Authorship attribution: using rich linguistic features when training data is scarce
Notebook for PAN at CLEF 2012

Ludovic Tanguy, Franck Sajous, Basilio Calderone, and Nabil Hathout

CLLE-ERSS: CNRS and University of Toulouse, France
firstname.lastname@univ-tlse2.fr

Abstract We describe here the technical details of our participation to PAN 2012's "traditional" authorship attribution tasks. The main originality of our approach lies in the use of a large quantity of varied features to represent textual data, processed by a maximum entropy machine learning tool. Most of these features make an intensive use of natural language processing annotation techniques as well as generic language resources such as lexicons and other linguistic databases. Some of the features were even designed specifically for the target data type (contemporary fiction). Our belief is that richer features, that integrate external knowledge about language, have an advantage over knowledge-poorer ones (such as words and character n-grams frequencies) when training data is scarce (both in raw volume and number of training items for each target author).
Although overall results were average (66% accuracy over the main tasks for the best run), we will focus in this paper on the differences between feature sets. If the "rich" linguistic features have proven to be better than trigrams of characters and word frequencies, the most efficient features vary widely from task to task.
For the intrusive paragraphs tasks, we got better results (73 and 93%) while still using the maximum entropy engine as an unsupervised clustering tool.

1 Introduction

As many other textual classification tasks, authorship attribution sees a competition between linguistic (lexical, syntactic, semantic) and information-poor features (such as character trigrams and word frequencies).

We have already measured that the latter need a minimal amount of data (both training and testing) in order to be efficient, while the injection of information with NLP techniques can lead to good results in contexts where data is scarce (forthcoming publication). [6] have studied the effect of training data size, and one of their experiments seems to show that character trigrams are the features that benefit the most from an increase of data.

In any case, the collaboration between linguistic features and character trigrams is a good bet, as shown in [6] and from our own results in last year's competition.

However, an important part of the process is the selection of features, as an increase in the number and variety of features does not necessarily entails a gain in accuracy. It now appears clearly that we have overlooked these questions in the work presented here.

We will first describe in section 2 the techniques we have used, in terms of machine learning and in more details of the features we designed and used. Section 3 describes the results obtained for the more classical subtasks (attributing authors from a given set to a text, with or without additional unknown authors), as well as a closer inspection of the features we used. We also performed a series of additional runs in order to test some hypotheses regarding the preprocessing of training data. The last section (4) describes the methods we used for the paragraph intrusion subtasks, where unsupervised machine learning techniques were necessary, but with a very similar approach in terms of features and tools.

2 Overview of the techniques used

2.1 Machine learning tool

Our approach is technically built around a maximum entropy machine learning tool, named *csvLearner*[1] which has been designed by our colleague Assaf Urieli based on the OpenNLP MaxEnt library. We chose this technique for its ability to provide good results for this task while processing a large number of (possibly redundant) features [8], and because it provides probabilities which can be directly used for the open-class subtasks, and also because it can be used as a basis for unsupervised learning (see section 4).

We have not tried to adapt the training parameters, and used the following values for all the runs:

- no cutoff (i.e. no minimal frequency for features);
- 100 iterations for the training phases;
- no smoothing for feature values;
- features values have been linearily normalised based on training data.

2.2 Data preparation

Each text file has been processed as follows:

1. *Normalisation:* character encoding normalisation to Latin1. This mostly concerned the diverse encodings encountered for apostrophes and dashes (custom program). If the variation between encodings could have been used as a feature, we deemed it would reflect the characteristics of the publishing (or acquisition) of the texts rather than those of the authoring process;
2. *Dehyphenation:* for each line ending with a dash, we checked if this could be considered as an hyphenation by looking up the hypothetical resulting word in a generic English lexicon. If so, hyphens were deleted;
3. *Tokenization*: identification of word and sentences boundaries, according to punctuation marks;
4. *POS tagging*: identification of each word's part-of-speech (POS) category (Noun, Verb, etc.), along with a number of inflexional features (number, tense, etc.);

[1] https://github.com/urieli/csvLearner

5. *Lemmatization*: identification of each word's lemma, or citation form;
6. *Syntactic parsing*: identification of the syntactic relationships between words in a sentence. We have used a dependency analyzer that provides tagged pairwise links between syntactically related words (subject, object, determiner, etc.).

Apart from the first two steps, which were adressed with custom programs, we used the Stanford CoreNLP suite [5] for the linguistic processing. We discarded some modules we found irrelevant for the task and data (named entities recognition and anaphora resolution).

In addition, for the intrusive paragraphs subtasks (E and F), the texts have been tokenized in paragraphs according to empty lines, and each paragraph processed as a single text (see below).

2.3 Features

We describe below the list of features we used for our main run. Most of them make use of the linguistic annotations described above and of additional generic ressources. Feature sets indicated with a star (*) only contains synthetic values, and have been selected for one of our submitted runs (number 4).

Most of these features (except for the last three sets) were used in our participation in last year's competition and are described in [8]. We excluded some previously used features, such as spelling errors (hopefully irrelevant for published fiction), UK/US variants, and other specific features tailored to last year's data (corporate emails). We give an indication of the number of features based on the training data for task C (16 short texts).

Frequency of character trigrams – This straightforward aspect of textual characteristics is one of the star features for authorship attribution tasks. A study of the notebooks from previous PAN campaigns has shown that it is the most commonly used over all tasks and methods (more than one run out of two uses them). As noted above, no selection has been performed (no cutoff). However, due to technical limitations given the cost of training with too numerous features, we limited their number for the tasks dealing with novel-length texts (tasks I and J), by filtering out the trigrams with a frequency below 5 in the training data.

For subtask C (16 short texts), we found 9684 different trigrams in the training data.

Contracted forms (*) – This single feature measures the relative frequency of contracted versus developed forms. Based on a list of 200 possible contractions (e.g. *we're*, *isn't*, etc.), we computed the ratio of contractions used by the author.

Phrasal verbs – This set of lexical features is based on the syntactic analysis, and consists of the frequency of each verb-preposition pair found in the text (*e.g.* *put up*, *go to*, etc.). The dependency analysis is able to detect such pairs of words even if they are not adjacent.

An additional synthetic feature (*) corresponds to the relative cumulated frequency of phrasal verbs in a text.

For subtask C, 357 different phrasal verbs were found.

Lexical genericity and ambiguity (*) – This small set of features is based on the Princeton WordNet lexical database [4][2], and aims to capture two characterisitcs of the lexicon of a text. Genericity is measured in terms of depth in the WordNet taxonomy: for each noun, verb, adjective and adverb, we computed the average and maximal depth of the synsets in which this word appears.

Ambiguity is the average number of synsets for the same classes of words.

This set thus comprises 12 values for each text.

Frequency of POS trigrams – Based on the part-of-speech tags produced by the parser, we simply computed the frequency of each trigram of such tags.

For subtask C, 8,827 different trigrams were found.

Syntactic dependencies – This set of features is based on the ouput of the syntactic dependency parser, in the form of triplets ($word_1$, *relation*, $word_2$), such as (*cat*, SUBJ, *eat*) if the noun *cat* is the subject of the verb *eats*. The features we designed are the relative frequencies of each triplet.

Two different subsets were computed: the first one takes the lemmas into account while the second only considers the POS tags of the two words.

For subtask C, the first subset contains 57,760 features and the second one 2,177.

Syntactic complexity (*) – In order to capture the relative syntactic complexity in a text, we measured two different parameters for each sentence.

The first is the depth of the syntactic tree resulting from the syntactic dependencies provided by the parser (after minor transformations). We measured both the maximal depth for each text and the average depth.

The second parameter is more directly related to the output of the parser. We measured the average and maximal distance (expressed in number of words) covered by each dependency link.

This subset of features thus comprises 4 values for each text.

Lexical cohesion (*) – The synthetic features in this set are based on the semantic similarity of words appearing in the target texts. The similarity measure has been extracted from the the Distributional Memory database [2], automatically built on large generic corpora. In this approach, two words are deemed similar if they appear in the same syntactic contexts (e.g. two nouns are similar if they appear as subjects (or objects) of the same verbs).

More precisely, we used the TypeDM model, and extracted, for each word, the most similar words based on a cosine measure. For each token in the analysed texts, we then

[2] http://wordnet.princeton.edu/

counted the number of similar words appearing in the same text. Only nouns, adjectives, adverbs and verbs were concerned, and we used 5 different sizes for the similar word lists (1, 5, 10, 15 and 20 most similar words). Computing the average values over each word in the text, we obtained 5 features for this set.

Morphological complexity (*) – The CELEX morphological database [1] was used to assess the morphological complexity of each text's lexicon.

The first feature is the ratio of morphologically complex words, obtained through a simple lookup in the database, ignoring the words absent from the list.

For the second feature, we wanted to expand the coverage of the database, and thus extracted the different suffixes. These suffixes were then used as a rough means to select whether each word is possibly suffixed or not. The feature is in the end the ratio of possibly suffixed words in the text.

Lexical absolute frequency (*) – This set of features has been computed in order to capture an author's lexical coverage and specificity. We used the frequency lists provided by Paul Nation [7], which comprise the most frequent 3000 words (and word families) in English. The words are divided into 3 lists according to their frequency range. These lists and associated frequency values have been used in Paul Nation's RANGE program mainly used for selecting course materials for vocabulary teaching.

For each list of words, we computed both the ratio of words and the average frequency of a text's vocabulary. We also computed the average frequency over the three lists, thus amounting to 8 features.

Punctuation and case (*) – A small set of features adresses the shallower specificities of an author's writing. We thus computed the frequency of each punctuation mark (and sequences of repeated marks) and the use of uppercase (ratio of fully uppercase words).

This set comprises a total of 22 features for subtask C.

Quotations (*) – As quotations are an important feature of fiction texts, we computed a single feature corresponding to the relative frequency (per sentence) of sequences between quotes.

First/third person narrative (*) – Another discrimnative feature of contemporary fiction is the difference between third person and first person narrative, i.e. if the narrator is referred to in the text or not. In order to do this, we computed the ratio of first person subject pronouns for each verb (outside quotations).

Proper names – This last set of features has only been used for the intrusive paragraphs subtasks (E and F). Again, as character names are important characteristics of fiction, we specifically calculated the frequency of individual proper names in each paragraph, according to the POS tagger.

2.4 Submitted runs

We submitted a total of four runs using different feature sets as a basis for comparison:

- Run 1: all of the above features;
- Run 2: frequency of character trigrams;
- Run 3: frequency of lemmas;
- Run 4: a selection of 60 features from the above list, focusing on synthetic linguistic measures (i.e. feature sets indicated with a star (*)).

3 Traditional attribution subtasks

This sections reports in more details the classification methods we used for the standard problems of authorship attribution.

Compared to previous campaigns, the amount of training data was extremely small: only two texts per author were available for training. Task A to D concerned short texts (from 2,000 to 13,000 words), while tasks I and J dealt with novel-length texts (30,000 to 160,000 words).

We initially considered splitting the text into smaller segments in order to increase the training items, but finally decided to use each whole text as a single item. The main point supporting this awkward approach was that artificially splitting the texts could lead to misleading results in the cross-validation process, as we suppose that segments from the same work will obviously share stylistic and lexical similarities. These similarities are not expected to be so obvious when addressing a different work by the same author (see below for a confirmation of this).

The main consequence of this decision was that we could not perform any cross-validation evaluation (having only 2 items per author for training), and thus blindly designed our runs based on previous experience on other kinds of data.

However, we performed some post-hoc lesion tests based on the result files made available after the evaluation in order to assess the relative efficiency of each feature set, as presented below.

3.1 Closed versus open class problems

Tasks A, C and I are closed-class problems, with no possible author apart from the ones in the training set. For our runs, the author with the highest probability according to the trained maximum entropy model was selected.

For tasks B, D and J, for which test data could contain texts by unknown authors, we applied a dynamic threshold for the highest probability. This threshold was computed for each text in the test dataset as the average probability (over all possible authors) plus 1.25 standard deviation. If the probability value for the most probable author was inferior to this value, the answer was “unknown author” instead.

Once again, the scarcity of training data led us to chose this particular value based on previous experience on other data.

3.2 Results and discussion

Table 1 shows the accuracy scores for each of our runs for each subtask, as well as the overall accuracy.

Run	Task A	Task B	Task C	Task D	Task I	Task J	Overall
1 (rich and char. trigrams)	100	60	37.5	41.2	85.7	75.0	**66.6**
2 (char. trigrams)	66.7	50	25.0	11.8	85.7	81.2	53.4
3 (bag of words)	100	60	37.5	11.8	92.9	81.2	63.9
4 (synthetic rich)	100	50	37.5	29.4	92.9	75.0	64.1

Table 1. Accuracy per task and per run

Based on the overall score, our best run (number 1) finished at the 10th position (out of 21 submitted runs).

As can be seen, results vary widely from one task to another. If task A clearly proved to be very easy (only three authors), task C was astonishingly difficult in comparison (short texts, 8 authors), although for other competitors the difference was not so important compared to other tasks.

Open-class problems systematically are more difficult, as it was the case for most of the submitted runs. Our main run (1) performed better for task D than C because all the selected unknown authors were right.

When comparing our different runs, the results confirm the advantage that linguistically rich features have over information-poor approaches, with the exception of task J for which runs 2 and 3 obtained slightly better results.

3.3 To split or not to split...

Our decision of not splitting the training and testing data has been explained above. Once the test results were available, we wanted to test a posteriori this decision, and ran a number of tests that used segments of texts as items. We arbitrarily decided to split both training and test data into segments of 500 and 1000 words. For attributing an author to a text from the test data, we selected the one that maximizes the cumulated probabilities over the segments. Table 2 shows the results over the three closed-class tasks. We measured both the accuracy over the texts, but also over the text segments (numbers in parentheses). We also performed a cross-validation measure (having splitted the training texts, it could be done), using a 10-fold measure over the text segments.

As these results clearly show, it has been a good move to keep the whole texts as items for the classification process. Only a few cases show an improvement, mainly for the character trigrams for tasks A and C. Taks I shows a dramatic decrease in accuracy for all four runs when splitting texts.

It is also interesting to notice that the cross-validation measures are all very high, and could not reliably help us in selecting the best runs, as was expected, as no obvious correlation appears between these values and the actual accuracy scores obtained on test data.

Run	Whole texts	500 words	1000 words	Cross-V (500)	Cross-V (1000)
Task A					
RUN1	100	100 (94)	66.7 (89)	100	100
RUN2	66.7	83.3 (88)	83.3 (93)	100	100
RUN3	100	100 (90)	100 (96)	100	100
RUN4	100	100 (84)	83.3 (81)	96	100
Task C					
RUN1	37.5	25 (36)	25 (35)	90	81
RUN2	25	37.5 (34)	25 (32)	97	97
RUN3	37.5	37.5 (37)	37.5 (38)	96	95
RUN4	37.5	25 (40)	25 (35)	84	80
Task I					
RUN1	85.71	21.43 (57)	50 (63)	96	99
RUN2	85.71	21.43 (51)	42.86 (59)	95	99
RUN3	92.86	14.29 (51)	35.71 (64)	96	99
RUN4	92.86	7.14 (51)	42.86 (59)	96	99

Table 2. Results obtained when splitting texts: accuracy over texts (and segments) and cross-validation values over training data

3.4 A closer look at the features

Although no cross-validation results were obtained over training data, the ground truth results also provided us with means to evaluate the relative efficiency of our linguistic features.

Lesion study for problem C – Starting with problem C, we performed a lesion study by considering different combinations of feature sets, and compared the relative use of each one. Our method consisted in removing feature subsets from the ones used for run number 1, each time measuring the difference in accuracy, while restricting at each step to the best performing combination.

The first interesting results is that we got very good performances with a very small set of features, namely the sole combination of morphological complexity (2 features) and punctuation-case (22 features). The accuracy (62.5%) was much higher than the one we got with run number 1, it is interesting to note that such level can be obtained with very few features.

While measuring the average (over all tested combinations) gain/loss value for removing each individual subset of features, we got the results shown in table 3. Feature sets are sorted in decreasing efficiency, as a negative value indicate an average decrease in accuracy when removed.

Sample author/features associations When looking up the detailed values, one can indeed find important variations between authors for the identified features. A few examples are shown below.

Regarding punctuations, author A is the only one who uses underscores, author C is the sole writer of ampersands, author E uses a lot of question marks, authors D and E have an affinity for colons, and so on.

Feature set	**Accuracy gain when removed (%)**
Punctuation/case	-0.204
Suffix frequency	-0.097
Absolute lexical frequency	-0.030
Syntactic complexity	-0.015
Ambiguity/genericity	-0.012
Lexical cohesion	-0.002
Phrasal verbs (synthetic)	-0.000
Morphological complexity	0.005
Phrasal verbs (detail)	0.006
Contractions	0.014
First/third person narrative	0.027
POS trigrams	0.028
Char. trigrams	0.034
Syntactic dependencies	0.059

Table 3. Results of lesion studies (task C, per features subset)

Author C uses a lot of (varied) suffixes, and is also the author that uses mostly low-frequency words.

The variation in syntactic complexity opposes author like E, with complex sentences like this one (12CtrainE2):

> But it also intrigued me just enough to explore it a little further - the possibility that a truly random event like the tossing of a coin, or the pattern of some chicken bones scattered upon the ground, or the decay of a radioactive substance could somehow reach back into the void between the atoms and truly gauge a moment in time.

In fact, it appears that both of author C's texts for tasks C and D were only correctly guessed by our runs that used rich features, while information-poor runs failed.

Lesion study for problem A – When performing the exact same lesion study for task A, we found very confusing results, leading to exactly opposite conclusions concerning the relative efficiency of the features. Table 4 shows that the most useful feature sets are the ones that were the more damageable for task C (character trigrams, syntactic dependencies and POS trigrams). Similarly, the best feature set for task C (punctuation and case) is the last one for task A.

All these studies, although confirming the importance of linguistically-rich features, do not enlighten us on the ways we can predict the best combination of features. As cross-validation cannot be relied on in this specific situation where too few different texts are available (whatever their size), it remains difficult to assess the a priori value of a combination and/or set of parameters.

4 Paragraph intrusion subtasks

Tasks E and F address a very different problem, which can be seen as a mix between plagiarism detection and authorship attribution. The test data consists of short texts in which the paragraphs are written by different authors.

Feature set	**Accuracy gain when removed (%)**
Char. trigrams	-0.206
Syntactic dependencies	-0.089
POS trigrams	-0.045
Phrasal verbs (synthetic)	-0.022
Contractions	-0.018
Suffix frequency	-0.009
Ambiguity/genericity	-0.008
Syntactic complexity	-0.006
Lexical cohesion	-0.000
Morphological complexity	0.002
Absolute lexical frequency	0.003
Phrasal verbs (detail)	0.006
First/third person narrative	0.026
Punctuation/case	0.040

Table 4. Results of lesion studies (task A, per features subset)

Task E texts are written by an unknown number of authors (they are extracted from different texts), supposedly at random. Task F texts only contain one small sequence of intrusive paragraphs. Both these tasks call for unsupervised machine learning, as no texts from the target authors are provided (the intrusive authors being unknown in a case of plagiarism).

Our approach used the same features we designed for the other tasks (with an additional subset consisting of the frequency of individual proper names). We also used our maximum entropy machine learning tool, but in a quite different way, following the method proposed by [3].

More precisely, each text in the dataset was processed independently. We isolated each paragraph (according to blank lines), computed the feature values for each one, and used them as training items with their unique ID (number) as a class value. Once the model was trained, each paragraph was submitted in turn to the classifier, and we focused on the resulting probabilities.

These probabilities are read as similarity measures based on the trained model: if two paragraphs have similar feature values, the classifier will attribute a higher probability for selecting the correct class/ID (of course, the ID of the paragraph itself will systematically have a higher probability).

We applied a $-log(p)$ transformation to get a dissimilarity (distance) matrix between the different paragraphs. This matrix was then analysed by a hierarchical ascendant clustering (HAC) algorithm.

Figure 1 shows the resulting dendogram of the clustering for text 4 of task F, as produced by our run number 1. Each paragaph is identified by its number, and the dendogram displays from bottom to top the iterative agglomeration of paragraphs based on their similarity. For example, the two most similar paragraphs are number 13 and 14, then 16 and 18, and so on toward the top of the figure.

The smallest topmost cluster has been highlighted, as it was our answer for the consecutive intrusive paragraphs (paragraphs 11 to 15).

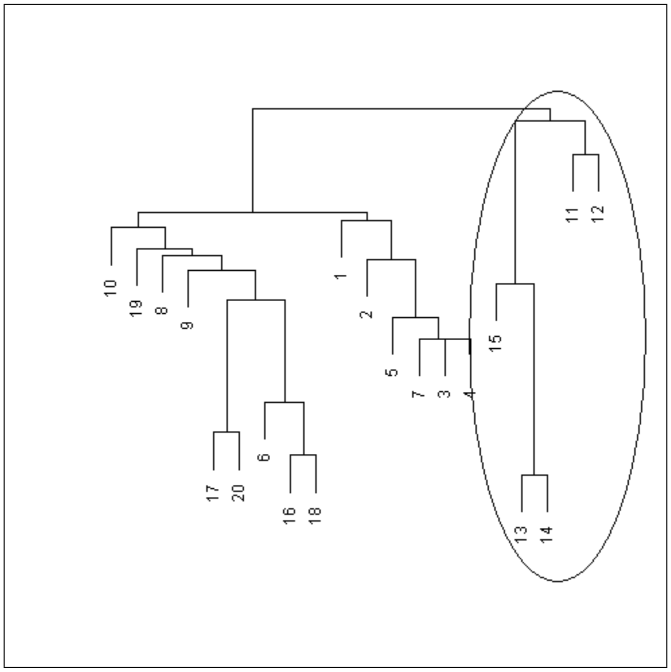

Figure 1. Dendogram of the hierachical ascendant clustering for text 4 from task F, run 1.

The most important parameter for the hierarchical clustering is the linkage strategy, i.e. the function that computes the distance between two clusters, in order to decide which set of paragraphs are clustered together at each step of the algorithm. It is vital in this kind of technique because the similarity measure we rely on is only defined between individual paragraphs (it would have been way too costly to retrain the model each time the algorithm has decided that two paragraphs are similar). The most common methods are to take the average or minimal distance among all pairwise distance measures.

Based on experiments with the supplied example files, we decided to use the following processes for each task:

- For task E (random mixed paragraphs), we used an average linkage strategy for the clustering and simply selected the two top-level clusters (obtained by cutting the tree along its top fork). We reached an accuracy of 73% of correctly attributed paragraphs with our main feature sets (rich features and character trigrams). Both our runs that used linguistic features gave better results than the other combinations.
- For task F (consecutive intrusive paragraphs), we used a single (minimum) linkage strategy, then applied the following post-processing. The smaller of the two top-level clusters was selected, but other paragraphs were added on the condition that they filled single-width gaps (i.e. if the cluster contains paragraphs 1 and 3, we added number 2). The longest subset of consecutive paragraphs was then selected as the answer. In case of a tie (the only cases encountered were for single paragraphs),

the highest subset in the clustering tree (i.e. more dissimilar) was selected. In case of further tie, the first subset in the natural text order was selected. This approach led to good results (94 and 89%, depending on the features used, with once more the best score reached by our main run).

Testing these methods on the provided training files gave perfect results regardless of the feature sets used. We did not use any threshold for the similarity measures, so this method is not as yet suited for the detection of intrinsic plagiarism in a text: our system always provide an answer, even if there are no intrusive parts in the analysed text. Also, for task F, the fact that the intrusive paragraphs are consecutive is an important clue for the process.

The resulting scores show a good promise for this approach, although the late arrival of the official results could not allow us to analyse them any further.

5 Conclusion

Our participation to this year's PAN competition can be said to be a disapointment, given the global results over the main tasks.

On the positive side, we confirmed that rich linguistic features outperform information-poor ones, whatever the task and parameters. On the negative one, much has yet to be done in terms of feature selection. It appears that some features are best suited to specific data or authors: a few well-chosen textual characteristics can give much better results than a simple accumulation of features. It thus invites us to continue designing new features, especially those that target specific aspects of an author's style. The inner mechanisms of modern machine learning techniques (such as maximum entropy) are in fact an important obstacle to our understanding of automated authorship attribution mechanisms. Our next move will be to have a closer look at the variation (over training data) of each individual feature in order to extract the most promising ones.

References

1. Baayen, R.H., Piepenbrock, R., Gulikers, L.: The CELEX lexical database (release 2). CD-ROM (1995), linguistic Data Consortium, Philadelphia, Penn.
2. Baroni, M., Lenci, A.: Distributional memory: A general framework for corpus-based semantics. Computational Linguistics 36(4), 673–721 (2010)
3. De Pauw, G., Wagacha, P.W.: Bootstrapping morphological analysis of gĩkũyũ using maximum entropy learning. In: Proceedings of the eighth INTERSPEECH conference (2007)
4. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press (1998)
5. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Meeting of the Association for Computational Linguistics. pp. 423–430 (2003)
6. Luyckx, K., Daelemans, W.: The effect of author set size and data size in authorship attribution. Literary and Linguistic Computing 26(1), 35–55 (2011)
7. Nation, P.: Using small corpora to investigate learner needs. Small Corpus Studies and ELT: theory and practice pp. 31–45 (2001)
8. Tanguy, L., Urieli, A., Calderone, B., Hathout, N., Sajous, F.: A multitude of linguistically-rich features for authorship attribution. In: Notebook for PAN at CLEF 2011. Amsterdam (2011)