Author Masking using Sequence-to-Sequence Models

Notebook for PAN at CLEF 2017

Oleg Bakhteev and Andrey Khazov

Antiplagiat CJSC, Moscow, Russia;
Moscow Institute of Physics and Technology (MIPT), Moscow, Russia
bahteev@ap-team.ru
Antiplagiat CJSC , Moscow, Russia;
hazov@ap-team.ru

Abstract The paper describes the approach adopted for Author Masking Task at PAN 2017. For the purpose of masking the original author, we use the combination of methods based either on deep learning approach or traditional methods of obfuscation. We obtain sample of obfuscated sentences from original one and choose best of them using language model. We try to change both the content and length of original sentence preserving its meaning.

1 Introduction & Related Work

PAN 2017 [16] is a series of tasks on digital text forensics, which is held as a part of the CLEF conference [8]. The main idea of one of the proposed tasks named author masking task [5] is to paraphrase a given document so that its writing style does not match that of its original author, anymore. Training corpus consists of set of documents from the same author. One of these documents should be obfuscated. Quality of suggested software is verified by following metrics:

- *safety* — a forensic analysis does not reveal the original author of its obfuscated texts,
- *soundness* — obfuscated texts are textually entailed with their originals,
- *sensibleness* — obfuscated texts are inconspicuous.

The related tasks that were proposed at PAN 2017 are author identification [21] and author profiling [18]. The evaluation of all the tasks is conducted using TIRA [14], a service for data analysis tasks evaluation.

On PAN'16 conference [15] in "Author Obfuscation" task participants proposed three different ways for author masking. The first approach consists of translation text from the source language (English) into an intermediate language before it gets eventually translated back to English [9]. The main advantage of this method is a strong modification of the original text, the main disadvantages — a vast amount of untranslated words and weak semantic coherence of the resulted text. The second approach used in [11] is to synonymize the most frequent words of original text. This approach keeps the original meaning of the text in most of cases, but gives a small amount of

modifications of the original text. The third approach combines strong context modification with preserving the original sense [12]. This algorithm is based on different types of text obfuscation and gave the best result by the metrics used in the contest.

Statistical and context features are used in modern detecting authorship approaches, for example in GLAD [7]. In our solution we try to obfuscate both of them. We use both traditional methods for author masking, such as synonimizing and splitting/joining sentences and obtain some modern methods based on recurrent neural networks. Using deep neural networks we took into account the papers [13,19,22,4,17,20] on the use of recurrent neural networks in paraphrase generation and detection. We use LSTM-based model [20] in Encoder-Decoder fashion.

2 Proposed Approach

Our approach is based on per-sentence obfuscation. At the first step we split text into sentences. After that we try to paraphrase sentences using methods described below. We paraphrase each sentence until Jaccard similarity score between set of tokens from an original $\mathbf{s}_{\text{original}}$ sentence and an obfuscated $\mathbf{s}_{\text{obfuscated}}$ sentence is less than threshold θ or unless we tested all the obfuscation methods for the original sentence:

$$J(\mathbf{s}_{\text{original}}, \mathbf{s}_{\text{obfuscated}}) = \frac{\left|\mathbf{s}_{\text{original}} \cap \mathbf{s}_{\text{obfuscated}}\right|}{\left|\mathbf{s}_{\text{original}} \cup \mathbf{s}_{\text{obfuscated}}\right|} \leq \theta. \tag{1}$$

All of the described obfuscation methods works with one or two sentences. Priority of using obfuscation methods is based on statistics of its previous successful appliance — we try to make the distribution of methods usage close to uniform since different methods of obfuscation can mask different style features of the original text. Therefore infrequently used approaches apply first for new sentences.

The methods we use to obfuscate sentences can be divided into 2 groups:

1. Methods that change the content of the sentences, trying to save the sense.
2. Methods that change the structure and length of the sentences.

2.1 Changing the Structure and Length of the Original Text

We use different types of changing sentences length. As a part of preprocessing, we replace short forms for long ones: words ended with *'ll, 've, 'm, etc.* replaces with their long forms — *will, have, am, etc*.

Our main approach of changing text length is to split and join sentences. As a trigger of splitting we use rather simple heuristic: we try to split sentences by coordinating (*and, but*) and subordinating (*because, since, so, therefore*) conjunctions. As a method of joining sentences we use the following rule: we can join sentences using the same conjunctions if both sentence have rather small length, we use range between 30 and 150 chars for this constraint.

The third method we used is an adjustment or removal introductory phrases from sentences. We use only general meaning phrases such as *it is important to note that, anyway, in fact, also, etc.*

2.2 Changing Content of the Original Text

We use two methods of changing content of the sentence.

Synonym replacing. First method is based on traditional synonimizing idea, where some words of the input sentence are replaced by their synonyms. However, instead of using existing dictionaries or ontologies we use word embedding as a source of synonimizing. We generate subsample set of k different combinations from nearest words lists and take best of generated sentence by the language model score.

Let $(\mathbf{w}_1, \ldots, \mathbf{w}_n)$ be a sequence of word embeddings from the sentence. For each word $\mathbf{w}_i$ except stopwords we take k nearest words by cosine similarity: $\mathbf{v}_i = (\mathbf{w}_{i1}^{'}, \ldots, \mathbf{w}_{ik}^{'})$. We generate s sentences $\mathbf{s}_1, \ldots, \mathbf{s}_s$ sampling from $\mathbf{v}_i$ words instead of original word $\mathbf{w}_i$. After that we find the sampled sentence with the maximal language model score:

$$\mathbf{s}_{\text{obfuscated}} = \underset{\mathbf{s} \in \{\mathbf{s}_1, \ldots, \mathbf{s}_s\}}{\arg\max} \text{LM}(\mathbf{s}), \tag{2}$$

where LM is a logarithm of language model probability [10].

For our experiments we used $k = 5$ and $s = 100$. The language model was trained on 3-grams from Shakespeare's Sonnets corpus from Project Gutenberg [1]. In our opinion the original author style will be masked because this procedure gives best scores for sentences, nearest to Shakespeare style. We did not use language model of higher order because of small size of the corpus.

Encoder-Decoder approach. Another method is based on LSTM recurrent neural network. The basic LSTM model can be described with the following equations:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\boldsymbol{\theta}_{xi}\mathbf{x}_t + \boldsymbol{\theta}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i), \\
\mathbf{f}_t &= \sigma(\boldsymbol{\theta}_{xf}\mathbf{x}_t + \boldsymbol{\theta}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{o}_t &= \sigma(\boldsymbol{\theta}_{xo}\mathbf{x}_t + \boldsymbol{\theta}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o), \\
\mathbf{c}_{in} &= \tanh(\boldsymbol{\theta}_{xc}\mathbf{x}_t + \boldsymbol{\theta}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c), \\
\mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{c}_{in}, \\
\mathbf{h}_t &= \mathbf{o}_t \circ \tanh(\mathbf{c}_t),
\end{aligned} \tag{3}$$

$$\mathbf{g}(\mathbf{h}_{t-1}, \mathbf{w}_{t-1}, \mathbf{c}_{t-1}) = \mathbf{h}_t. \tag{4}$$

We train our model in Encoder-Decoder way [20,19] with modification of LSTM described in [20]: we decompose our model into Encoder model and Decoder model.

Encoder recursively combines the sequence of word embeddings $\mathbf{w}_1, \ldots, \mathbf{w}_n$ into a fixed-length vector $\mathbf{h}_{n-1}$:

$$\mathbf{h}_t = \mathbf{g}_e(\mathbf{h}^e_{t-1}, \mathbf{w}_{t-1}, \mathbf{c}^e_{t-1}),$$

where $\mathbf{g}_e$ is a stack of LSTM functions, $\mathbf{h}^e_{t-1}$ is a hidden state, $\mathbf{c}^e_{t-1}$ is a cell state vector.

Decoder tries to reproduce the input sequence $\mathbf{w}_1, \ldots, \mathbf{w}_n$ by hidden vector sequence $\mathbf{h}^e_{n-1}, \ldots, \mathbf{h}^e_1$ and vector $\mathbf{c}$:

$$\hat{\mathbf{w}}_t = \mathbf{f}_d(\mathbf{h}^d_{t-1}, \hat{\mathbf{w}}_{t-1}, \mathbf{c}^d_{t-1}, \mathbf{c}),$$

where $\mathbf{g}_d$ is a stack of LSTM functions, $\mathbf{h}^d_{t-1}$ is a hidden state, $\mathbf{c}^d_{t-1}$ is a cell state vector, $\mathbf{c}$ is a cell state vector from the last step of the encoder.

Encoder and Decoder models are jointly trained in order to minimize reconstruction error:

$$\sum_{i=1}^{n} ||\mathbf{w}_i - \hat{\mathbf{w}}_i||^2.$$

For the end of sentence determination we added *"End of sentence"* token to our embedding model so that in general the length of our original sentence $\mathbf{s}_{\text{original}}$ and the obfuscated sentence $\mathbf{s}_{\text{obfuscated}}$ may differ. Further we use reproduced sequence $\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{n_o}$ the same way as we use in our *synonym replacing* approach (2), where n_o is the number of tokens before *"End of sentence"* token.

2.3 Evaluation

We considered two automatic metrics for evaluating final obfuscation. For the *sensibleness* evaluation we used average language model score from KenLM language model [6]. The language model from our obfuscation method differs from the model we use for evaluation: whenever we used model trained on Shakespeare corpus for obfuscation, the model for the evaluation was trained on Wikipedia corpus. Therefore despite the fact we tried to mask the original author style using Shakespeare style, during the evaluation step we considered how the obfuscated text fitted into common English language.

For the *safety* evaluation we used the similar method as described in [12]: we measured how much the prediction from GLAD [7] author verification system changed. We used random forest classifier in GLAD.

We did not consider any automatic metric for the *soundness* and used peer review.

3 Experiment Details and Results

On preprocessing step we used the NLTK toolbox [2] to extract separate sentences from the original text. We used FastText library [3] for word embedding. Our model was trained on the latest dump of Wikipedia corpus, with word vector dimension equal to 300. For the recurrent neural network training we used Seq2Seq library[1] also trained on Wikipedia corpus. Based on peer review we set $\theta = 0.75$ in (1). We used 2-layer LSTM as it showed better results than 1-layer model.

Our average language model score for *sensibleness* was -99.4 ± 61.9 whenever the score for the original sentences was -79.4 ± 55.8. As we can see, the scores are rather close since the means of distributions lie in the range of the standard deviations of each other.

The average change in GLAD probabilities is -0.11 ± 0.22. The number of correctly verified texts was lowered after obfuscation from 189 to 153. We observe that our obfuscation method works successfully and lowers the verification probabilities for the obfuscated texts.

[1] https://github.com/farizrahman4u/seq2seq

An example of our obfuscation method is listed in table 3. As we can see, the obfuscated sentences obtained by Encoder-Decoder can lead to some grammatical errors. However, the significant part of the sentences we viewed was grammatically correct. The other interesting feature of the sentences with synonym replacement and Encoder-Decoder method is an appearance of word “scabbard” in obfuscated sentences. We consider it is a result of using Shakespeare corpus in the final sentence scoring (2).

Table 1. An example of our method usage

Method	**Obfuscated sentences**
Original	The quick brown fox jumps over the lazy dog. The five boxing wizards jump quickly with knifes.
Synonym replacing	The rapid reddish fox grabs over the sloppy terrier. The five boxer spellcasters jumper quickly with scabbard.
Encoder-Decoder	The better brown fox tosses overlapped in scary pig even. The Seven boxing superhero trampolining eventually with scabbard.
Introductory words	All in all, the quick brown fox jumps over the lazy dog. In a word, the five boxing wizards jump quickly with knifes.
Join sentences	The quick brown fox jumps over the lazy dog, because the five boxing wizards jump quickly with knifes.

4 Conclusion

The paper describes our system for the PAN 2017 Author Masking Task. Our main approach based on using recurrent neural networks for text obfuscation. Also we use more traditional methods of obfuscation, such as synonimizing and changing statistical text features. We used language model for selection best masking result.

Further development includes improving obfuscation quality of seq2seq model by tuning its parameters and taking into consideration many other heuristics.

References

1. Project Gutenberg. http://www.gutenberg.org/wiki/Main_Page, http://www.gutenberg.org/wiki/Main_Page
2. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O’Reilly Media, Inc." (2009)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349 (2015)
5. Hagen, M., Potthast, M., Stein, B.: In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes Papers of the CLEF 2017 Evaluation Labs
6. Heafield, K.: KenLM: faster and smaller language model queries. In: Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation. pp. 187–197. Edinburgh, Scotland, United Kingdom (July 2011), https://kheafield.com/papers/avenue/kenlm.pdf

7. Hürlimann, M., Weck, B., van den Berg, E., Šuster, S., Nissim, M.: GLAD: Groningen Lightweight Authorship Detection—Notebook for PAN at CLEF 2015. In: Cappellato, L., Ferro, N., Jones, G., San Juan, E. (eds.) CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers, 8-11 September, Toulouse, France. CEUR-WS.org (Sep 2015)
8. Jones, G.J.F., Lawless, S., Gonzalo, J., Kelly, L., Goeuriot, L., Mandl, T., Cappellato, L., Ferro, N. (eds.): Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017, Proceedings
9. Keswani, Y., Trivedi, H., Mehta, P., Majumder, P.: Author Masking through Translation—Notebook for PAN at CLEF 2016. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (eds.) CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal. CEUR-WS.org (Sep 2016), http://ceur-ws.org/Vol-1609/
10. Koehn, P.: Statistical Machine Translation. Cambridge University Press, New York, NY, USA, 1st edn. (2010)
11. Mansoorizadeh, M., Rahgooy, T., Aminiyan, M., Eskandari, M.: Author Obfuscation using WordNet and Language Models—Notebook for PAN at CLEF 2016. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (eds.) CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal. CEUR-WS.org (Sep 2016), http://ceur-ws.org/Vol-1609/
12. Mihaylova, T., Karadjov, G., Nakov, P., Kiprov, Y., Georgiev, G., Koychev, I.: SU@PAN'2016: Author Obfuscation—Notebook for PAN at CLEF 2016. In: Balog, K., Cappellato, L., Ferro, N., Macdonald, C. (eds.) CLEF 2016 Evaluation Labs and Workshop – Working Notes Papers, 5-8 September, Évora, Portugal. CEUR-WS.org (Sep 2016), http://ceur-ws.org/Vol-1609/
13. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 2786–2792. AAAI Press (2016)
14. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014)
15. Potthast, M., Hagen, M., Stein, B.: Author Obfuscation: Attacking the State of the Art in Authorship Verification. In: Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings, CLEF and CEUR-WS.org (Sep 2016), http://ceur-ws.org/Vol-1609/
16. Potthast, M., Rangel, F., Tschuggnall, M., Stamatatos, E., Rosso, P., Stein, B.: Overview of PAN'17: Author Identification, Author Profiling, and Author Obfuscation. In: Jones, G., Lawless, S., Gonzalo, J., Kelly, L., Goeuriot, L., Mandl, T., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 8th International Conference of the CLEF Initiative (CLEF 17). Springer, Berlin Heidelberg New York (Sep 2017)
17. Prakash, A., Hasan, S.A., Lee, K., Datla, V., Qadir, A., Liu, J., Farri, O.: Neural paraphrase generation with stacked residual lstm networks. arXiv preprint arXiv:1610.03098 (2016)
18. Rangel, F., Rosso, P., Potthast, M., Stein, B.: In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes Papers of the CLEF 2017 Evaluation Labs
19. Socher, R., Huang, E.H., Pennin, J., Manning, C.D., Ng, A.Y.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information

Processing Systems 24, pp. 801–809. Curran Associates, Inc. (2011), http://papers.nips.cc/paper/4204-dynamic-pooling-and-unfolding-recursive-autoencoders-for-paraphrase-detection.pdf
20. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
21. Tschuggnall, M., Stamatatos, E., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes Papers of the CLEF 2017 Evaluation Labs
22. Wieting, J., Bansal, M., Gimpel, K., Livescu, K.: Towards universal paraphrastic sentence embeddings. CoRR abs/1511.08198 (2015), http://arxiv.org/abs/1511.08198