SU@PAN'2016: Author Obfuscation

Notebook for PAN at CLEF 2016

Tsvetomila Mihaylova[1], Georgi Karadjov[1], Yasen Kiprov[1], Georgi Georgiev[1],
Ivan Koychev[1], and Preslav Nakov[2]

[1] Faculty of Mathematics and Informatics, Sofia University "St. Kliment Ohridski", Bulgaria
{tsvetomila.mihaylova, georgi.m.karadjov}@gmail.com,
{yasen.kiprov, g.d.georgiev}@gmail.com, koychev@fmi.uni-sofia.bg
[2] Qatar Computing Research Institute, HBKU, Qatar
pnakov@qf.org.qa

Abstract The anonymity of a text's writer is an important topic for some domains, such as witness protection and anonymity programs. Stylometry can be used to reveal the true author of a text even if s/he wishes to hide his/her identity. In this paper, we present our approach for hiding an author's identity by masking their style, which we developed for the Author Obfuscation task, part of the PAN-2016 competition. The approach consists of three main steps: the first one is an evaluation of different metrics in the text that can indicate authorship; the second one is application of various transformations, so that those metrics of the target text are adjusted towards the average level, while still keeping the meaning and the soundness of the text; as a final step, we are adding random noise to the text. Our system showed the best performance for masking the author style.

1 Introduction

Stylometry is a well-studied topic. Detecting the author style in particular, has been studied for years and different approaches have been explored.

However, the reverse process, i.e., hiding the style of an author, is less explored. It has a lot of challenges, not only the author style has to be hidden, but the text needs to remain grammatically correct and the meaning of the original text needs to be preserved.

The PAN-2016 [2] Author Obfuscation task [16] is divided into two subtasks - *Author Masking* and *Obfuscation Evaluation*. The *Author Masking* task seeks for solutions solving the following problem: "Given a document, paraphrase it so that its writing style does not match that of its original author, anymore." The documents given for obfuscation have to be split into parts of up to 50 words each and each part is then subject to obfuscation. The outcome is evaluated by three criteria: *safety*, i.e., whether the author verification systems can detect the author from the obfuscated text; *soundness*, which measures whether the result is entailed from the original, and *sensibleness*, which checks whether the obfuscated text is meaningful. The latter two were evaluated with peer review. The *Obfuscation Evaluation* subtask asks the participants to propose automated measures for evaluation of the first subtask. Measures for one or more of the criteria can be suggested. We have participated in both subtasks.

2 Related Work

Author identification is a well-studied topic. For instance Juola et al., 2011 [10] analyzed the features in JGAAP (Java Graphical Authorship Attribution Program) and built a model using them, e.g., words, parts of speech, characters, and word bi-grams.

Author Identification has been explored as a task at the PAN competition since 2011. The PAN-2015 task description paper [18] summarizes the approaches and features used for author identification. Among the most used features are the lengths of words, sentences, and paragraphs, type-token ratios, hapax legomena, character n-grams (including unigrams), words, punctuation marks, stopwords, part of speech n-grams. Other features analyze the text more deeply by checking style and grammar.

Kacmarcik et al. 2006 [11] explored author masking by detecting the most used words by the author and trying to change them. They also mention the application of machine translation as a possible approach for author obfuscation. Some authors describe using machine translation as a means for author obfuscation ([17], [4]) (translating passages of text from English to one or more other languages and then back to English). Brennan et al. 2012 [4] investigate three different approaches for adversarial stylometry: obfuscation (masking author style), imitation (trying to copy another author's style) and machine translation. They have summarized the features people use most when trying to obfuscate their own writing style.

Juola et al. 2011 [10] experiment with different techniques for author obfuscation. Their system consists of three main modules - canonization (unifying case, normalizing whitespaces, spelling correction, etc.), event set determination (extraction of events significant for author detection, such as words, parts of speech bi- or tri- grams, etc.), statistical inference (measures that determine the results and confidence in the final report). The same authors used this approach ([9]) to detect deliberate style obfuscation. Some other features used for author recognition are personal pronouns, sentence length, unique words, and parts of speech ([1]).

In our work here, we study most of the features mentioned by the research described above in order to mask the author style, i.e., to address the *Author Masking* task.

3 Method

Our approach measures some of the most significant features of the text used for author identification as mentioned in the work of Brennan et al. 2012 [4]. After that we apply transformations that change the calculated metrics, so the text has average values for the aforementioned metrics.

The system consists of three main parts. First, we calculate "average" metrics based on the training corpus provided for the Author Obfuscation task and a corpus of several public domain books from Project Gutenberg [7]. Having the average metrics, before transforming each document, we calculate the corresponding metrics for it. Then transformation for each metric is applied, depending on whether its value is below or above the calculated average. After the targeted transformations are applied, additional transformations are added to transform the text beyond the target metrics. We chose very safe transformations, so that the text would not change its meaning. We used dictionaries to transform abbreviations, equations, and short forms to their text alternatives.

3.1 Calculating text metrics

We used the following metrics:

1. Average sentence word count;
2. Punctuation to word count ratio;
3. Stop words to word count ratio;
4. Type-token ratio;
5. POS to word count ratio: measured for four part-of-speech groups: nouns, verbs, adjectives, and adverbs; we used the Python NLTK [3] with Universal Tagset for part of speech tagging;
6. Words in all capital letters to word count ratio;
7. Count of each word in the text.

We calculate "average values" that were obtained by calculating the average for the above metrics on the training corpus and a corpus of several public domain books from Project Gutenberg [7]. Before splitting each document in parts to be obfuscated, we calculate the above measures for it. For each part, we compare the document measure values of the calculated averages and we apply transformations to increase the value if it is below the corresponding average, or to decrease it when it is above it.

3.2 Modulizing the text

The given texts are split into parts of up to 50 words each, according to the task requirements. First, the text is split into sentences using the NLTK sentence splitter. Each part of the text was obtained by merging sentences while the sample had less than 50 words. We ignored paragraph separation for this splitting.

3.3 Text Transformations

1. **Splitting or merging sentences**
   If the average sentence length of the whole document is below the average, we perform merging of the sentences for each text part. We merge all the sentences for a given text part into one sentence. Merging is done by adding a random connecting word (*and, as, yet*) and randomly inserting punctuation - comma (,) or semicolon (;). When the average sentence length of the entire document is above the average, we split the sentence into shorter ones. We use a simple sentence splitting algorithm: We go through all POS-tagged words in the text, and we count the nouns and the verbs. When we reach a conjunction *and*, if the sentence so far contains a noun and a verb, we replace the *and* with a comma (,) and we capitalize the next word's initial as it will now start a new sentence.
2. **Stop Words**
   Stop words can be strong indicators for author identification due to the fact that some authors have the tendency to use specific stop words or to have specific stop words to other words ratio. Thus, we perform two kinds of transformations regarding stop words:
   – Removing stop words that carry little to no information.
   – Replacing stop words with their alternatives or with a phrases with the same meaning.

3. **Spelling**
   The spelling score of a document is high if there are no spelling mistakes, and low when there are some.
   – To increase the spelling score we apply spelling correction. The spell-checker uses a probability model and a previously mentioned corpus from publicly-available books.
   – To decrease the score, we use a dictionary to insert common mistakes in the text. The aforementioned dictionary was manually created using data from various sources.
4. **Punctuation**
   If the punctuation use is above average, we remove all punctuation used within the sentence. This is limited to the symbols comma (,), semicolon (;) and colon (:) If the punctuation use is bellow average, we apply two techniques to improve that score:
   – We randomly insert comma or semicolon before prepositions. We insert comma with a higher probability then for inserting semicolon.
   – We insert redundant symbols using the following schema:

   ```
   ! can be replaced with one of [!, !!, !!!]
   ? can be replaced with one of [?, ??, ???, ?!?, !?!]
   ```

5. **Word Substitution**
   In order to change the ratio of unique words, we replace the most or the least common words. Replacement is done with synonyms, hypernyms or word descriptions from WordNet [5,15].
   If the document type-token ratio is above average, most words used in the document are randomly replaced with their synonym or hypernym. If the unique words ration in the document is below average, we randomly replace the least used words with their definition from WordNet.
6. **Paraphrase Corpus**
   We randomly replace phrases from the text with their substitutions from a paraphrase corpus. We use the short version of the phrasal corpus of PPDB, or the Paraphrase Database ([6]). This transformation appeared to be very useful for the results. By changing small phrases, the meaning of the text was still preserved, and there was an improvement in changing the metrics for unique work count and parts of speech.
7. **Uppercase Words**
   For decreasing the uppercase words ratio, we only transform words that are all upper case and are longer then three symbols. We assume that if the word is in upper case and is shorter then three symbols, it is an acronym and thus is supposed to stay in uppercase. The transformation is straightforward: all uppercase letters are substituted with lowercase ones.

3.4 Noise

After the transformations are added to mask the author identification features, we apply some transformations that insert some noise in the text.

1. **Switching British and American English**
   Randomly changing words from British to American English and vice versa. The words are taken from a vocabulary.
2. **Inserting random functional words**
   Randomly selected functional words are inserted in the beginning of the sentence. The words are taken from a discourse marker vocabulary.

3.5 General Transformations

We also apply some general transformations that would keep the meaning of the text, but would mask the author style.

1. **Replacing short forms**
   We replace short forms such as *I've, I'd, I'm, I'll, don't, etc.* with their full forms.
2. **Replacing numbers with words**
   We replace the parts of the text, POS-tagged as numbers, with their word representation in English.
3. **Replacing equations**
   As there were some examples of scientific text in the training corpus, if the text contains equations, the operations in them are being replaced with words. The equations are captured if the text contains both comparing and inner equation symbols:

   ```
   ".[<>=]+." and ".[\+\-\*\/]+."
   ```

   The following symbols are being replaced if an equation is found: + *(plus), - (minus), * (multiplied by), / (divided by), = (equals), > (greater than), < (less than), <= (less than or equal to), >= (greater than or equal to).*
4. **Replace symbols and abbreviations with words**
   We replace symbols and abbreviations with their word representations. Such symbols are currency symbols, *% (percent), @ (at)*, abbreviations of person titles (such as *Prof., Mr., Dr.*, etc.).
5. **Simple transformations with regular expressions**
   Possessions are replaced with their short forms:

   ```
    "(\w+) of (\w+)"  is replaced with  "\2’s \1"
   ```

   This will slightly change the stop words rate and also could obfuscate some specific manner of writing as the first method of describing possessions is not commonly used.

3.6 Experiments with Machine Translation

We also experimented with applying machine translation as described in [4]. We translate from English to two languages (Croatian and Estonian) and then back to English. *Microsoft Translation API* is used for the translations.

The measured results in Section 4 show that the transformations we are applying work better for most of the metrics and are comparable to changing the metrics for parts of speech. Manual evaluation of the text obtained with machine translation show that very often the meaning of the obfuscated text differs from the original text.

4 Evaluation and Discussion

There is no adequate metric that could automatically measure the soundness of the text, but we can evaluate how much text metrics had changed after the obfuscation process.

Table 1 shows the average values calculated on the training set and on the books from the Project Gutenberg corpus that we used. For each measure, the average from all the documents in the training set is displayed: before and after obfuscation. The average change rate from all documents is shown, as well as the minimum and the maximum change rate from the documents in the training set.

We have compared the results from the transformations described in Section 3 with machine translation. Table 1 shows the results: average, minimum and maximum change from the custom transformations. Table 2 shows the same measures of the obfuscation with machine translation. We can see that our transformations work better than just applying two-way machine translation for changing the measured indicators of item style. Machine translation has close values for the part of speech ratio and is better than verb ratio.

		Average on Train Set		**Transformations Change**		
Text Metric	**Average**	**Before**	**After**	**Avg.**	**Min**	**Max**
Sentence word count	19	18.42	28.43	136.71%	2.04%	800.85%
Stop words ratio	0.5	0.52	0.45	12.30%	0.63%	28.79%
Type-token ratio	0.44	0.44	0.47	7.32%	0.49%	22.68%
Adjective rate	0.06	0.08	0.09	19.46%	0.27%	73.26%
Adverb rate	0.076	0.07	0.09	28.16%	0.94%	140.00%
Noun rate	0.24	0.23	0.24	9.62%	0.88%	32.28%
Verb rate	0.19	0.20	0.21	5.26%	0.58%	29.04%
Punctuation ratio	0.15	0.14	0.14	48.51%	9.26%	157.68%
Words in all caps ratio	0.02	0.03	0.01	43.43%	0.93%	100.00%

Table 1. Text measures on the training set. Results from obfuscation with transformations. Shown are the values of the used text measures on the training set. Column *Average* shows the calculated average on the training set and the corpus on Project Gutenberg. The next two columns show the average metrics of the train corpus before and after the obfuscation. The last columns show the change rate of the transformations.

	Translations Change		
Text Metric	Average	Min	Max
Sentence word count	4.27%	0.42%	12.14%
Stop words ratio	5.54%	0.50%	15.09%
Type-token ratio	2.50%	0.23%	5.35%
Adjective rate	13.85%	9.25%	19.55%
Adverb rate	10.72%	1.85%	27.13%
Noun rate	4.38%	0.13%	11.55%
Verb rate	7.63%	0.30%	19.63%
Punctuation ratio	28.40%	0.58%	66.88%
Words in all caps ratio	30.54%	0.00%	107.42%

Table 2. Results from obfuscation with multilanguage translation. The results in the columns show the average, the minimum, and the maximum change for the corresponding measure.

Participant	PAN 2013	PAN 2014 EE	PAN 2014 EN	PAN 2015
Mihaylova *et al.* (our system)	-0.10	-0.13	-0.16	-0.11
Keswani *et al.* [12]	-0.09	-0.11	-0.12	-0.06
Mansoorizadeh *et al.* [14]	-0.05	-0.04	-0.03	-0.04

Table 3. Average performance drops in terms of 'final scores' of the authorship verifiers submitted at PAN 2013 to PAN 2015 when run on obfuscated versions of the corresponding test datasets as per the submitted obfuscators. The smaller the number (i.e., the higher the performance drop), the better.

The results from the evaluation of the Author Obfuscation task authors [16] are shown in Table 4. Our system performs best in terms of fooling the state-of-the-art systems that participated in the Author Identification tasks in the previous years.

The metrics that change the most are the average sentence length and the punctuation to word count rate. The metrics whose values changed the least are the rates of the different parts of speech: nouns, verbs and adjectives and unique words.

The soundness of the obfuscated texts was checked manually for randomly selected documents from the corpus. The observations are that after applying the transformations mentioned above, the result text is close to the meaning of the original. The most useful transformations were word replacement using the paraphrase corpus and WordNet. Splitting and merging sentences also contributes to changing the author style. Insertion of random noise - spelling and punctuation mistakes - lowers the quality of the resulting text, but contributes to changing the measures used for author identification.

After the submission, we further checked our results for sensibleness and noticed that some of the transformations were applied too often and resulted in tests of lower quality. These are replacements of numbers, replacement of words with their definitions from WordNet, insertion of too many spelling and punctuation errors. Lowering those transformations improves the quality of the resulting texts.

5 Obfuscation Evaluation

5.1 Evaluation Metrics

For the Obfuscation Evaluation subtask, we provide metrics for safety and soundness.

The metrics used for *safety* measure how much each of the metrics has changed in the obfuscated text compared to the original text. We are giving the metric as follows:

```
|original_value-obfuscated_value| / original_value
```

or 0 if the metric value for the original text is 0. The metrics for safety are the difference in the metrics described in the previous chapters.

One metric for measuring *soundness* is proposed. It measures the semantic similarity between the original and the obfuscated text. We measure this for all of the text parts and we use the average as a metric for the obfuscated document ([13]). The similarity is a measure for each original-obfuscation pair. The similarity between two text parts is measured as a weighted sum of their semantic similarity and word order similarity. The semantic similarity between the two texts is measured as the cosine distance between their semantic vectors. For calculation of the semantic vectors, the union of all words in the two texts is taken. The sentence is passed in as a collection of words. The size of the semantic vector is the same as the size of the joint word set. The elements are 1 if a word in the sentence already exists in the joint word set, or the similarity of the word to the most similar word in the joint word set if it does not. Word order similarity computes the word order vector for a sentence. The sentence is passed in as a sequence of words. The size of the word order vector is the same as the size of the joint word set. The elements of the word order vector are the position mapping (from the windex dictionary) of the word in the joint set if the word exists in the sentence. If the word does not exist in the sentence, then the value of the element is the position of the most similar word in the sentence. However, our experiments showed that this metric works too slowly and we did not include it in the final evaluation of the systems.

5.2 Results from Peer Review

Part of our task was to evaluate the results of the systems for all participants who submitted results for the Author Obfuscation task. This subsection shows our results from this evaluation.

We measure the three dimensions specified in the task: safety, soundness and sensibleness.

Soundness and **sensibleness** are measured manually. We get random samples of original-obfuscation pairs and we count the pairs that correspond to the given metric.

Safety is measured in two ways. The first one is running the system GLAD [8], one of the best systems for author identification from PAN-2015 [18]. The results from the original texts were compared with the results for the obfuscation of each participant.The model was trained on the PAN-2015 corpus for English.

The second way was by using our approach, which measures the change in the metrics mentioned in the previous sections, which are commonly used for author identification: parts of speech ratio, type-token ratio, average sentence length, punctuation rate, rate of words in all capital letters.

The submissions were anonymized for the evaluation. The participant team names were replaced with “Participant A”, “Participant B” and “Participant C”. Our results for each of the participants is shown below.

1. **Participant A: mansoorizadeh:2016** [14]
   - **Safety**
     This participant performs worst in terms of safety. The results from the evaluation with author identification software as well as with custom metrics show that they have the least change of the values output by the evaluations.
   - **Soundness**

Sound	Not Sound	Total
212	13	225

   - **Sensibleness**

Sensible	Not Sensible	Total
195	30	225

   - **Overall thoughts**
     Overall the documents are not obfuscated except for several phrases. The phrases that are switched often do not carry the same meaning and/or are out of context. There are occasional grammatical errors. The obfuscation is sound and sensible, but one can detect the style of the original author.
2. **Participant B: keswani:2016** [12]
   - **Safety**
     This participant performs second best in terms of safety.
   - **Soundness**

Sound	Not Sound	Total
2	200	202

   - **Sensibleness**

Sensible	Not Sensible	Total
1	201	202

   - **Overall thoughts**
     The texts are obfuscated, but almost all of them do not make any sense. The obfuscation is neither sound, nor sensible.
3. **Participant C: mihaylova:2016** (our system)
   - **Safety**
     This participant (i.e., our system) performs best in terms of safety. They achieve the highest scores from both author obfuscation software and custom metrics.
   - **Soundness**

Sound	Not Sound	Total
200	2	202

- **Sensibleness**

Sensible	Not Sensible	Total
100	100	200

- **Overall thoughts**

  The text is safe as the author's style is masked. Most of text fragments preserve the meaning of the original text. Grammatical mistakes and typos are common and one could tell that the text is obfuscated. Some phrases are out of context.

5.3 Results for Safety

We show the results from our evaluation of safety.

Evaluation with author identification software

We ran one of the best-performing systems of the PAN-2015 Author Identification task - GLAD [8]. We trained it on the training set from PAN-2015, then we used as test sets the original texts and the texts obfuscated with each of the given systems for participants A, B and C.

We measured how much the prediction changes for each document and then we get the average, the maximum and the minimum change per participant. The results are given in Table 5.3.

	Participant A	**Participant B**	**Participant C**
Average change	0.14	0.20	0.24
Min change	0.00	0.00	0.00
Max change	0.68	0.82	1.93

Table 4. Results from evaluation with author identification software. Shown are the average, min and max change for each participant.

Evaluation with custom metrics

For this part of the evaluation of safety, we measure how much each of the metrics we use has changed for each of the obfuscated documents. Then we compare the average, min and max change for each participant. Table 5.3 shows the results of this evaluation. Participant A does not have any metric for which they have the max average change. Participant B has a max average change for noun, verb, punctuation rate and unique words rate. Participant C has a max average change for adjectives and adverbs rates, stop words rate, average sentence length and words in all capital letters.

	Participant A	**Participant B**	**Participant C**
Average adjective rate	0.03 0.00 0.35	0.15 0.00 0.96	**0.19** 0.00 0.94
Average adverb rate	0.03 0.00 0.34	0.12 0.00 1.18	**0.29** 0.02 2.57
Average noun rate	0.01 0.00 0.11	**0.25** 0.07 0.59	0.10 0.00 0.27
Average verb rate	0.01 0.00 0.07	**0.20** 0.00 0.45	0.05 0.00 0.27
Average punctuation rate	0.01 0.00 0.04	**1.42** 0.26 6.68	0.49 0.08 1.90
Average sentence length	0.01 0.00 0.08	0.87 0.26 0.99	**1.37** 0.04 8.21
Stop words ratio	0.02 0.00 0.09	0.05 0.00 0.25	**0.12** 0.00 0.28
Unique words ratio	0.01 0.00 0.04	**0.12** 0.02 0.35	0.07 0.00 0.23
Words all capitals ratio	0.02 0.00 0.43	0.29 0.00 4.04	**0.42** 0.00 1.00

Table 5. Results for evaluation with custom metrics. Each cell shows the average, min and max change of the corresponding metric and participant. The best results for the average change are highlighted.

6 Conclusion and Future work

We have described the system of the Sofia University's *mihaylova16* team for the PAN-2016 Author Obfuscation task. Our main approach is based on measuring popular text characteristics used for Author Identification and applying transformations aiming to change those measures for the given text.

Further development includes adding more features used for author identification. The existing transformations should be improved in terms of producing more meaningful text.

The task requirements included splitting the text into smaller parts and applying obfuscation on those parts, and we have implemented transformations suitable for such smaller text parts. We would like to try experiment with transformations that could be applied to the entire text or to paragraphs, which is closer to the way people transform texts.

What our approach is lacking is a proper evaluation measure about whether it performs well in terms of soundness; designing one is a challenging but necessary and enabling research direction.

Finally, we plan to use the techniques used in this paper for author imitation. One key difference will be that the goal for the transformations should not be the average metrics, but the metrics of the author that should be imitated.

Acknowledgments

This research was performed by a team of students from MSc programs in Computer Science in the Sofia University "St Kliment Ohridski".

We thank the Sofia University "St Kliment Ohridski" for the support and guidance to our team participation at the CLEF 2016 Conference.

References

1. Afroz, S., Brennan, M., Greenstadt, R.: Detecting hoaxes, frauds, and deception in writing style online. In: Proceedings of the 2012 IEEE Symposium on Security and Privacy. pp. 461–475. SP '12, IEEE Computer Society, Washington, DC, USA (2012)
2. Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (eds.): CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal. CEUR Workshop Proceedings, CEUR-WS.org (2016)
3. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media (2009)
4. Brennan, M., Afroz, S., Greenstadt, R.: Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. ACM Trans. Inf. Syst. Secur. 15(3), 12:1–12:22 (Nov 2012)
5. Fellbaum, C.: WordNet: An Electronic Lexical Database. Bradford Books (1998)
6. Ganitkevitch, J., Van Durme, B., Callison-Burch, C.: PPDB: The paraphrase database. In: Proceedings of NAACL-HLT. pp. 758–764. Atlanta, Georgia (June 2013)
7. Hart, M.: Project gutenberg. Project Gutenberg (1971)
8. Hürlimann and Benno Weck and Esther van den Berg and Simon Suster and Malvina Nissim, M.: Glad: Groningen lightweight authorship detection. In: CLEF (2015)

9. Juola, P.: Detecting stylistic deception. In: Proceedings of the Workshop on Computational Approaches to Deception Detection. pp. 91–96. Avignon, France (2012)
10. Juola, P., Vescovi, D.: Advances in Digital Forensics VII: 7th IFIP WG 11.9 International Conference on Digital Forensic, chap. Analyzing Stylometric Approaches to Author Obfuscation, pp. 115–125. Orlando, FL, USA (2011)
11. Kacmarcik, G., Gamon, M.: Obfuscating document stylometry to preserve author anonymity. In: Proceedings of COLING/ACL: Poster Sessions. pp. 444–451. Sydney, Australia (2006)
12. Keswani, Y., Trivedi, H., Mehta, P., Majumder, P.: Author Masking through Translation—Notebook for PAN at CLEF 2016. In: Balog et al. [2]
13. Li, Y., McLean, D., Bandar, Z.A., O'Shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. on Knowl. and Data Eng. 18(8), 1138–1150 (2006)
14. Mansoorizadeh, M., Rahgooy, T., Aminiyan, M., Eskandari, M.: Author Obfuscation using WordNet and Language Models—Notebook for PAN at CLEF 2016. In: Balog et al. [2]
15. Miller, G.A.: WordNet: A lexical database for English. Commun. ACM 38(11), 39–41 (1995)
16. Potthast, M., Hagen, M., Stein, B.: Author Obfuscation: Attacking State-of-the-Art Authorship Verification Approaches. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CLEF and CEUR-WS.org (2016)
17. Quirk, C., Brockett, C., Dolan, W.: Monolingual machine translation for paraphrase generation. In: Proceedings of EMNLP 2004. pp. 142–149. Barcelona, Spain (2004)
18. Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., Stein, B.: Overview of the author identification task at PAN 2015. In: CLEF (2015)

A Appendix 1 - Project Gutenberg books

- The Adventures of Sherlock Holmes by Sir Arthur Conan Doyle
- History of the United States by Charles A. Beard and Mary R. Beard
- Manual of Surgery Volume First: General Surgery by Alexis Thomson and Alexander Miles. Sixth Edition.
- War and Peace, by Leo Tolstoy