Data Augmentation Technique for Process Extraction in Chemistry Publications

Yuni Susanti Hikaru Yokono Hiroaki Yoshida
Fujitsu Laboratories, Ltd.
{susanti.yuni, yokono.hikaru, yoshida.hiro-15}@fujitsu.com

1 Introduction

In the field of materials science and chemistry, deep understanding of the relation between structure and properties is required for creating new materials. Academists and scientists often seek for that information in scientific publications, since experimental observations have been the primary means to get the various chemical and physical properties of materials. Another important information from experimental observations includes the process of conducting the experiment or manufacturing procedure, because they can refer to that to conduct similar experiments, or to understand how some materials are made. Thus, access to a structured information of experiment process or manufacturing procedure is needed to improve the efficiency of their work.

The process extraction task aims to extract information such as materials, conditions, apparatus and other process-specific information from text describing some process. This task suffers from insufficient number of labeled resource, and data annotation is much harder for domain-specific tasks such as in chemistry or material science. One of the popular techniques for increasing the size of labeled training sets is data augmentation. In this paper, we explore simple data augmentation technique focused on process extraction task in chemistry domain.

Previous work has proposed some techniques for data augmentation in NLP. One popular study generated new data by backtranslation [7]. Other work has used data noising as smoothing [6] and predictive language models for synonym replacement [3]. Easy Data Augmentation, or EDA [5], comprehensively explores simple text editing-based data augmentation techniques for text classification task and showed a strong performance gain despite the simplicity of the technique. Inspired by their work, in this paper we explore simple data augmentation technique for process extraction in chemistry domain without using any dictionary. We also explore the possibility to create meaningful new data from present data by preserving the meaning of the original information of the present data.

2 Process Extraction

Process extraction task aims to extract entities from a process sentence and label them with process-specific entity-type labels such as operation/process, material, condition, apparatus, and so on. These labels define the roles of the entities in the process sentence. In this paper, we focus only on process sentence, which we define as a sentence containing *process predicates*. The following shows an example of a process sentence, consists of one process predicate **biotinylated**.

> Antibodies were **biotinylated** using a 5-fold molar excess of biotin-LC-NHS ester.

In this study, *process predicate* is defined as an expression representing the operation of a process, e.g. the word *biotinylated* in the above example. We assume $S = P_1, P_2, ...P_n$ where n is the number of process predicate P in sentence S.

Given a chunked text as the input, the process extraction task is to identify all the entities and their types/ roles in the process sentence. In this work, we represent entity as a word or sequence of words; the task is thus identifying the span of the entities. We cast this problem as a sequence labeling task.

3 Data Augmentation

EDA [5] explores four simple text editing-based data augmentation techniques for text classification task: synonym replacement, random swap, random insertion, and random deletion. Those techniques can be easily applied to process extraction task, but in our experiment, it led to performance decrease. This result is expected, considering the different nature of the task. For instance, it is natural to think that the order of the words in text classification task might not be as important as in a sequence labeling task such as process extraction. Thus, swapping the words or randomly insert and delete the entities could be the main reason of the decrease in the performance.

In this work, we introduce simple text editing-based data augmentation method for process extraction task. For this task, the general idea is replacing an entity with entity having the same entity type

to create a new augmented sentence. Therefore, we do not have to worry about the entity label in the new sentence since the entities are replaced with the entity having the same label (the original labels are maintained).

Similar to EDA [5], our proposed method also does not require a domain-specific dictionary or any dictionary at all[1]. Suppose a sentence has a set of entities and a pattern (Sentence = Sentence_{NE}, $\text{Sentence}_{pattern}$). The *pattern* here refers to the grammatical structure of the sentence including the order of words, the stop word choices, and the like. The main idea of our proposed method is for a set of entities in an **input sentence** (Input_{NE}), a new sentence using the entity set is created using a sentence pattern from a **source sentence** ($\text{Source}_{pattern}$). This new sentence, or the **augmented sentence**, would have the entities from the input sentence and the pattern from the source sentence (Augmented = Input_{NE}, $\text{Source}_{pattern}$).

In the following, we explain in detail of our proposed data augmentation method. Given an input sentence, the data augmentation method is twofold:

1. **Source sentence selection**: search for sentence in the training data to be the pattern for the new sentence. The source sentence candidates are all sentences in the training data except the input sentence. To make k-new sentences, k-number of source sentences are selected.

2. **Entity replacement**: create a new sentence, i.e. the augmented sentence, by substituting the entities in the source sentence having the same entity types with the corresponding entities in the input sentence.

In this paper, we introduce several methods on source sentence selection.

Label-similarity method (LSIM). In the first method, we simply choose the sentence having the biggest number of label overlap with the input sentence as the source sentence. The idea is sentences having high label overlap should be similar in structure. For instance, the following input and source sentences have 100% label overlaps: MATERIAL MATERIAL O PP O MAT-DESC MATERIAL.

> Input sentence: Oxalic acid were dissolved in deionized water.
> Source sentence: Borac acid was added to boiling alcohol.

As we can see, both sentence indeed has the same structure, and it would be safe to use the structure of the source sentence with the set of entities from input sentence, and *vice versa*. In our implementation, we rank the source sentence candidates according to their label overlap, and choose the k-highest scoring sentences as the source sentences to make k-new augmented sentences. For above input-source sentences pair, the augmented sentence would be

> Augmented sentence: Oxalic acid was added to deonized water.

The augmented sentence uses the pattern from source sentence and entities from input sentence, e.g. *oxalic acid* substitutes *borac acid* since they have the same entity type, i.e. *material*. As we can see, the newly-created augmented sentence still somehow makes sense grammatically even though the meaning slightly differs.

Process-similarity method (PSIM). In the label-similarity method, the generated sentence often loses the meaning of the original sentence, or sometimes they form a non-sense new sentence. This could be problematic since instead of giving a useful information to the model, they can be a noise to the training data. To be able to preserve the meaning of the original sentence, we choose the sentence describing similar process as the source sentence in this second method of source sentence selection.

In a process sentence, to preserve the meaning of the original sentence, the most important part is the process predicate. Therefore, we focused on the process predicate similarity to find similar process sentence as the source sentence. We used Word2vec pretrained model[2] to calculate the vector similarity of the process predicate of the input sentence and the source sentence candidates. The sentence having the highest process predicate similarity with those of the input sentence is chosen as the source sentence.

Since one sentence could have more than one process predicates, we make pair combinations of process predicates from the sentence pair and calculate the similarity between them. For example,

> Input sentence: The pH of the mixed aqueous solution was adjusted to 6 with aqueous ammonia (28%).
> Source sentence candidate: The pH value of this K2HPO4 was adjusted to 0.1 using 0.1 M phospate solutions.

For above example, all pairs of process predicates are (mixed, adjusted), (mixed, using), (adjusted, adjusted), (adjusted, using). In our implementation, the final similarity score for a candidate would be 1) all pairs average (**PSIM**), which is the average

[1] Nonetheless, EDA [5] uses WordNet lexical dictionary to find synonyms

[2] Available on https://github.com/olivettigroup/materials-word-embeddings

Table 1. Performance (F1 scores) of data augmentation methods on the process extraction model by varying the training set fractions and the number of augmented sentences per original sentence (k). **Bold** indicates the best score for each training set fraction; *italic* indicates decrease in performance over the original model.

fractions	training set size original (org)	training set size org+augmented	k	original model	LSIM	PSIM	PSIM-A
10%	189	1004	5	46.3	52.2	**56.9**	52.3
20%	379	1989	5	53.2	53.7	57.6	**58.4**
30%	568	2963	5	57.8	59.3	60.2	**62.1**
40%	758	4063	5	62.2	62.4	66.0	**66.2**
50%	947	5027	5	63.2	*61.8*	**66.2**	64.0
60%	1136	6021	5	64.5	**66.1**	**66.1**	65.8
70%	1326	7056	5	66.2	68.0	**69.7**	68.0
80%	1517	8077	5	66.7	67.4	**69.4**	67.4
90%	1706	9121	5	66.0	67.4	69.6	**70.0**
100%	1896	10106	5	68.2	68.9	71.7	**72.1**
10%	189	1493	8	46.3	55.9	57.2	**57.9**
40%	758	6046	8	62.2	62.6	**67.5**	67.3
90%	1706	13570	8	66.0	67.9	69.4	**70.1**
10%	189	2797	16	46.3	54.0	**55.6**	55.0
40%	758	11334	16	62.2	63.6	67.1	**67.6**
90%	1706	25434	16	66.0	68.3	**71.2**	70.1

similarity scores of all pairs combined, or 2) aligned pair average (**PSIM-A**), which is the average of the highest scores for each of the input sentence's process predicate. For example, the highest score for *mixed* would be either the similarity score of (mixed, adjusted) or (mixed, using), while for *adjusted* would be either (adjusted, adjusted) or (adjusted, using).

In the entity replacement step, we keep the process predicates of the source sentence in the augmented sentence. The augmented sentence as the result of the above input-source sentence pair would be

> Augmented sentence: The pH value of this solution was adjusted to 6 using 28% ammonia aqueous.

As we can see, the original meaning of the input sentence is kept in the augmented sentence to some extent. In addition, by keeping the original process predicates (the process predicates of the source sentence), the augmented sentence could maintain the sentence structure of the source sentence.

4 Experimental Setup

To test the effectiveness of the data augmentation, we conducted experiments on process extraction task model trained with and without data augmentation method. The following describes the dataset and the model used in the experiments.

4.1 Dataset

In the experiment, we used a dataset of 235 synthesis procedures (1896 sentences in training set and 210 sentences test set) annotated by domain experts for named entity recognition task such as identifying reaction conditions and materials [2, 4]. The dataset includes the labeled entity mentions associated with entity types which specify a category/kind for the entity mention. The dataset is collected from paragraph describing inorganic material synthesis procedure selected from 2.5 million publications. There are in total 21 entity types in the dataset, including: material, operation, amount-unit, condition-unit, material-descriptor, apparatus etc.

4.2 Process Extraction Model

We conducted the experiments for one of the state-of-the-art models in sequential labeling task: Bidirectional LSTM network with CRF model (BI-LSTM-CRF), which shows good performance for sequence tagging [1]. We run the model without data augmentation as the original model for experiment in current study.

5 Result and Discussion

We run both the training using only original data and training with data augmentation for the following training set fractions (%): 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. For all experiments, we used the same hyperparameters settings. We generate $k : 5, 8, 16$-augmented sentences per each original sentence. Table 1 shows the results (F1 scores, in %) of the process extraction model described in section 4.2 equipped with or without the data augmentation methods described in section 3.

To sum up the results, almost all of the models trained using data augmentation outperformed the original model trained without data augmentation. The improvement varies depending on the size of the data used in the training, ranging from as small as 0.2 to 10.6 points improvements. The best F1 scores without data augmentation, 68.2, was achieved using 100% of the training data. The model PSIM, which was trained using data augmentation, surpassed this number by achieving F1 score of 69.7 while only using 70% fraction of the training data.

Table 1 shows that the data augmentation has the most significant improvements (up to 10.6 points) for training on the smaller dataset fractions (10%, 20%). Nevertheless, the improvements were quite reasonable, up to 4 points, using the higher training set fractions (90%, 100%). Since overfitting tends to be more severe when training on small data, this result is encouraging since it shows that the data augmentation could reduce the risk of overfitting, especially on smaller size of training set.

While the improvements across all data augmentation methods are more or less the same, PSIM and PSIM-A data augmentation methods constantly contributed to the performance gain on the process extraction model on all training set fractions. LSIM model gave performance improvement on all experiments except on experiment using 50% fraction of the training data. It showed lower F1 scores than the original model on that experiment. Since in all other experiments LSIM model showed improvements over the original model, we considered this as a peculiar case and more thorough investigation is needed to understand the cause of this case.

Does k affect performance? We also conducted experiments where we varied the number of augmented sentences, k, per each original input sentence (k: 5, 8, 16). The result is shown in the lower part of Table 1. To summarize, the performance gains are not significantly different across different numbers of k in our experiments.

Adding pretrained model. EDA [5] stated that it might not yield substantial improvements when using pretrained models, so we conducted experiments where we added a pretrained Word2vec model. The result is summarized in Table 2. It showed some small improvements (up to 4.2 points) on the PSIM and PSIM-A, while we can notice some decrease in performance on the LSIM model. We may consider this decrease negligible since they are very small.

Table 2. Performance (F1 scores) of the models after adding a pretrained model.

% train	original	LSIM	PSIM	PSIM-A
10%	57.7	60.7	61.6	**61.9**
20%	65.7	*64.7*	65.8	**65.9**
30%	67.9	*66.4*	**68.7**	68.4
90%	73.0	*72.6*	**73.5**	73.0
100%	73.4	73.6	**74.1**	73.6

6 Conclusion

We have introduced simple data augmentation methods to be used on process extraction task. In the chemistry domain, process extraction task aims to extract entities from a process sentence and label them with process-specific entity-type labels such as operation/process predicate, material, condition, apparatus, and so on. This task suffers from insufficient number of labeled resource, and data annotation is much harder for domain-specific tasks such as in chemistry or material science.

In this paper, we have shown that simple data augmentation method can boost performance on the process extraction task, which is a sequence tagging task. Our proposed method attempts to create meaningful augmented sentences by utilizing process information (the *process predicate*). We showed that the proposed data augmentation substantially improves performance, up to 10.6 points, and could potentially reduce overfitting especially when training on small dataset. Future work includes exploring more sophisticated data augmentation methods and evaluating on more general sequence labeling tasks.

References

[1] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015.

[2] Edward Kim, Zach Jensen, Alexander van Grootel, Kevin Huang, Matthew Staib, Sheshera Mysore, Haw-Shiuan Chang, Emma Strubell, Andrew McCallum, Stefanie Jegelka, and Elsa Olivetti. Inorganic materials synthesis planning with literature-trained neural networks. *arXiv preprint arXiv:1901.00032*, 2018.

[3] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[4] Sheshera Mysore, Zachary Jensen, Edward Kim, Kevin Huang, Haw-Shiuan Chang, Emma Strubell, Jeffrey Flanigan, Andrew McCallum, and Elsa Olivetti. The materials science procedural text corpus: Annotating materials synthesis procedures with shallow semantic structures. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 56–64, Florence, Italy, August 2019. Association for Computational Linguistics.

[5] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6381–6387, Hong Kong, China, November 2019. Association for Computational Linguistics.

[6] Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. Data noising as smoothing in neural network language models. *CoRR*, abs/1703.02573, 2017.

[7] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR*, abs/1804.09541, 2018.