TauJud: Test Augmentation of Machine Learning in Judicial Documents

Zichen Guo
Jiawei Liu
Tieke He*
Zhuoyang Li
Peitian Zhangzhu
State Key Laboratory for Novel Software Technology, Nanjing University, China

ABSTRACT

The booming of big data makes the adoption of machine learning ubiquitous in the legal field. As we all know, a large amount of test data can better reflect the performance of the model, so the test data must be naturally expanded. In order to solve the high cost problem of labeling data in natural language processing, people in the industry have improved the performance of text classification tasks through simple data amplification techniques. However, the data amplification requirements in the judgment documents are interpretable and logical, as observed from CAIL2018 test data with over 200, 000 judicial documents. Therefore, we have designed a test augmentation tool called TauJud specifically for generating more effective test data with uniform distribution over time and location for model evaluation and save time in marking data. The demo can be found at https://github.com/governormars/TauJud.

CCS CONCEPTS

• **Software and its engineering** → **Software testing and debugging**.

KEYWORDS

Test Augmentation, Machine Learning, Judicial Documents

ACM Reference Format:
Zichen Guo, Jiawei Liu, Tieke He, Zhuoyang Li, and Peitian Zhangzhu. 2020. TauJud: Test Augmentation of Machine Learning in Judicial Documents. In *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '20), July 18–22, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3395363.3404364

1 INTRODUCTION

The development of judicial research in the era of big data has produced many applications[5]. Among those applications, the use of machine learning algorithms combined with referee paper data to predict the case to assist trials has become the current focus[10]. Benefited from big data mining, employing machine learning techniques to construct a judicial prediction model is increasingly commonplace in the legal field[4]. These models need to be deployed in the big data environment to process continual evaluations. Not only the algorithms chosen in the judicial prediction model, but also the training set of the judgment documents used as the input affect the performance of the model.

The complexity of machine learning systems makes it difficult for developers to understand the details of system decisions, which has led developers to measure the quality of the system only through test results[6]. Therefore, the development of machine learning systems will rely more on testing than traditional software system development[1]. However, as a critical role in the testing phase, test data does not guarantee its quality, let alone test the quality of intelligent software systems. A high-quality case screening system test set should cover all possible classification results without bias. In contrast, there are specific problems with the existing judicial data, so data augmentation is required for the judgment documents. On the one hand, the larger the scale of the testing data, the higher the accuracy of the prediction model. On the other hand, due to the significant imbalance in the distribution of judicial cases, the data of the judgment documents in some unpopular cases is too small to meet the needs of the machine learning prediction model.

Many researchers have tried many ways to augment the text data[11]. To solve the problem of the high cost of manually labeling data in natural language processing, Jason Wei et al. present a tool called EDA[8]. In order to improve the performance of text classification tasks, simple data augmentation techniques are needed. EDA can help models become more stable and robust with augmentation of testing data. However, it is not applicable to test augmentation in judicial documents that need high interpretability and logic for judgment. It is generally known that a more massive amount of test data can better reflect the real performance of models, so natural augmentation on test data is necessary.

Another popular model called QANet generates new training data by translating sentences into French and back into English[12]. Moreover, synonym replacement is also an ideal way to augment judicial documents. Based on the context of judicial cases, we design particular synonyms of legal terms as corpus. Fairness in machine learning raises concern nowadays[2]. It is an essential requirement of the legal system, where judicial injustice will severely damage the credibility of the relevant departments and organizations. Garg et al .[3] proposed blindness and counterfactual augmentation to evaluate the model's fairness.

*Corresponding author: hetieke@gmail.com

ISSTA '20, July 18–22, 2020, Virtual Event, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8008-9/20/07...$15.00
https://doi.org/10.1145/3395363.3404364

Accordingly, we design TauJud especially for test augmentation in judicial documents, which can generate more effective test data for model evaluation and save time for labeling data. By combining Universal Augmentation Methods with judicial augmentation, we developed a tool for data amplification based on a 200,000 referee paper data. By using TauJud, it will be able to solve the amplification problem of the judgment document data, and then improve the accuracy of the current judicial prediction model.

2 APPROACH

The subject of the test amplification in this article is the referee instrument. As the carrier of the facts of the trial of the case, the judgment documents are described in natural language and expressed in Chinese.

The judicial judgment documents take an important part in the judicial trial and contains the truth about the trial case. Judicial documents have the characteristics of complex format and large amounts of redundancy, which results in much cost of manpower and material resources in the screening of documents. When the judges are dealing with the legal cases, they need to match the referee documents with similar cases in the judicial database for reference.

In judicial prediction, word vectors are often used to parse Chinese text. In this case, there will be more redundant data (usually stop words), which not only causes an unnecessary increase in the volume of the data set, but also leads to the chance of information irrelevant with the trial to participate in decision-making, thus affecting the quality of the model. From the perspective of machine learning, a well-performing model should allow more *important information* to participate in decision-making, while *irrelevant information* does not exist or as little as possible participate in decision-making. Therefore, the removal of stop words and the clipping of text are important operations. In addition, in order to obtain better test augmentation, we hope to be able to transform the vector model. By iterating through the Encoder-Decoder method, the data can complete the transformation without changing its information characteristics. Since this method is often used in machine translation, it is also called back translation.

We divided the *important information* mentioned above into *sensitive attributes* and *non-sensitive attributes*. From a judicial perspective, sensitive attributes describe the basic information of the case but are not relevant to the trial. For example, gender and race are sensitive attributes. In order to make the uneven distribution of sensitive attributes not affect or fairly affect the effect of the dataset, we use the blind method and counterfactual method. Synonym replacement is a common text augmentation method that is classified into a machine learning perspective. However, there are strict requirements for the expression of words in the judicial field, so the scope of synonym mentioned is well-designed.

As can be seen from Fig.1, we provide the framework of TauJud and its three main components. TauJud consists of three main components and add-ons in high-level design. It takes judicial test data as input, and the cloud server is on standby for calling corresponding services. The first component, called universal augmentation, is responsible for the basic methods of text augmentation on machine learning, including stop words removing, back translation and clipping. After this, the second component called judicial augmentation is specifically designed for judicial documents. After two levels of augmentation, TauJud can generate reports of relevant information on test data before and after augmentation. In the following of this section, we describe TauJud's components in detail.

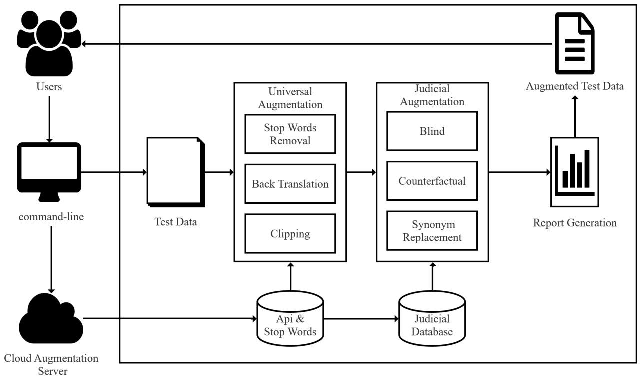

Figure 1: TauJud Workflow

2.1 Universal Augmentation

- **Stop Words Removal**. Stop words are words filtered out before processing natural language data. It is a common operation to reduce the disturbance of irrelevant words from the whole document. In TauJud, we adopt *jieba* stop words corpus as a benchmark.
- **Back Translation**. Unlike the back translation method in QANet, we focus on translating sentences into English and back into Chinese, which means it is a one-to-one document augmentation task. We adopt the *Baidu translation API* for translation.
- **Clipping**. Given the complicated structure of judicial documents, we use documents clipping method to ensure the most important parts remaining. We provide an interface for users with custom regular expression to extract appropriate parts from a whole document. In some scenarios, the important paragraph of document can be easily extracted according to a prescribed format and a word library of action words by the local court. It is noteworthy that our dataset(CAIL2018) is composed of causes of action, so there's no need to use clipping method.

2.2 Judicial Augmentation

Besides, based on the characteristics of judicature, we develop three possible methods to augment judicial documents. We define a formative judicial document as a five-tuple $D_m(W_m, c, y, l, p)$ which denotes that judicial document D_m is mainly composed of the collections of words W_m, the cause of action c, the year y, the location l and people involved p. In order to ensure credibility within a community, augmentation must take all these complicated factors into consideration. We construct a judicial corpus with tokens with the following three methods. $\mathcal{A}$ is designed to modify the original test

Table 1: Notations

Symbol	Description
M	Number of judicial documents
m	Index of a judicial document
D_m	m-th judicial document
c	Cause of action of a judicial document
l	Location of a judicial document
y	Year of a judicial document
p	People involved of a judicial document
$\mathcal{A}$	Collection of groups of counterfactual tokens as followings
W_m	Collection of counterfactual words in judicial document D_m

documents, including three groups of counterfactual tokens(year, location, people). For convenience, we define the custom data formats and definitions used in Table 1.

- **Blindness**. Blindness substitutes all tokens with a special identity token, which shows identity term in test data. In TauJud, for document D_m, blindness can be defined as:

  $$\mathrm{D}_m(W_m, c, y, l, p) \longrightarrow D'_m(W_m, c, b_y, b_l, b_p)$$

  where b_y, b_l, b_p represent fixed symbols as 'YEAR', 'LOCATION' and 'PERSON'.
- **Counterfactual Augmentation**. Counterfactual augmentation does not make the model blind to the recognition items, it uses the generated counterfactual instances to increase the training set of the model. The additional example aims to guide the model to keep the perturbed identity term unchanged. This is a standard technology in computer vision, which is used to keep the model unchanged to the target position, image direction, etc. Counterfactual examples are labeled the same as the original. To guarantee counterfactual token fairness, TauJud can deal with complicated counterfactuals that involve more than just one token substitutions, e.g. "2011, Hua went to Beijing." and "2017 Guo went to Nanjing." The counterfactual augmentation is defined as follows:

  $$\mathrm{D}_m(W_m, c, y, l, p) \longrightarrow D'_m(W_m, c, y', l', p')$$

  where $y' \in \mathcal{A}_Y$, $l' \in \mathcal{A}_L$ and $p' \in \mathcal{A}_P$. For M test judicial documents, X denotes a group of counterfactual tokens. We can get the general distribution of counterfactual tokens:

  $$\forall x_m \in X, x'_m \sim \mathrm{Unif}[\mathcal{A}(x_m)]$$
- **Synonym Replacement**. Randomly choose words from the sentence except stop words. Replace each of these words with one of its synonyms chosen at random. In Judicial field, considering the high sensitivity of legal term, we deploy a large number of pairs of neutral synonyms *Syn* and store them as key-value pairs in the database. When W_m in keys, TauJud queries the database on a cloud server to get the synonym.

2.3 Report Generation

We have deployed TauJud as a command-line tool which can be used offline on personal computers as well. In order to help users have a deeper insight into the process of augmentation and data itself, TauJud implements visual figures based on the augmented documents.

3 EVALUATION

In order to verify the usability and applicability of our augmented documents in real-world scenarios, in this section, we design an experiment for evaluation. We perform experiments on the law case dataset CAIL2018, which contains 204, 231 test documents in total[9]. All of the test documents are the type of criminal cases, in JSON format. They contain the text of the judicial case and corresponding labels of the term of imprisonment and penalty.

According to the study of Lottier and Stuart[7], the time and space distribution of criminal cases is significant. Our judicial database involves all the cities and provinces of Chinese mainland. For document D_m, if W_m is a region in the database, the province of this region can be taken into account for analysis. Besides, if W_m represents one year, TauJud will randomly generate a new year between 1990 and 2018 as an alternative. We use counterfactual augmentation combined with stop words removal in universal augmentation.

After counterfactual augmentation and back-translation, we get the new province distribution of documents.The province distribution meets uniform distribution after augmentation. 7, 500 cases (the number of generated cases decided by the input dataset) are evenly distributed in every province of the mainland. The augmented documents provide a more reasonable regional distribution which can help model take the comprehensive evaluation.

Also, we conduct an analysis on year distribution of test documents, as shown in Fig.2. CAIL2018 test data contains most of the cases between 2010 and 2018. However, the number of cases from 1990 to 2004 is rather limited. After augmentation, data appears uniform distribution over the year from 1990 to 2018. About 40, 000 cases can be found in the dataset in each five-year. In this way, test documents can have more comprehensive coverage of China's judicial year.

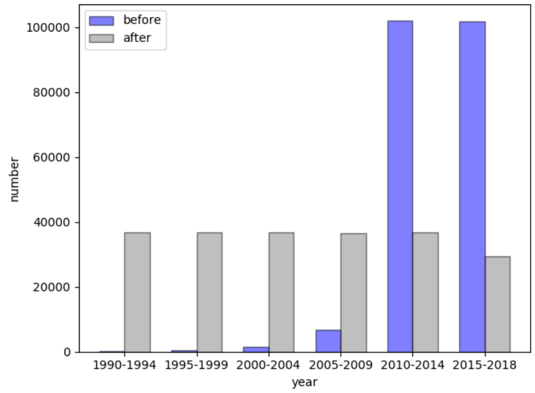

Figure 2: Year distribution before and after augmentation

Table 2 takes an example of the augmentation. If sentences are significantly changed, however, the original class labels may no longer be valid. For convenience, we show document D_{12428} in

Table 2: Examples of augmentation

D_{12428}	Before	After
c	In 2017, the defendants Zhao and Zhang quarreled at the west side of the gate of Nanjing Environmental Protection Bureau due to parking problems. Zhao beat Zhang and injured him.	In 2004, the defendants Hu and Guo had words with each other at the west side of the gate of Changsha Environmental Protection Bureau due to parking problems. Hu beat Guo and hurt him.
l	Nanjing	Changsha
y	2017	2004
p	$(Zhao, zhang)$	(Hu, Guo)

Table 3: The evaluation of ML models.

model	n=1	n=5	n=10	n=20
CNN	0.82	0.78	0.61	0.58
Text-CNN	0.52	0.71	0.67	0.65
ATT-CNN	0.83	0.73	0.62	0.59
ResNet	0.50	0.31	0.55	0.54
SVM	0.27	0.66	0.39	0.32
LinearSVM	0.44	0.29	0.29	0.25
RandomForest	0.15	0.57	0.34	0.31

English. After judicial augmentation, the cause of this judicial document is not changed, but it appears in different time, location and people without changing the raw labels, which corresponds to the real world very well. It should be noted that TauJud can also deal with counterfactual tokens such as colour, sex and orgnization.

To simulate the lack of test data situation, we initiate test data of only $10,000$ documents and augment test data for evaluation as a contrast. In order to achieve certain results, we take all of the following approaches: Stop Words Removal, Back Translation, Counterfactual Augmentation and Synonym Replacement. In cf, we simulate real crime distribution instead of random distribution of location of China. We further performed a comparative study on the widely used 7 widely used ML models for real-world penalty prediction analysis on more than two million judicial documents, including both classic machine learning models (SVM, LinearSVM, RandomForest) and deep learning models (CNN, text-CNN, attention-CNN, and ResNet). To evaluate the performance of our ML models, we use accuracy rate as the basic evaluation metric. We augment test documents with TauJud for n times, $n = 1, 5, 10, 20$.

As we can see in Table 3, when the amount of test documents is small ($n = 1, 5$), the result turns out to be very volatile, which means the test data can't show the precise performance of ML models. When n is large ($n = 10, 20$), it is found that the numerical solutions are convergent and are in good coincidence with experimental results, which proves that TauJud can augment test data for evaluation effectively.

4 USER MANUAL

To augment test judicial documents, run "python3 TauJud.py -i <arg> -o <arg> -jud <arg> -uni <arg> [-c <arg>]".

- "-h, –help": This argument is optional and is used if users want to get concrete usage of all the arguments.
- "-i, –input": This argument is required.The argument should be a file in JSON format as test data input.
- "-o, –output": This argument is required. Pass the name of the output file in JSON format as the argument.
- "-jud, –judicial": This argument is required. Only three parameters (blind, cf, sysnonyms) can be selected.
- "-uni, –universal": This argument is required. Only two parameters (backtrans, stopworddel) can be selected.
- "-c, –clipping": This argument is optional and is used when users want to select some paragraphs of the document. The scope should satisfy the regular expression.

For each run, for each iteration, the tool provides: (1) total running time; (2) total number of documents; (3) multiple combinations of argumentation; (4) province distribution chart; (5) year distribution.

5 CONCLUSION

In this demo paper, we present TauJud, a useful tool to realize the test augmentation of machine learning models in judicial documents. We have shown that judicial data augmentation operations can follow the uniform distribution over time and location, which simulates a large number of judicial cases in the real world. Continued work on TauJud could explore using ELMO embeddings for context preservation. Another thing to explore would be the performance of the newly generated test data to be trained on sequence models(LSTM, BiLSTM, etc.). We hope that TauJud's simplicity makes judicial test data augmentation easy and saves time and cost for labelling data.

ACKNOWLEDGEMENTS

This work was partially supported by the Fundamental Research Funds for the Central Universities (14380023).

REFERENCES

[1] Daniel Berrar and Werner Dubitzky. 2019. Should significance testing be abandoned in machine learning? *IJDSA* 7, 4 (2019), 247–257.

[2] Reuben Binns. [n.d.]. Fairness in Machine Learning: Lessons from Political Philosophy. ([n. d.]). arXiv:1712.03586 http://arxiv.org/abs/1712.03586

[3] Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H. Chi, and Alex Beutel. [n.d.]. Counterfactual Fairness in Text Classification through Robustness. ([n. d.]). arXiv:1809.10610 http://arxiv.org/abs/1809.10610

[4] Zichen Guo, Tieke He, Zemin Qin, Zicong Xie, and Jia Liu. 2019. A Content-Based Recommendation Framework for Judicial Cases. In *ICPCSEE*. Springer, 76–88.

[5] Tie-Ke He, Hao Lian, Ze-Min Qin, Zhen-Yu Chen, and Bin Luo. 2018. PTM: A Topic Model for the Inferring of the Penalty. *JCST* 33, 4 (2018), 756–767.

[6] Michael Kamp. 2019. *Black-Box Parallelization for Machine Learning*. Ph.D. Dissertation. Universitäts-und Landesbibliothek Bonn.

[7] Stuart Lottier. [n.d.]. Distribution of Criminal Offenses in Metropolitan Regions. 29, 1 ([n. d.]), 37. https://doi.org/10.2307/1137347

[8] Jason Wei and Kai Zou. [n.d.]. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. ([n. d.]). arXiv:1901.11196 http://arxiv.org/abs/1901.11196

[9] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478* (2018).

[10] Zihuan Xu, Tieke He, Hao Lian, Jiabing Wan, and Hui Wang. 2019. Case Facts Analysis Method Based on Deep Learning. In *WISA*. Springer, 92–97.

[11] Ge Yan, Yu Li, Shu Zhang, and Zhenyu Chen. 2019. Data Augmentation for Deep Learning of Judgment Documents. In *IScIDE*. Springer, 232–242.

[12] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. [n.d.]. QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. ([n. d.]). arXiv:1804.09541 http://arxiv.org/abs/1804.09541