CrossLang: the system of cross-lingual plagiarism detection

Oleg Bakhteev
Moscow Institute of Physics and Technology
Moscow, Russia
bakhteev@phystech.edu

Alexandr Ogaltsov
Higher School of Economics
Moscow, Russia
aogalcov@hse.ru

Andrey Khazov
Antiplagiat Company
Moscow, Russia
hazov@ap-team.ru

Kamil Safin
Moscow Institute of Physics and Technology
Moscow, Russia
kamil.safin@phystech.edu

Rita Kuznetsova
IBM Research
Rueschlikon, Switzerland
kuz@zurich.ibm.com

Abstract

Plagiarism and text reuse become more available with the Internet development. Therefore it is important to check scientific papers for the fact of cheating, especially in Academia. Existing systems of plagiarism detection show the good performance and have a huge source databases. Thus now it is not enough just to copy the text "as is" from the source document to get the "original" work. Therefore, another type of plagiarism become popular — cross-lingual plagiarism. We present a CrossLang system for such kind of plagiarism detection for English-Russian language pair.

1 CrossLang design

The key idea for CrossLang[1] system is that we use the monolingual approach. We have suspicious Russian document and English reference collection. We reduce the task to the one language — we translate the suspicious document into English, because the reference collection is in English. After this step we perform the subsequent document analysis. Due to this fact the main challenge with the CrossLang design is that the algorithms should be stable to the translation ambiguity. The main stages of CrossLang service is depicted in Figure 1 . CrossLang receives the suspicious document from Antiplagiat system, when user send it for originality checking. Then it goes to *Entry point* — main service, that routes the data between following stages:

1. *Machine Translation system* — microservice, that translates suspicious document into English. For these purposes we use Transformer Vaswani et al., open-source neural machine translation framework.
2. *Source retrieval* — this stage unites two microservices: *Shingle index* and *Document storage*. Entry point receives the translated suspicious document's shingles (n -grams) and Shingle index returns to it the documents ids from the reference English collection. To deal with the

[1] This work was supported by RFBR project No.18-07-01441 and FASIE project No.44116.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

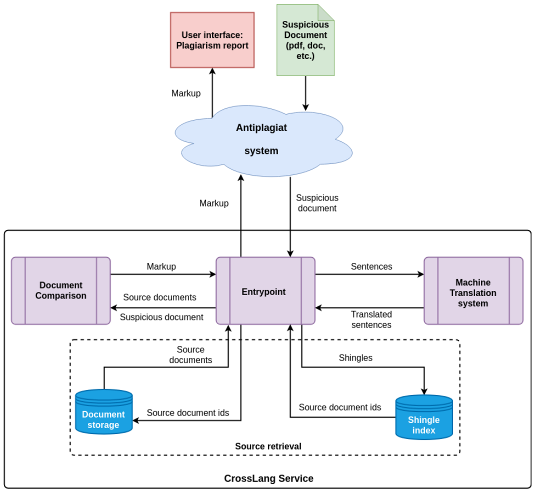

Figure 1: CrossLang service design.

translation ambiguity we use modified shingle-based approach. Document storage returns the Source texts from the collection by these ids.

3. *Document comparison* — this microservice performs the comparison between translated suspicious document and source documents. We compare not the texts themselves, but the vectors corresponding to the phrases of these texts. Thus we deal with the translation ambiguity problem.

1.1 Machine Translation system

We create machine translation system using state-of-the-art Transformer algorithm Vaswani et al.. We utilize Tensorflow realization [2] of it. Training dataset consists of approximately 30M parallel sentences. They were obtained from open-source parallel OPUS Tiedemann [2012] corpora, but also we mine parallel sentences from Common Crawl.[3] We evaluate BLEU score Papineni et al. [2002] for *Russian* → *English* translation on news test 2018 dataset [4] and compare it with Google translator via API [5]. Results are in Table 1.

Table 1: BLEU of different systems

System	BLEU
Google	31.34
CrossLang Transformer	28.18

The CrossLang BLEU score lower than Google's BLEU score — this was to be expected. But it is very important to notice that we are not interested in ideal translation. Our main goal is to translate with sufficient quality for the next stages: Source retrieval and Document comparison.

1.2 Source retrieval

The method of source retrieval in the case of verbatim plagiarism is inverted index construction,where a document from the reference collection is represented as a set of its shingles, i.e. overlapping word n -grams, and a suspicious document's shingles are checked for matches with the indexed documents. There is one major problem with using the standard shingles — in our case the machine translation stage generates texts that differ too much from the sources of plagiarism. We argue that the source

[2] https://tensorflow.github.io/tensor2tensor/

[3] http://commoncrawl.org/

[4] http://www.statmt.org/wmt18/translation-task.html

[5] https://cloud.google.com/translate/docs/

retrieval task can be solved with the help of a similar method that performs better than the method mentioned above; this improvement is achieved by moving from word shingles to word-class shingles, where each word is substituted by the label of the class it belongs to:

$$\{\text{word}_1, \dots, \text{word}_n\} \to \{\text{class}(\text{word}_1), \dots, \text{class}(\text{word}_n)\}.$$

Clustering the word vectors is a convenient and relatively fast way of obtaining semantic word classes. For the word embedding model we used `fastText` Bojanowski et al. [2016] trained on English Wikipedia. The dimension for word embedding model was set to 100. For the semantic word classes construction we applied agglomerative clustering on word embeddings with the cosine similarity measure to group words into word classes. We got 777K words clustered into 30K classes.

1.3 Document Comparison

For the comparison between retrieved documents and translated suspicious documents we introduce the phrase embedding model. We split documents (retrieved and suspicious) into phrases s and compare its vectors. For mapping the word sequence into low dimensional space we use the encoder-decoder scheme with L-2 reconstruction error minimization $E_{rec} = \| \mathbf{s} - \hat{\mathbf{s}} \|^2$. Encoder-decoder model is completely unsupervised and does not use any information whether the phrase pair is paraphrased or not. We train Seq2Seq model with attention Bahdanau et al. [2014] on 10M sentences from Wikipedia. In order to use information about phrase similarity we extend the objective function. We employ the margin-base loss from Wieting et al. [2015] with the limited number of similar phrase pairs $\mathcal{S} = \{(s_i, s_j)\}$:

$$E_{me} = \frac{1}{|\mathcal{S}|} \left(\sum_{(s_i, s_j) \in \mathcal{S}} \max(0, \delta - c_-) + \max(0, \delta - c_+) \right), \tag{1}$$

where $c_- = \cos(\mathbf{s}_i, \mathbf{s}_j) - \cos(\mathbf{s}_i, \mathbf{s}_{i'})$, $c_+ = \cos(\mathbf{s}_i, \mathbf{s}_j) + \cos(\mathbf{s}_j, \mathbf{s}_{j'})$, δ is the margin,
$\mathbf{s}_{i'} = \arg\max_{\mathbf{s}_{i'} \in \mathcal{S}_b \setminus (\mathbf{s}_i, \mathbf{s}_j)} \cos(\mathbf{s}_i, \mathbf{s}_{i'})$, $\mathcal{S}_b \in \mathcal{S}$ — current mini-batch.

The sampling of so named "false neighbour" $\mathbf{s}_{i'}$ during training helps to improve the final quality without strict limitations on what phrases we should use at dissimilar. This part of objective requires a dataset of similar sentences $\mathcal{S} = \{(s_i, s_j)\}$. We used double translation method as a method of similar sentences generation comparable to paraphrase. The final objective function is:

$$\alpha E_{rec} + (1 - \alpha) E_{me}, \tag{2}$$

where α is a tunable hyperparameter that weights both of errors. [6]. For each phrase embedding from the suspicious document find nearest vectors by cosine similarity from source documents using *Annoy*[7] library.

2 Main contributions

1. The best of our knowledge it is the first system for cross-lingual plagiarism detection for English-Russian language pair. It is deployed on production and we could analyze the results. We could not find another examples of such system (even for other language pairs).
2. The Source retrieval 1.2 stage is often employed using rather simple heuristical algorithms such as shingle-based search or keyword extraction. However, these methods can significantly suffer from word replacements and usually detect only near-duplicate paraphrase. We present modified method, see 1.2.
3. Many articles on the cross-lingual plagiarism detection topic investigate the solutions based on bilingual or monolingual word embeddings Ferrero et al. [2017] for documents comparison, but almost none of them uses the phrase embeddings for this problem solution. We present phrase embeddings comparison in 1.3.

[6] The objective (2) had the following value: $\alpha = 0.1.$. In the objective 1 $\delta = 0.3$

[7] https://github.com/spotify/annoy

3 Experiment

There are no results and datasets for cross-lingual plagiarism detection task for language pair English-Russian. We create dataset for the problem and make it available. Visit [8] for dataset download and details about generation. For the whole framework we got Precision $= 0.83$, Recall $= 0.79$ and $F1 = 0.80$.

Since our system translates the suspicious document into the language of the collection it's natural to analyze the performance of our system for monolingual problem. For such experiment we do not use the machine translation service. In order to check performance of monolingual paraphrased plagiarism detection we exploit PAN'11 contest dataset and quality metrics Potthast et al.. Results of CrossLang and top-3 known previous methods are in Table 2.

Table 2: PAN'11 performance comparison

Model	P	R	F	Plagdet
CrossLang	**0.94**	**0.76**	**0.84**	**0.83**
PDLK Abdi et al. [2015]	0.90	0.70	0.79	0.79
Sys-1 Wang et al. [2013]	0.86	0.69	0.76	0.75
Sys-2 Grozea et al. [2009]	0.75	0.66	0.7	0.69

4 Architecture

Our service is deployable on an 8-GPU cluster with Tesla-K100 GPUs, 128GB RAM and 64 CPU Cores. Depending on the requirements, the service is able to scale horizontally. For the fast rescaling we use Docker containerization and Consul and Consul-template for the service discovery and automatic load balancing. The stress testing of our system showed that the system is able to check up to 100 documents in a minute. Despite the fact the average loading on our service is much lower, this characteristic of our service is important for withstanding peak loads.

5 Conclusion

We introduced CrossLang — a framework for cross-lingual plagiarism detection for English Russian language pair. We decomposed the problem of cross-lingual plagiarism detection into several stages and provide a service, consists of a set of microservices. The CrossLang use a monolingual approach — reducing the problem to the one language. For this purpose we trained the neural machine translation system. Another two main algoithmic components are Source Retrieval and Document Comparison stages. For the Source Retrieval problem we used a modification of shingling method that allow us to deal with ambiguity after translation. For the Document Comparison stage we used phrase embeddings that were trained with slight supervision. We evaluated the effectiveness of main stages.

References

A. Abdi, N. Idris, R. Alguliyev, and R. Aliguliyev. Pdlk: Plagiarism detection using linguistic knowledge. *Expert Systems with Applications*, 07 2015.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

J. Ferrero, F. Agnès, L. Besacier, and D. Schwab. Using word embedding for cross-language plagiarism detection. In *EACL 2017*, volume 2, pages 415–421, 2017.

[8] http://tiny.cc/cl_ru_en

C. Grozea, C. Gehl, and M. Popescu. Encoplot: Pairwise sequence matching in linear time applied to plagiarism detection. *3rd PAN Workshop. Uncovering Plagiarism, Authorship and Social Software Misuse*, 502:10, 01 2009.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318, 2002.

M. Potthast, A. Eiselt, A. Barrón-cedeño, B. Stein, and P. Rosso. Overview of the 3rd international competition on plagiarism detection. In *In Working Notes Papers of the CLEF 2011 Evaluation*.

J. Tiedemann. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, 2012.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*.

S. Wang, H. Qi, L. Kong, and C. Nu. Combination of vsm and jaccard coefficient for external plagiarism detection. In *2013 International Conference on Machine Learning and Cybernetics*, volume 04, pages 1880–1885, 2013.

J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198, 2015.