for a given feature-space is an important aspect of feature engineering. This is especially true for information retrieval applications, but also applies on supervised machine learning. Depending on the feature space different strategies may lead to better performance of the classifier.

Binary Feature Value A numeric feature value is transformed into a binary value based on a given threshold. In our current system configuration the input to this normalization step is a term-count and the threshold is 1.

Locally Weighted Feature Value A more general approach than the binary feature value normalization is the application of an unary operation to a numeric feature value. In the current version we apply the $sqrt(x)$ function on the feature value x (e.g. term-count).

[10] `http://www.americannationalcorpus.org/OANC/index.html`

Externally Weighted Feature Value For the authorship identification we adapted a weighting scheme which is rooted on the well-known BM-25 retrieval model[11] - although we replaced the length normalization as the data set is expected to be vastly different to the OANC corpus. Additionally to the commonness of a term we also integrated the dispersion of its occurrences, which has proven to be beneficial to capture the semantics of short text documents [7].

For a term x, the equation of the global weighting function incorporates the feature value tf_x, the number of documents N in the OANC corpus, the number of documents the term occurs in df_x, the length of the current document and the dispersion of the term $DP(x)$:

$$w_{ext} = \sqrt{tf_x} * \frac{\log(N - df_x + 0.5)}{df_x + 0.5} * \frac{1}{\sqrt{length}} * DP(x)^{-0.3} \tag{1}$$

Globally Weighted Feature Value Another weighting strategy uses the training documents to calculate the document frequency of features. In contrast to the external weighting scheme, the term dispersion is not integrated into the weighting function. The equation for the global weighting incorporates the number of training document N and the number of documents the term occurs in.

$$w_{global} = \sqrt{tf_x} * \frac{\log(N - df_x + 0.5)}{df_x + 0.5} * \frac{1}{\sqrt{length}} \tag{2}$$

Purity Weighted Feature Value The final weighting strategy is a derivative of the global weighting function. Instead of using the individual documents of the training set, we concatenate all documents from each author into one single document. Thus, for each author there is one document which contains all terms that have been used by this author. The weight calculation is identical to the global weighting scheme. As the weighting will generate the highest values if a term is only used by a single author, we refer to this weighting scheme as a measure of purity. In the equation the term $|A|$ denotes the number of authors and af_x denotes the number of authors who have used the term at least once.

$$w_{purity} = \sqrt{tf_x} * \frac{\log(|A| - af_x + 0.5)}{af_x + 0.5} * \frac{1}{\sqrt{length}} \tag{3}$$

2.4 Feature Spaces

Our system produces two kinds of feature spaces. The first three feature-spaces (basic statistics, token statistics and grammar statistics) represent statistical properties of the documents. The range of the values for each of the features depends on the statistic property. Each of the remaining feature spaces represent a vector space model, where each feature corresponds to a single term. We conducted a series of evaluations based on the training set to find the best suited weighting strategy for each of these feature-spaces. In the following construction of the feature spaces is described in detail.

Basic Statistics Feature Space The first feature space captures basic statistics of the layout and organization of the documents. Additionally these feature spaces also carry information of the sentences and token annotations. Table 1 lists the features of this feature space.

Table 1. Overview of the features of the basic statistics feature space.

Feature	Description
number-of-lines	Number of lines
number-of-characters	Number of characters
number-of-tokens	Number of tokens
number-of-sentences	Number of sentences
number-of-text-blocks	Number of text-block annotations
number-of-text-lines	Number of lines containing more the 50% letter characters
number-of-shout-lines	Number of lines containing more than 50% upper-case characters
empty-lines-ratio	$\frac{\#emptylines}{number-of-lines}$
text-lines-ratio	$\frac{number-of-text-lines}{number-of-lines}$
mean-line-length	Average number of characters per line
mean-nonempty-line-length	Average number of character per non-empty line
max-line-length	Number of characters of the longest line within the document
text-blocks-to-lines-ratio	$\frac{number-of-text-blocks}{number-of-lines}$
{max,mean}-text-block-line-length	Number of lines per text-block (maximum, average)
{max,mean}-text-block-char-length	Number of characters per text-block (maximum, average)
{max,mean}-text-block-token-length	Number of tokens per text-block (maximum, average)
{max,mean}-text-block-sentence-length	Number of sentences per text-block (maximum, average)
{max,mean}-tokens-in-sentence	Number of tokens per sentence (maximum, average)
{max,mean}-punctuations-in-sentence	Number of punctuations per sentence, in relation to the number of tokens (maximum, average)
{max,mean}-words-in-sentence	Number of words per sentence, in relation to the number of tokens (maximum, average)
number-of-punctuations	Number of tokens tagged as punctuations
number-of-stopwords	Number of tokens marked as stop word
number-of-words	Number of non-punctuation tokens
capitalletterwords-words-ratio	Ratio of number of words with at least one capital letter divided by the number of words
capitalletter-character-ratio	Ratio of capital letters divided by the total number of characters

Token Statistics Feature Space The token and the part-of-speech (POS) annotations build the base for the token statistics feature space. For this feature space only a restricted set of word classes, namely adjectives, adverbs, verbs and nouns, are evaluated.

For each POS tag a feature is generated and its value reflects the relative number of times this tag has been detected in relation to all tokens. Furthermore the average length of all tokens for each tag is collected. Finally a set of features encodes the histogram of token length. Table 2 gives an overview of the features of this feature-space.

Table 2. Overview of the features of the token statistics feature-space.

token-length	Average number of characters of each token
token-<POS>	Relative frequency of each POS tag
token-length-<POS>	Average number of characters of each token for each POS tag
token-length-[01-20]	Relative number of tokens for the range between 1 and 20 characters

Grammar Statistics Feature-Space From the grammatical annotations a set of features have been generated. This grammar statistics feature space captures the difference in the richness of grammatical constructions of different authors. Furthermore this feature space is motivated by the assumption that individual writing styles differ in the complexity of sentences. Table 3 summarizes these features.

Slang- Word Feature Space The slang-word annotations are transformed into an own feature space. The values of the features are incremented for each occurrence of a slang word (Internet slang, smiley or swear word).

Pronoun Feature Space Based on the hypothesis that different people prefer different pronouns, we constructed a specific feature space. As with the slang-word features, each occurrence of a pronoun is counted. Finally for each document this feature-space contains the frequency of all used pronouns.

Stop Word Feature Space All tokens marked as stop-word are counted and collected. They form the stop-word feature space. In contrast to the previous two feature spaces, the occurrence counters are not directly used. Instead the binary feature value transformation is applied. Thus this feature space only captures whether a document contains a specific stop word, or not.

Pure Unigrams Feature Space All tokens, which have not been marked as stop-word or punctuation, are used to create the pure unigram feature space. Again the raw occurrence counter for each token are further processed. For this feature-space the purity based feature weighting strategy is applied. Therefore the values of this feature-space are in inverse proportion to the number of author who use a specific word.

Bigrams Feature Space Some authors may have the tendency to reuse specific phrases, a property which is not captured by the previously introduced feature spaces. Therefore we created the bigram feature space (two consecutive terms represent a single feature). For this feature-space we apply the local feature weighting strategy.

Intro-Outro Feature-Space It can be observed that some people have the tendency to reuse the same phrase to start and end their e-mails. In our authorship identification system we have developed a simple heuristic to generate a feature-space for these phrases. For each document the first and last text-block is inspected. If any of these text-blocks consists of less than 3 lines and less than 50 characters per line, its terms are added to the feature-space. Finally these features are weighted using the external feature weighting strategy.

Table 3. Feature generated out of the grammatical annotations. The values for the phrase types (<TYPE>) and relation types (<REL>) are directly taken from the parser component.

sentence-tree-depth	Average depth of the sentence parse tree for all sentences
phrase-count	Average number of phrases as detected by the parser
phrase-<TYPE>-ratio	Relative number of a specific phrase type (noun-phrase, verb-phrase, ...)
relation-<REL>-ratio	Relative number of a specific grammatical dependency type

Term Feature Space The final feature-space of our system is build using all tokens of a document. For each token its occurrence counter is tracked. Finally the feature values are processing using the weighting scheme which incorporates the commonness of word via an external resource.

2.5 Classification Algorithms

We have integrated the open-source machine learning library WEKA [4] into our system for the base classifiers. We have developed a meta classifier, which combines the results of the underlying base classifiers.

Base Classifiers We use different classification algorithms for each feature space. The configuration which has been used to produce the final result makes use of two different base classifiers. For all three statistical feature-spaces we applied Bagging [1] with Random Forests [2] as the base classifier. For the remaining feature-spaces, we used the LibLinear classifier [3] setting the type to `SVMTYPE_L2_LR` in order to get posterior probabilities of the classification results. For all classifiers, we used the default parameter settings of the library, and did not conduct any detailed analysis of the influence of the parameters on the classifiers' performance.

Meta Classifier The meta classifier combines the result of all base classifiers based on their performance on the training set, as well as on the posterior probabilities. Each of the base classifiers is trained using all documents from the training set, and 10-fold cross-validation is performed. For each of the classes (authors in this case) contained in the training set, the precision and recall on the cross-validated training set are recorded. If the precision of a class exceeds a pre-defined threshold - t_p - the base classifier is allowed to vote for this class. For all classes where the recall is higher than another threshold - t_r - the classifier is permitted to vote against a class (veto). Additionally each base classifier has a weight w_c controlling its impact on the final classification. Further, for unseen documents, the posterior probabilities of the base classifiers are also evaluated to asses whether a base classifier's result will be included in the final prediction. If the probability for an author is higher than p_p and the classifier is allowed to vote for this author, the probability is multiplied by w_c and added to the authors score. If the probability is lower than p_r (again assuming the classify may veto against this author), the probability times the weight is subtracted from the authors score.

The classifier, which operates on the term feature space, is treated differently. The classification results of this classifier are always used without any additional weighting

Table 4. Assessment of the base classifiers performance after the training phase. For each of the voting authors the precision of the base classifier exceeds a threshold. For each of the veto authors the recall of the base classifier has been higher than a pre-defined threshold.

Classifier	#Authors Vote	#Authors Veto
basic-stats	4	14
token-stats	5	7
grammar-stats	5	5
slang-words	3	2
pronoun	6	1
stop-words	4	10
intro-outro	25	11
pure-unigrams	6	15
bigrams	20	23

or filtering. Thus all predicted probabilities of this classifier will be added to the scores of the authors.

Finally, the author with the highest score is considered to be the most probable author. In the evaluation section we report the performance of our meta-classifier using the parameters: $w_c = 0.9, t_p = 0.5, p_p = 0.3, t_r = 0.3, p_r = 0.01$. We conducted several experiments to find the parameters values that provide the best performance. During these evaluations we found that individual settings for each base classifier produced the optimal results. But finally we decided to use a single set of parameters for all classifiers as our experiments were conducted only on a single test corpus (LargeTrain) and we want to (i) avoid any over-fitting problems, and (ii) keep an already complex system as simple as possible.

3 Results

For brevity we only report the results of the evaluation achieved on the data-set labeled as "LargeTrain". Table 4 shows the number of votes an vetos of the different base classifiers. Table 5 summarizes the vote and veto performance of the base classifiers on the validation data set.

Further our results were evaluated by the PAN team on unseen evaluation data sets. The PAN author identification challenge consisted of two tasks: authorship attribution and authorship verification. Altogether 17 runs were submitted by 12 different groups. We focused on the authorship identification task and used the resulting system (as outlined in the previous section) for the verification task as well. We present the macro and micro precision, recall and F1-meassure of our system on the four authorship attribution datasets and the three authorship verification datasets in Table 6.

Our system performed second and third best in the authorship attribution task on the large data sets, and ranked middle for the small data set in the attribution task and in the verification task. On the author attribution datasets we compared very favorable in terms of precision compared to the best ranked systems. Our macro recalls tended to

Table 5. Vote and Veto - Performance of the base classifiers on the TrainValid data-set.

Classifier	Vote Accuracy	Vote Count	Veto Accuracy	Veto Count
basic-stats	0.958	5141	1	252380
tokens-stats	0.985	1056	1	77492
grammar-stats	0.980	2576	1	89085
slang-words	0.819	94	0.997	9277
pronoun	-	0	1	85
stop-words	0.532	1924	0.998	107544
intro-outro	0.826	2101	0.998	102431
pure-unigrams	0.995	186	0.999	35457
bigrams	0.999	6239	1	281442

be lower on average. Given that our system has not explicitly been developed for author verification, it did perform acceptably for the first two verification datasets.

4 Conclusion and Future Work

We presented our system for authorship attribution in the context of the PAN 2011 author identification challenge. Our system is based on heavy feature engineering as well as combining multiple classifiers in a unique way. In terms of feature engineering we employed standard corpus statistics as well as features tailored towards the domain of e-mails. In addition we also integrated features determined from annotations created by a syntactical natural language parser.

Given this abundant number of different feature spaces we settled on a voting strategy for classification in which we train base classifiers for each individual feature space and apply voting and vetoing as well as weighting to form the classification model of an author. These two stages proved to perform acceptably in the context of the PAN challenge, especially in the case of the large author attribution data sets. Future work will include the augmentation of our vetoing scheme with boosting and ensemble classification principles as well as applying our system to other domains.

Acknowledgement

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. Breiman, L.: Bagging predictors. Mach. Learn. 24, 123–140 (August 1996)
2. Breiman, L.: Random forests. Mach. Learn. 45, 5–32 (October 2001)

Table 6. Macro & micro average precision, recall and F1 on the evaluation datasets. The last column indicates our rank (out of 17) for the dataset compared to the other teams. Additionally the relative difference to the highest rank run is given below each evaluation result.

Test Set	Macro Prec	Macro Recall	Macro F1	Micro Prec	Micro Recall	Micro F1	Rank
LargeTest	0.615	0.442	0.465	0.642	0.642	0.642	2
	+0.066	-0.090	-0.055	-0.016	-0.016	-0.016	
LargeTest+	0.673	0.179	0.226	0.802	0.383	0.518	3
	-0.015	-0.088	-0.095	+0.023	-0.088	-0.069	
SmallTest	0.79	0.345	0.348	0.685	0.685	0.685	5
	+0.128	-0.106	-0.127	-0.032	-0.032	-0.032	
SmallTest+	1	0.03	0.05	1	0.095	0.173	8
	+0.263	-0.131	-0.143	+0.176	-0.362	-0.415	
Verify1	1	0.333	0.5	0	0	0	2
	±0	±0	±0	±0	±0	±0	
Verify2	0.5	0.2	0.286	0	0	0	6
	+0.100	-0.600	-0.247	±0	±0	±0	
Verify3	0	0	0	0	0	0	10
	-0.211	-1.000	-0.348	±0	±0	±0	

3. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. 9, 1871–1874 (2008)
4. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11, 10–18 (November 2009)
5. Juola, P.: Authorship attribution. Found. Trends Inf. Retr. 1, 233–334 (December 2006), http://portal.acm.org/citation.cfm?id=1373450.1373451
6. Kern, R., Granitzer, M.: Efficient linear text segmentation based on information retrieval techniques. In: MEDES '09: Proceedings of the International Conference on Management of Emergent Digital EcoSystems. pp. 167–171 (2009)
7. Kern, R., Granitzer, M.: German Encyclopedia Alignment Based on Information Retrieval Techniques. In: Lalmas, M., Jose, J., Rauber, A., Sebastiani, F., Frommholz, I. (eds.) Research and Advanced Technology for Digital Libraries. pp. 315–326. Springer Berlin / Heidelberg (2010)
8. Kern, R., Muhr, M., Granitzer, M.: KCDC: Word Sense Induction by Using Grammatical Dependencies and Sentence Phrase Structure. In: Proceedings of SemEval-2. Uppsala, Sweden, ACL (2010)
9. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - ACL '03 pp. 423–430 (2003)
10. de Marneffe, M., MacCartney, B., Manning, C.: Generating typed dependency parses from phrase structure parses. In: LREC 2006 (2006)
11. Robertson, S., Gatford, M.: Okapi at TREC-4. In: Proceedings of the Fourth Text Retrieval Conference. pp. 73–97 (1996)
12. Stamatatos, E.: A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol. 60, 538–556 (March 2009), http://portal.acm.org/citation.cfm?id=1527090.1527102