A Case study on Start-up of Dataset Construction: In Case of Recipe Named Entity Corpus

Yoko Yamakata
Graduate School of Informatics
Kyoto University
yamakata@i.kyoto-u.ac.jp

Keishi Tajima
Graduate School of Informatics
Kyoto University
tajima@i.kyoto-u.ac.jp

Shinsuke Mori
Academic Center for Computing and Media Studies
Kyoto University
forest@i.kyoto-u.ac.jp

***Abstract*—In this paper, we report our experience in constructing a corpus of annotated recipe documents. We describe problems we found and explain how we managed them. One of the problems we faced in the construction of our recipe corpus is the difficulty of establishing a clear, stable, and complete guideline instructing annotators how to annotate. During the annotation, we found many unexpected cases for which the pre-defined guideline is not clear enough, and even cases for which the pre-defined guideline provides no guidance at all. As a result, we need to update the guideline multiple times during the annotation, and also need to revise annotations we have done before the updates. During that process, we have several trade-offs, and it is not easy to decide when and how often we should revise the annotations. It is even unclear whether we should revise them or should use the human resource for that for annotating more data. We show an experiment, whose result suggests that we should revise the old annotations. Another problem we had is the management of versions of guidelines, sets of annotations corresponding to them, and communication between participants. In this paper, we explain how we managed these problems.**

I. Introduction

Supervised machine learning technology is becoming more and more important because of its recent improvements and success, especially those by deep learning techniques. As supervised machine learning requires a corpus for training, methodologies for efficiently producing high quality corpora have also become an important research issue. One approach to the issue is the adoption of crowdsourcing. Crowdsourcing is, however, useful only when the following conditions hold:

- the task is a simple task, such as binary labeling, that does not require a complex task description and only requires basic knowledge that anyone has, and
- we have enough amount of data so that some errors in them are tolerable.

There have been much improvement of methods for handling noises in the training data in these years, but we still need the latter condition.

When these conditions are not satisfied, e.g., when the task includes complex annotation, crowdsourcing cannot be applied at least in a straight-forward manner, and a methodology for producing corpora for such tasks is yet to be established.

In addition, best practices in corpus construction known so far quite depends on the type of data and the problem to solve. As a result, there have been many research papers reporting corpora they created for specific domains, such as general object recognition [1]–[3], handwritten character recognition [4], face recognition [5], food recognition [6], and visual font evaluation [7]. These corpora and the lessons reported in these papers are very useful for and shared by other researchers.

We have also constructed a corpus of annotated text describing recipes, which is publicly available at http://www.ar.media.kyoto-u.ac.jp/how-to/recipe-NLP/. One of the serious problems we experienced in its construction is difficulty in establishing a clear, stable, and complete guideline instructing annotators how to annotate. A corpus constructed without it would be inconsistent, and learning from such a corpus would result in a model with low precision. Providing annotators with such a guideline in advance is, however, a very difficult task when we need complex annotation. Even if we carefully write down the guideline covering all expected cases, we will find many unexpected cases for which the pre-defined guideline is not clear enough or does not provide a guidance at all.

When we receive a report from an annotator on such an unexpected case, we need to update the guideline, and notify it to the annotators so that they can handle such cases from now on. An update of the guideline, however, often requires retrospectively revising the annotation we have done before the update. Revising them is important to have a consistent corpora on one hand, but on the other hand, we could annotate more data by using the human resources used for that. That is, we have a trade-off between revising the annotations that were done under old guidelines (the consistency of the corpus) and annotating more new data (the size of the corpus).

We also need to decide how often we revise the old annotations. If we do it every time we update the guideline, one annotation can be repeatedly revised, which would be inefficient compared with revising it only once after we have finished annotation of all data and have fixed the guideline. On the other hand, if we revise an annotation long after we annotated it and after many guideline updates, it is sometimes difficult to revise it by re-understanding the old annotation and by tracing back the many updates of the guidelines.

In addition, during these processes of updating guidelines, notifying them, and revising annotations, we need to manage several data and their workflows, and also need much communication between project members and annotators. There have been several proposals of systems for supporting corpus

This is the submitted version of the paper. The final revised version is available at: https://doi.org/10.1109/BigData.2018.8622382

creation phase for machine learning tasks [8], [9] and also proposals of systems for supporting general crowdsourcing tasks [10], [11], but none of them has discussed these problems and data management in detail. We need a system supporting the management of them.

In this paper, we describe these problems in more detail, and report how we managed them. We also show the result of an experiment comparing the benefit of revising old annotations and the benefit of annotating more new data. The result shows that the annotations under the initial guidelines in our corpus creation was quite unreliable, and the benefit of revising old annotations under new guidelines can be bigger than the benefit of annotating more data even when we have not annotated enough data and the size of the data set is very small.

II. Recipe Named Entity Corpus in English

In this section, we introduce the English cooking recipe corpus that we had constructed as a concrete example for analyzing annotation process on corpus construction. Cooking recipe corpus and its annotation guideline had been originally defined for Japanese cooking recipes [12]. We adjusted the guideline to English one while leaving the meanings of tags as it is in Japanese, as much as possible. The original recipe data was sampled from each category of "dish type" in the Allrecipes UK/Ireland web site (http://allrecipes.co.uk/) as of December 2016. The sample selection criteria are based on the proportions and rank orders in which recipes are listed within each dish type. In total, we annotated 100 recipe documents from 15 dish types. The number of the recipes of each dish type in the web site and the corpus are described in our previous paper [13].

We annotated named entities used in the recipe text, which we call recipe named entities (r-NEs). Though the original r-NE classification defines eight r-NE tags for Japanese recipes, we added two more tags, "**Ac2**" and "**At**", in order to account for additional phenomena that occur only in English recipes. Table I shows the resulting ten r-NE tags. An English native speaker annotated the 100 recipes according to translated guidelines of Japanese one using the IOB2 chunking format [14]. Fig. 1 shows the correct annotation results for the sentence "Preheat oven to 200 C/Gas mark 6.". Each tag is given to a word or a set of words which designates a single and inseparable object/action/phenomenon. For example, "*Gas mark 6*" in the sentence above designates the dial of the oven at 6 so it designates a single meaning and the first word "*Gas*" is annotated as "**St-B**" and continuous words "*mark*" and "*6*" are annotated as "**St-I**", in which "**-B**" and "**-I**" are abbreviation of "Begin" and "Inside". "*to*" are annotated as "**O**" because this word is **Outside** of named entities. Average annotation time was 24 minutes per recipe, including initial training.

TABLE I
Recipe Named Entity (r-NE) Tags

Tag	Meaning	Remarks
F	Food	Eatable, also intermediate products
T	Tool	Knife, container, etc.
D	Duration	Duration of cooking
Q	Quantity	Quantity of food
Ac	Action by chef	Verb representing a chef's action
Ac2	Discontinuous Ac **(Eng. only)**	Second, non-contiguous part of a single action by chef
Af	Action by food	Verb representing action of a food
At	Action by tool **(Eng. only)**	Verb representing a tool's action
Sf	Food state	Food's initial or intermediate state
St	Tool state	Tool's initial or intermediate state

Preheat	oven	to	200	C	/	Gas	mark	6	.
Ac-B	*T-B*	*O*	*St-B*	*St-I*	*O*	*St-B*	*St-I*	*St-I*	*O*

Fig. 1. Example of annotation

A. Communications between Requesters, Annotators and Supervisors

Next, we explain the annotation process used in our corpus construction. Our annotation process includes three types of participants: *requesters*, *annotators*, and *supervisors*. The communication between them are illustrated in Fig. 2.

1) The requester describes the annotation guideline and sends it to annotators.
2) The annotators annotate the data according to the guideline and
3) return questions and exceptional cases that are not clearly specified in the guideline.
4) The requester discusses with supervisors if required and
5) revises the guideline from ver. Nth to $(N+1)$th.
6) The requester sends the revised version of the guideline to the annotators and
7) the annotators update the previous annotation results to fit the current guideline.

In the case of our English recipe corpus construction, there were one requester (the first author of this paper), two annotators (one was a beginner and the other was the fist author) and two supervisors (both were specialists of natural language processing and one of them also had knowledge and experiences on machine learning on NLP). First, all the recipes were annotated by the beginner and then the first author verified the annotation to ensure it adhered to the final guidelines. The questions and exceptional cases were reported by the annotator by describing them in square brackets in the same file with the annotated data and the requester replied them frequently at the beginning of the work. The guideline were revised and announced to the beginner in the middle of first annotation work and then revised to the final version after completion of the first annotation.

III. Problems and Managements

In this section, we report what problems had happened in our annotation process, and how we managed them.

A. Guideline Revision Problem

The requester responds to the question sent by an annotator soon if she can answer it according to the current guideline.

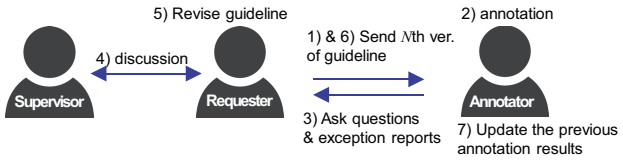

Fig. 2. Communication between requesters, annotators and supervisors

However, if it contains a problem out of the current guideline, the requester had a discussion with the supervisors about whether we should revise the current guideline, and also discuss how to do it when we need it. Because simplistic revise of the annotation guideline sometimes causes serious contradiction on guideline and results in noisy annotation results with low recognition accuracy, the guideline should be revised carefully under deep discussion from various view points.

The NLP professionals suggested reasonable annotation rules to keep linguistic consistency. For example, the requester decided to add a new tag Ac2 that was not used for Japanese recipe corpus guideline according to the NLP professional's suggestion. The reason was as follows; The tagging guidelines specify that each chef action in the cooking process should be tagged as a single r-NE Ac. In Japanese, words corresponding to a single action are always contiguous. However, in English, a single chef action can be expressed as a discontinuous phrase in such situations as phrasal verbs (e.g., "*throw (something) away*"), verb/purpose combination (e.g., "*toss (something) to coat*") and collocation (e.g., "*bring (something) to the boil*"). The new tag Ac2 is used to annotate such discontinuous second phrase in our English recipe corpus guideline.

The ML professional advice were also useful to narrow down the variety of tags as he had such knowledge as what are easy and what are difficult for the learning method we are using. For example, we decided that we do not annotate adverbs, such as "well" in "mix well" because ML professional suggested that adverbs have large vocabulary and are difficult to be classified while these words are not so important to explain cooking procedure.

To decide when the new version of the guideline would be distributed is very difficult problem. If it is too early, another controversial exceptions would come very soon and the requester is required to revise the guideline again. However, if it is later, more data is annotated under old version of guideline, which means more data has to be updated to the current version. There is also data status management problem which is introduced in the next section.

B. Data Status Management Problem

The annotator always annotated the data according to the current version of the guideline at that moment. Repeatedly revising the guideline resulted in a situation that some data was annotated under one version of the guideline while the other was annotated under the other versions. In our annotation task, the annotated data under a different version of the guideline was stored in a different folder whose name was associated with its guideline version. Fig. 3 illustrates annotation data status under multiple versions of the guideline.

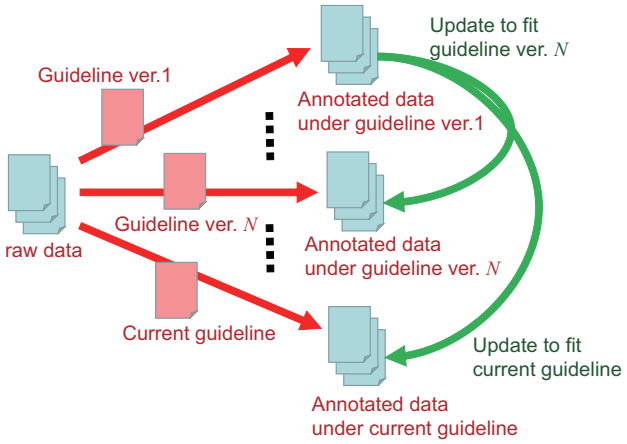

Fig. 3. Multiple annotation results can exist under different versions of guideline.

Here, the annotators have two types of work; i) annotation to raw data and ii) update annotated data under the old guidelines to fit the current guideline. To decide which work should be prioritized, we needed to consider the following parameters:

- How easy to update the annotation results to fit the current version of the guideline comparing to newly giving annotation.
- Which is more effective to improve classification accuracy; the bigger size of annotation data even though it is annotated under multiple versions of the guideline, or the smaller size of annotation data under the final version of the guideline.

We discuss this questions based on an experimental result in Sec. IV.

IV. Experiments and Discussions

A. Data Size v.s. Update

To discuss the questions given at the end of the previous section, we conducted the following experiment.

As mentioned in Sec. II, 100 recipes were annotated by the beginner first and then the annotation results were updated to fit the final guideline by the requester by herself. Hereafter we call the first annotation results as "1st Rslt." and updated results as "Final Rslt." We adopted named entity recognizer PWNER [15] that is based on pointwise prediction of whether each word either Begins or is Inside or Outside an NE (i.e. to have one of the tags BIO) through a search for the best sequence of tags under the tag sequence constraints.

We had prepared another 120 recipes that were annotated according to the final guideline, in which 20 recipes were used for adjusting hyper-parameters and the remaining 100 recipes were used for testing. In the Fig. 4, the blue line shows the F-measures of PWNER trained with 25, 50, 75 or 100 of "Final Rslt." while the orange line shows it trained with 25, 50, 75 or 100 of "1st Rslt." According to this figure, 25 of "Final Rslt."

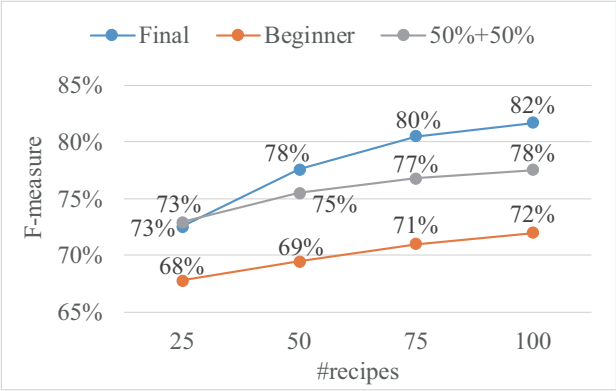

Fig. 4. Classification accuracies with different size of first and final annotation results.

brought almost the same accuracy with 100 of "1st Rslt." This result suggests that final annotation results are significantly effective to obtain higher classification accuracy.

We also conducted an experiment that PWNER trained with a half-and-half mixture of "Final Rslt." and "1st Rslt.". The gray line of Fig. 4 shows the F-measures of PWNER trained with 25, 50, 75 or 100 of the mixture. According to the comparison of 100 half-and-half mixture (which means 50 "Final Rslt.' and 50 "1st Rslt.") with 50 of "Final Rslt.", these two are almost the same as 78%, which means 50 of "1st Rslt." did not contribute to improve the classification accuracy. These results suggests that the requester should choose the strategy that she asks the annotator to give annotation according to the final version of the guideline as soon as she fix it. However, she could not forecast when the guideline would be fixed until the end of this annotation for all data at least one time. Consequently, it was the best strategy in this case that the annotator gave annotations under current version of the guideline and after the end of giving the first annotation to all the data, the requester fixed the guideline and updated all to fit the final version of guideline.

V. Conclusion

In this paper, for discussing what kinds of support are required for start-up of dataset construction, we had picked up our work about constructing cooking recipe named entity corpus as an concrete example and reported what problems happened and how did we dealt with them. Communications between requester, annotators and supervisors (expert in related fields) were important especially when the annotation guideline was revised because the requester has to collect exceptions arose during annotation from the annotators and discuss with the supervisor to keep consistency of the guideline in various viewpoints. Data status management is also very important because repeatedly revising the guideline results in multiple versions of the guideline and data are also annotated under different versions of the guideline in different places. We focused on a question which is better, big size of annotation data under old versions or small size of annotation data under final and consistent guideline, and conducted an experiment. The results suggest that the annotation data under old version of the guideline did not contribute to improve classification accuracy, which means the requester should take the strategy to obtain as much data annotated under the final guideline as possible.

Of course the best strategy must differ for each case. In this case, the noise of the 1st annotation results were caused not only by immaturity of annotator but also ambiguous definition of the guideline. If the guideline is fixed beforehand, the best strategy might change as giving annotation to all the data anyway and then verifying the results to ensure it adhered to the last guidelines. The future work is to find a way of choosing the best annotation strategy according to each situation. Also construction of an annotation support tool for helping communications and data status management is another future work as well.

Acknowledgment

This work was supported by JST CREST Grant Number JPMJCR16E3, Japan.

References

[1] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE CVPR*, June 2009, pp. 248–255.

[2] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

[3] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," online: http://www.cs.toronto.edu/kriz/cifar.html, 2014.

[4] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, Nov 2012.

[5] D. Miller, I. Kemelmacher-Shlizerman, and S. M. Seitz, "Megaface: A million faces for recognition at scale," *CoRR*, vol. abs/1505.02108, 2015.

[6] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *Computer Vision – ECCV 2014*, 2014, pp. 446–461.

[7] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. S. Huang, "DeepFont: Identify your font from an image," in *Proc. of the 23rd ACM International Conference on Multimedia*, 2015, pp. 451–459.

[8] L. Reeve and H. Han, "Survey of semantic annotation platforms," in *Proc. of the 2005 ACM Symposium on Applied Computing*, ser. SAC '05, 2005, pp. 1634–1638.

[9] F. Ciravegna, A. Dingli, D. Petrelli, and Y. Wilks, "User-system cooperation in document annotation based on information extraction," in *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, 2002, pp. 122–137.

[10] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Commun. ACM*, vol. 54, no. 4, pp. 86–96, Apr. 2011.

[11] M. Yuen, I. King, and K. Leung, "A survey of crowdsourcing systems," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, Oct 2011, pp. 766–773.

[12] T. Sasada, S. Mori, Y. Yamakata, H. Maeta, and T. Kawahara, "Definition of recipe terms and corpus annotation for their automatic recognition (in Japanese)," *Journal of Natural Language Processing*, vol. 22, no. 2, pp. 107–131, 2015.

[13] Y. Yamakata, J. Carroll, and S. Mori, "A comparison of cooking recipe named entities between Japanese and English," in *CEA 2017*, 2017, pp. 7–12.

[14] E. F. T. K. Sang and J. Veenstra, "Representing text chunks," in *EACL '99*, 1999, pp. 173–179.

[15] T. Sasada, S. Mori, T. Kawahara, and Y. Yamakata, "Named entity recognizer trainable from partially annotated data," in *PACLING 2015*, 2015, pp. 148–160.