GuideLink: A Corpus Annotation System that Integrates the Management of Annotation Guidelines

Kenta Oouchida[a], Jin-Dong Kim[b], Toshihisa Takagi[a,c], and Jun'ichi Tsujii[b,d]

[a] Database Center for Life Science Research Organization of Information and Systems
2-11-16 Yayoi, Bunkyo-ku, Tokyo, 113-0032, Japan
{oouchida, takagi}@dbclis.rois.ac.jp
[b] Department of Computer Science, Faculty of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
{jdkim, tsujii}@is.s.u-tokyo.ac.jp
[c] Department of Computational Biology, University of Tokyo
Kashiwa-no-ha 5-1-5, Kashiwa-shi, 277-8568 Japan
[d] School of Computer Science, University of Manchester
National Centre for Text Mining, University of Manchester

Abstract. This paper presents an annotation framework wherein the management process of the annotation guidelines is integrated into the annotation process. Such an integration allows systematic management and reference of guidelines during annotation. For the evaluation of the proposed annotation system, we compare the conventional and proposed annotation frameworks, experiments using automatic guideline suggestion, and describe a unique feature of the integrated framework.

Keywords: Annotation Guideline, Annotation Tool, Corpus Annotation

1 Introduction

It is generally recognized that maintaining consistency is a key problem in manual corpus annotation. To maintain consistency in the annotation, the annotators have to share the common annotation policy throughout the annotation project. Some parts of the annotation policy are documented in the early stages of an annotation project, while other parts are documented during the annotation process. We will refer to the former parts as the *annotation scheme* and the latter parts as the *annotation guidelines*.

The annotation scheme typically documents the core of the annotation policy, including the goal of the annotation work, the vocabulary of terms related to the annotations including labels, on the syntax of the annotation. For example, the MUC-7 named entity annotation scheme (Chinchor and Robinson, 1997) defines three labels for tagging text span: *ENAMEX* for named entities, *TIMEX* for temporal expressions, and *NUMEX* for number expressions. The goal of the annotation work is to find all mentions of the named entities and to tag them with their proper labels.

The annotation guideline details how to treat some borderline cases that the annotator cannot decide how to treat with the annotation scheme. Although sometimes guidelines are prepared together with annotation scheme, it is often impossible to provide a complete set of guidelines beforehand. During the annotation process, it is typical for annotators to communicate in developing guidelines when difficult cases arise. Guidelines are consequently important not only for annotators to keep the consistency of annotation process, but also for the users to understand the annotation later (see Section 2.2). Nevertheless, there has only been a few studies done on the guideline production (Lu *et al.*, 2006).

In this paper, we propose a framework in which the association between the guidelines and the corpus is important, and support the accessibility between the guideline and the corpus. In addition, we can systemically integrate the management of the annotation guideline into the annotation process in the proposed framework. We present **GuideLink**, an implementation of the annotation framework, which is integrated with the existing annotation tool, XConc Suite.[1]

2 Related works

2.1 Tools for corpus annotation

Many software tools have been developed for supporting corpus annotation. Well-known ones include WordFreak (Morton and LaCivita, 2003), MMAX (Mueller and Strube, 2001), Knowtator (Ogren, 2006), GATE (Cunningham *et al.*, 2002) and XConc Suite. While they are all widely used, each has its own strength. For example, MMAX is designed for multi-level annotation. Knowtator puts its focus on ontology-based annotation. GATE is a language engineering infrastructure. Both WordFreak and XConc Suite focus on flexibility of the format of corpus and annotation. As far as the authors know, however, there is no tool supporting guideline production in an integrated way.

2.2 Annotation guidelines and their production

Although only few studies on guideline production exist, researchers have long recognized the importance of documenting the annotation policy. One of the most popular annotated corpora, Penn Treebank (Marcus *et al.*, 1993), is also well known for the comprehensive documentation of its annotation policy. Its well documented annotation guidelines are indispensable tools for the proper use of Penn Treebank. The annotaiton policy of the SUSANNE corpus (Sampson, 2002) is published as a part of a book.

Despite the importance of annotation guidelines, the process of guideline production has not been studied extensively. Even some of the latest annotation projects relied on traditional ways of communication and documentation for guideline production. For example the Caderige project (Alphonse *et al.*, 2004) used e-mails for communication between annotators, and the archive became database of guidelines. PennBioIE (Kulick *et al.*, 2004) repeatdly updated a web page dedicated to documentation of guidelines. GENIA made use of a Wiki system.[2]

Although adopting web-based documentation enhanced the guideline production and utilization in sharing and searching, it is difficult to conclude that the guideline production process is well integrated with the annotation process. On the other hand, we produce the guideline using the examples from annotated corpus, and we annotate the corpus using the annotation guideline. The annotator must very often switch between the annotation system and guideline management system during annotation process.

3 Modeling the workflow of corpus annotation

Figure 1 shows a common workflow of corpus annotation that the authors modeled through discussion with annotation practitioners. It concerns the common practice of annotators to make decisions required for annotation to a given text span. If the decision is trivial, the annotators can perform annotation without help of guidelines, jumping from (1) to (4). If the decision is difficult, it is common for annotators to consult the guidelines for solutions (2). If the guidelines applicable to the case are found, annotation can be performed following the guidelines (4). If not, annotators have to find a solution themselves through, e.g. discussion. The new solution has to be articulated and stored as a guideline for future reference (3). The annotators can then proceed to the annotation (4). The result of the annotation is an attachment of information to the text. Sometimes the

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi?page=XConc+Suite+User+Manual
[2] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi?page=GENIA+corpus

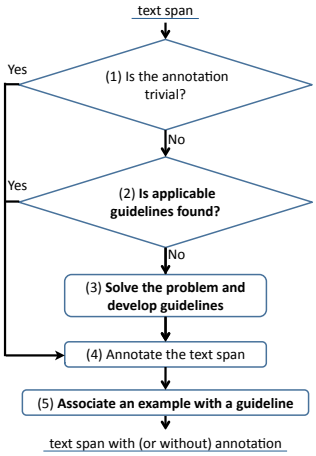

Figure 1: The annotation flow model considering guidelines

text span remains without any attached information depending on the annotator's decision. We call this negative annotation instance, which will be discussed in detail in Section 4.4. The newly obtained annotation is often associated with relevant guidelines for future reference (5).

Conventional annotation tools (see Section 2.1) do not consider the role of guidelines, and support only the step (4) of the workflow. The typical functionality of such tools includes convenient UI for text selection, label browsing and selection, among others. For this reason, the development process of the annotation guidelines has always been separated from the annotation process with the conventional annotation tools. In this paper, we propose an annotation framework that integrates the development of the guideline based on the workflow presented in this section.

4 Three-layer model for annotation and guideline management

For the integration of the guideline management into the annotation framework, we propose a three-layer model for the data management. Most of the corpus annotation tools (Section 2.1), which do not consider the guideline management, can be described as the two-layer model which consists of the text layer and the annotation layer. For the guideline management, the annotation guideline layer is added, making it a three-layer model. In this section we describe data structures for these three layers and their connectivity.

4.1 Text Layer

The text layer maintains the text documents to be annotated (Text Layer in Figure 2). A text document is usually treated as a sequence of characters or words, and a specific span of a text document is expressed by the offset of the beginning and ending characters with the document ID.

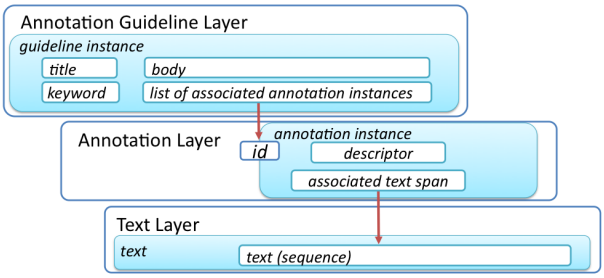

Figure 2: The data structure for the three-layer model

4.2 Annotation Layer

The annotation layer facilitates the location where the annotations to the text documents are maintained (Annotation Layer in Figure 2). Corpus annotation can then be defined as a task to populate the annotation layer for a given corpus within the text layer. An annotation instance is usually expressed as a pair (*text span*, *descriptor*). The text span is a pointer to a span of a text document that is maintained in the text layer, and the descriptor is the information to be attached to the text span. Usually a set of available descriptors is defined in advance. Since an instance of annotation has the pointer to a text span in the corpus, the annotation layer is dependent on the text layer.

4.3 Annotation Guideline Layer

The guideline layer is the location in which the annotation guidelines are stored (Annotation Guideline Layer in Figure 2). The title and body are an abstract and a detailed description of the guideline, respectively. The keywords are maintained to support better access to the guideline. The list of associated annotations provides easy access to annotation examples of the guideline. Note that an annotation example may be associated to more than one guideline.

4.4 Extension of the annotation layer

With the three-layer model, introduced so far, we can manage guidelines with associated annotation instances. In practice, however, guidelines are often associated with negative annotation instances. We call such a case a negative annotation instance.

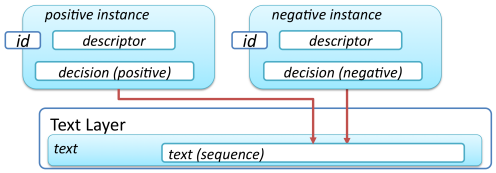

Figure 3: Extended expression of annotation

Although negative annotation instances are not explicitly maintained by usual annotation tools, they are sometimes useful in clarifying the meaning of guidelines, and thus need to be exemplified. To support that, the annotation layer introduced in Section 4.2 needs to be extended. In the extended annotation layer, an annotation is expressed by a triple (*text span*, *descriptor*, *decision*) instead of a pair (Figure 3). The decision is either ”positive,” that the descriptor of attached to the text span), or “negative,” that the descriptor is not attached). Note that any text span without associate annotation may be treated as potentially negative annotation instance. In the proposed framework, only the negative annotation instances which need to be associated with annotation guidelines will be explicitly maintained.

5 Implementation

In this section, we present GuideLink, a guideline management system, which is an implementation of the annotation framework proposed in the previous sections. GuideLink is an add-on to the annotation tool, XConc Suite, which is an implementation of the two-layer model described in Section 4. GuideLink adds the guideline layer and the extended annotation layer. Using GuideLink and XConc Suite, the annotator does not need to switch between two separate systems. GuideLink supports the tasks of step (2), (3), and (5) in the annotation flow model in Figure 1.

Figure 4 shows a snapshot of GuideLink with XConc Suite. Annotation Collection Viewer is to browse the collection of annotation instances. Annotation Viewer/Editor is to view and edit annotation instances in documents. Guideline Collection Viewer is to browse and search guidelines in a collection. Guideline Viewer/Editor is to view and edit individual guidelines. Annotation

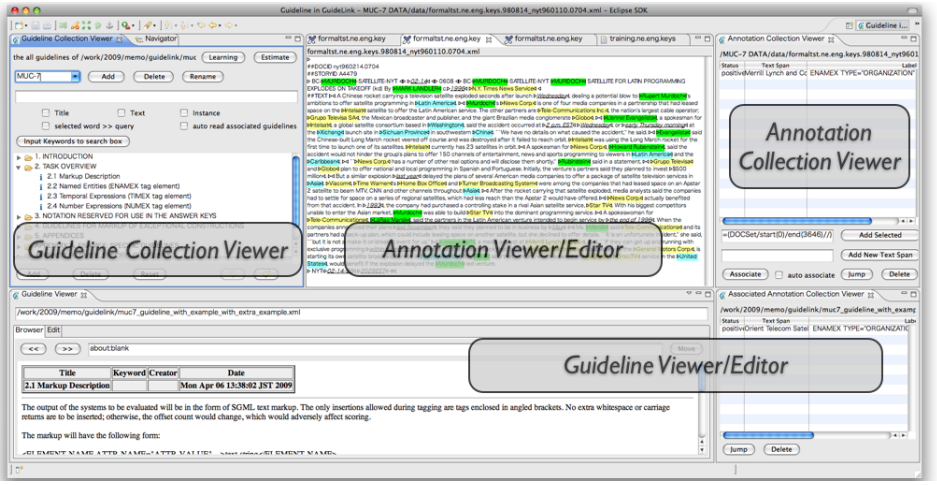

Figure 4: A snapshot of GuideLink plugged in XConc Suite

Viewer/Editor is the annotation window provided by on XConc Suites. The primary goal of the proposed framework is to provide an easy access between annotations and guidelines.

The way to access annotation instances from annotation guidelines is implemented in the Guideline Viewer/Editor and the Annotation Collection Viewer. When a guideline is shown in the Guideline Viewer/Editor, the Annotation Collection Viewer shows the annotation instances that are associated with the guideline. Like this, annotators can quickly access the relevant annotation instances for the guideline.

5.1 Keyword-based guideline search

In order to provide easy access to the desired guidelines, we implemented two methods: keyword-based guideline search and similarity-based guideline suggestion. The first method is traditional keyword-based search. In this method, a user can input a query for searching relevant guidelines. A query is a concatenation of keywords via boolean operators. The system returns retrieved guidelines based on the query.

Since this method has long been a primary search method of general information retrieval systems, there is a high likelihood that many users are already familiar with it and that with some experience users can quickly write effective queries.

5.2 Similarity-based guideline suggestion

The second method is similarity-based guideline suggestion. In this method, the system automatically retrieves guidelines that are determined to be relevant to the text under annotation. To know the text under annotation, the system considers the position of the cursor on the text, and when the cursor stops on a certain word without any input for a certain amount of time, it assumes the word in the that position is considered to be annotated, and tries to retrieve and show relevant guidelines. We implemented a similarity-based method, which calculates the similarity between example to be annotated and the examples that are associated with guidelines.

For the determination of similarity, we used the support vector machine (SVM) and the k-nearest neighbor (KNN) classifiers as implemented in Weka (Witten and Frank, 2005). In our implementation, training a classifier by SVM or KNN is done offline. For the representation of each example, we considered the target word, the preceding three words, and following three words. Each word is expressed by its word form, word shape and part-of-speech. We used OpenNLP tools[3] to get the part-of-speech of words.

[3] http://opennlp.sourceforge.net/

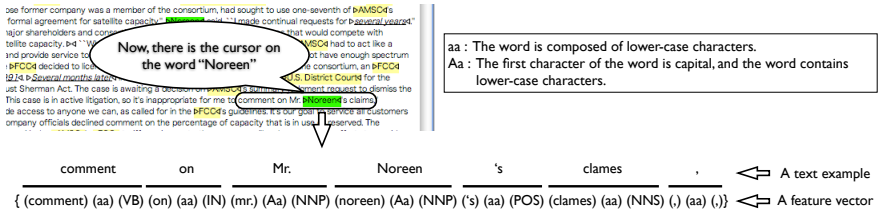

Figure 5: A text example and its feature vector

Table 1: Comparison between an annotation work with and without the proposed system

Step in Figure 1	Conventional Framework	Proposed Framework
(2) Consult applicable guidelines	Use the search functionality of a word processer or Wiki to find guidelines.	Keyword-based guideline search or Similarity-based guideline suggestion.
(3) Solve the problem and develop guidelines	Use, for example, a word processor or Wiki to update guidelines. The guidelines are usually written in a plain text.	Create a record in the guideline layer. The record is written in a structural data.
(5) Associate the annotation example with a guideline	Include the example in the section of related guidelines.	Create a link to the annotation in the annotation layer from the guideline.

Figure 5 shows an example of feature vector generation. When the cursor stops at the word *"Noreen"* in the text *"to comment on Mr. Noreen's claims,"Pemberton said. ,"* the system produce the feature vector from the target word, *"Noreen"* and its surrounding words. We considered the preceding and following three words. Three features are extracted from each word: word form, word shape, and part-of-speech (POS). These features work well for named entity recognition (Finkel *et al.*, 2004). For word shape, we considered 6 types: aa (all lower-case letters), AA (all capital letters), aA (mixed, begins with a lower-case letter), Aa (mixed, begins with a capical letter), Num (all numerical letters) and Num_a (mixed with numerical and alphabet letters).

6 Evaluation

The purpose of this work is to provide an integrated framework for corpus annotation and guideline production, and to improve the accessibility between an annotated corpus and guidelines. In this section, we present two evaluations. The first evaluation is for the advantage of the proposed framework in the actual annotation flowchart. The second evaluation is for the accessibility of relevant annotation guidelines.

6.1 Comparison between conventional and proposed annotation frameworks

Table 1 highlights the differences between the conventional and the proposed annotation frameworks. The step numbers in the table correspond to that of Figure 1. Step (2), (3) and (5) of the annotation work flow described in Section 3 involve development or reference of guidelines.

In the conventional annotation framework, there is no particular system developed for guideline production, and various general documenting tools, e.g. word processors, are used for the development and reference of the guidelines, allowing a conventional way of editing and searching guidelines, as shown in the second column of the table. In the proposed framework, the guidelines are managed in an integrated way with the annotation, as shown in the third column.

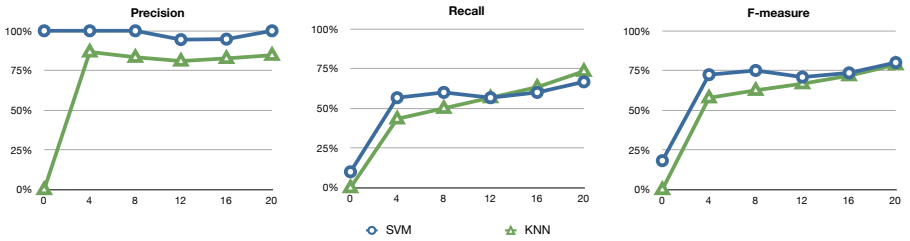

Figure 6: The simulation of auto relevant guideline searching system.

6.2 Evaluation of similarity-based guideline suggestion

We performed a series of experiments to evaluate the performance of the similarity-based guideline suggestion described in the Section 5.2. For the experiment, we import the guidelines for MUC-7 named entity annotation in such a way as to copy manually all content of guidelines for the MUC-7 and to paste the 76 guidelines for the MUC-7 named entity annotation are all imported in GuideLink. For the determination of relevant guidelines, SVM and KNN are used as implemented in Weka. With SVM, a binary classifier is developed for each guideline to determine if the guideline is relevant or not to the given example to be annotated. With KNN, the annotation examples associated with guidelines are searched for similar ones to the given example. We empirically decided the threshold to be 0.1. Since each MUC-7 guideline is accompanied with annotation examples, they are used as positive example of the guideline. As negative training examples, the annotation examples of the other guidelines are used together with randomly selected 1000 examples from the MUC corpus. An expected problem of this method is the small number of positive examples: Most guidelines are accompanied with only a few examples. With the initial set of training samples, we cannot expect to get a good performance. The performances were 18% (SVM) and 0% (KNN) in f-measure.

We then performed more experiments to see the effect of increasing the number of positive examples. Although we could not expect a good performance of automatic guideline suggestion at the initial stage of annotation, as the annotation proceeds, the number of positive examples increases and the performance improve. For the experiments, we chose following three guidelines (A.3.1[4], B.3[5], and C.1[6]) for which the system showed poor suggestion performance in the initial experiment. For each of them, we manually added positive examples up to 20. For the evaluation of performance, we prepared 40 test examples: 10 that are relevant to each of A.3.1, B.3, and C.1, respectively, and 10 that are not relevant to any of the three.

Figure 6 shows the performance change of the guideline suggestion as the number of positive training examples increases, in terms of precision, recall, and f-measure, respectively. The horizontal axis is the number the positive examples, which we add to each guideline of the three.

It is observed that the performance improves as the number of positive example increases. SVM-based guideline suggestion showed good performance in terms of precision, while KNN showed good recall. Although the experiments have been performed in a very limited way, using only a few guidelines and examples, the final results show good potential for the automatic guideline suggestion, approaching 80% f-measure.

[4] A.3.1 Titles vs. Generational Designators (Titles such as “Mr.” and role names such as “President” are *not* considered part of a person name. However, appositives such as “Jr.,” “Sr.,” and “III” *are* considered part of a person name.)

[5] B.3 Temporal Expressions Containing Adjacent Absolute and Relative Strings (When a time expression contains both relative and absolute elements, the entire expression is to be tagged. The following examples illustrate some of the ways in which elements of relative and absolute time expressions may combine to form taggable time expressions.)

[6] C.1 Scope of Numeric Expressions (The entire string expressing the monetary or percentage value is to be tagged.)

7 Conclusion

Although it is generally understood that the development of annotation guidelines is critical in maintaining annotation consistency and in understanding the annotation results, the guideline production process has been treated as secondary work and separated from the annotation process. In this paper, we presented a framework where guideline production is integrated into the annotation process. Qualitative comparison with a conventional framework showed that guidelines can be systematically produced and maintained during the annotation process using the proposed framework, and that the relevant guidelines can be efficiently accessed by the annotators using the search system implemented based on the framework.

References

Alphonse, Erick, Sophie Aubin, Gilles Bisson, Thierry Hamon, Rine Lagarrigue, Adeline Nazarenko, Alain-pierre Manine, Claire Nédellec, Mohamed Ould, Abdel Vetah, Thierry Poibeau and Davy Weissenbacher. 2004. Event-Based Information Extraction for the Biomedical Domain: The Caderige Project. *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*.

Chinchor, N. and P. Robinson. 1997. *MUC-7 Named Entity Task Definition (version 3.5)* http://www.itl.nist.gov/iad/894.02/related_projects/muc/proceedings/ne_task.html

Cunningham, H., D. Maynard, K. Bontcheva and V. Tablan. 2002 GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

Finkel, Jenny, Shipra Dingare, Huy Nguyen, Malvina Nissim, Christopher Manning and Gail Sinclair. 2004. Exploiting context for biomedical entity recognition: from syntax to the web. *JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, 88-91.

Kulick, Seth, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, Lyle Ungar, Scott Winters and Pete White. 2004. Integrated Annotation for Biomedical Information Extraction. In *Proceedings of HLT-NAACL 2004 Workshop: Biolink 2004, Linking Biological Literature, Ontologies and Databases*, 61-68.

Lu, Z., M. Bada, P. V. Ogren, K. B. Cohen and L. Hunter. 2006. Improving biomedical corpus annotation guidelines. *Proceedings of the Joint BioLINK and 9th Bio-Ontologies Meeting*, 89-92.

Marcus, M.P, B. Santorini and M. A. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Tree Bank. *Computational Linguistics*, VOLUME 19, NUMBER 2, 313-330.

Morton, Thomas and Jeremy LaCivita. 2003. WordFreak: An Open Tool for Linguistic Annotation. *Proceedings of HLT-NAACL*, 17-18.

Mueller, Christoph and Michael Strube. 2001. MMAX: A Tool for the Annotation of Multi-modal Corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, 45-50.

Ogren, P. V. 2006. Knowtator: a plug-in for creating training and evaluation data sets for biomedical natural language systems. In *Proceedings of the 9th International Protégé Conference*, 73-76.

Sampson, G. 2002. English for the Computer: The SUSANNE Corpus and Analytic Scheme. *Computational Linguistics*, VOLUME 28, NUMBER 1, 102-103.

Witten, Ian H. and Eibe Frank 2005. Data Mining: Practical machine learning tools and techniques (Second Edition).