IJCLA VOL. 3, NO. 2, JUL-DEC 2012, PP. 93–104
RECEIVED 04/11/11 ACCEPTED 09/12/11 FINAL 01/10/12

Using the ILCI Annotation Tool for POS Annotation: A Case of Hindi

RITESH KUMAR, SHIV KAUSHIK, PINKEY NAINWANI, ESHA BANERJEE, SUMEDH HADKE, GIRISH NATH JHA

Jawaharlal Nehru University, New Delhi

ABSTRACT

In the present paper, we present an annotation tool, ILCIANN (Indian Languages Corpora Initiative Annotation Tool), which could be potentially used for crowd-sourcing the annotation task and creation of language resources for use in NLP. This tool is expected to be especially helpful in creating annotated corpora for the less-resourced languages. ILCIANN is a server-based web application which could be used for any kind of word-level annotation task in any language. In the paper a description of the architecture of the tool, its functionality, its application in the ILCI (Indian Languages Corpora Initiative) project for POS annotation of Hindi data and the extent to which it increases the efficiency and accuracy of the annotators is given. It describes the results of an experiment conducted to understand the increase in the efficiency (in terms of time spent on annotation) and the reliability (in terms of the inter-annotator agreement) with the use of the tool when compared to the manual annotation.

KEYWORDS: ILCIANN, ILCI, POS annotation, server-based annotation, Hindi POS annotation

1 INTRODUCTION

ILCIANN is a server-based web application which could be used for any kind of word-level annotation task in any language. It is developed using Java/JSP as the programming language and is running on Apache

Tomcat 4.0 web server. It is meant to facilitate the job of manual annotation (and not be a tagger in itself) by providing a user interface. It also provides the facility of limited automatic tagging for closed grammatical categories like pronouns, postpositions, conjunctions and quantifiers which reduces the burden of human annotators.

Some other annotation tools have been developed for similar purposes. Bird et al. [6] came up with a tool which targeted at facilitating the development of linguistic annotations called Atlas (Flexible and Extensible Architecture for Linguistic Annotations). It consists of three levels:

1. The logical level: defines a set of procedures for creating, modifying, searching, and storing well-formed annotation sets
2. The physical level: free to access in various ways- via networked client server modes , or via linked libraries into application binaries, or via scripting languages
3. The application level: reduces the burden of human annotators and also language engineering application development.

Though the tool is comprehensive in nature but it works best for speech database and corpus.

Kaplan et al. [7] designed a web based annotation tool (SLATE: Segment and Link-based Annotation Tool Enhanced), which addresses ten major annotation needs:

1. Managing the role of annotator and administrator,
2. Delegation and monitoring work,
3. Adaptability to new annotation tasks,
4. Adaptability within the current annotation task,
5. Diffing and merging (diffing and merging of data from multiple annotators on a single resource to create a gold standard),
6. Versioning of corpora,
7. Extensibility in terms of layering,
8. Extensibility in terms of tools,
9. Extensibility in terms of importing/exporting and,
10. Support for multiple languages.

This tool to a great extent addresses to the purpose of the management of large and parallel data but it does not address the issue of the annotation of translated parallel corpora.

2. THE ILCIANN TOOL

The tool is being developed and currently used for POS annotation in the Indian Languages Corpora Initiative (ILCI) project funded by the Department of Information Technology (DIT), Govt. of India ([3, 4]). The first phase of the project involved developing a POS annotated parallel translated corpus of 50,000 sentences in 12 major Indian languages (which included Hindi, Urdu, Bangla, Oriya, Punjabi, Gujarati, Marathi, Konkani, Telugu, Tamil, Malayalam and English). It is a consortium project running parallel in 10 different universities of India spread across the country. The basic corpus was prepared in Hindi, which was translated in 10 other languages to prepare the parallel corpus. Once the corpus creation was complete, the data had to be annotated with labels for part of speech (POS) using the BIS tagset (a newly framed tagset, approved by Bureau of Indian Standards (BIS), which is now the national standard and supposed to be used in all kinds of POS annotation work across the nation).

In order to manage the whole process of annotation in such a way that it could be done efficiently and with minimum errors, the ILCI Annotation Tool (ILCIANN) is being used. The use of the tool ensured that the data is saved in a centralized server in a uniform format which could be later utilized for any NLP task without much need of pre-processing or noise cleaning.

The following sections describe the architecture and working of the tool.

2.1 *Architecture of the Tool*

2.1.1 *Module 1 (Admin Module)*

This is the module where all the administrative work related to any annotation project is carried out. The following steps are carried out in this module (and they are the most basic steps that need to be taken before starting any annotation project and during the project also)

1. **Step 1 (Creating the user login)**: This step involves creating the login of users who would annotate the data. The project administrator has the authority to create the login for the number of specific human annotators who want to annotate/tag the data. It

ensures the safety as well as authenticity of the tagged data, while theoretically giving an opportunity to a huge community to support and help in building language resources for their language. Moreover if the annotation project involves more than one language then the user is also assigned the language on which (s)he is supposed to work. For instance, if x is Hindi language annotator in a multi-language project, (s)he can only work on Hindi data and cannot do any modification (tagging the data, editing the data and saving it) in other language files. Furthermore, each user is assigned a set of maximum 3 files for annotation at one time (and a new file is assigned only after one of the files is completed) to ensure that multiple users do not work on same file (which also helps in keeping a record of the progress of the individual annotators) and also that one or more files are not left incomplete.

2. **Step 2 (Uploading the Files)**: This step involves uploading various files which would be used for the annotation and include the data files which need to be tagged, the tagset which is to be used and a file called the *autotag* file. The autotag file consists of a list of words (which belong to closed grammatical category) and their POS label. This file is used by the tool to tag the function words automatically.
3. **Step 3 (Monitoring the Progress)**: The admin could also monitor the progress of each and every user in his/her project. The information includes the number of files completed by each user, the name of the files assigned to each user, the files on which each user is currently working, etc.
4. **Step 4 (Downloading the Files)**: The file is ready for download only when each sentence of the file is tagged. Downloading the completed file is optional and only the administrator of the project has the right to download these files.

2.1.2 *Module 2 (Annotation Module)*

1. **Step 1 (Selection)**: After the user logged in, the left hand side of the page shows two options: select the file and sentence id. The user is required to select the file in which (s)he wants to work. Once the file gets selected, the untagged sentence immediately appears. . Further, if the user wants to do some modifications in previously tagged sentences, (s)he can do it with the option of “select a sentence id”. The right hand side of the page shows the

progress of tagging status i.e. number of completed tagged sentences and also completed files.

2. **Step 2 (Editing/Segmentation)**: This step is optional. The user uses this button only when there is some error in the original data which needs to be corrected.
3. **Step 3 (Annotation):** This is the major step in the tool. As "tag the sentence" button is clicked, each word of the sentence with the default tag (the first tag in the tagset) appears except for the words which are automatically tagged. As mentioned above, to minimize the human efforts, the ILCIANN tool automatically tags closed categories like pronouns, postpositions, quantifiers, symbols and punctuations. These automatically tagged words are not frozen, as we know that part-of-speech is purely contextual, therefore, one may want to do further modifications on automatically tagged words also if (s)he finds it inappropriate according to the context, (s)he has the option to do so. Words which are not tagged, the user selects the appropriate tag from the given tagset list.
4. **Step 4 (Saving)**: After assigning the appropriate tag to each word, there is the button of "save" which saves the tagged sentence. The whole file cannot be saved in one go, each and every sentence needs to be saved individually. The saved tagged sentence is stored on the server in the format of "sentence id" and respective "tagged sentence".

2.1.3. *Module 3 (Statistics Module)*

1. **Information 1 (File Information):** This includes information regarding the number of files completed and the number of files on which work is in process.
2. **Information 2 (Sentence Information)**: The information regarding the number of sentences completed in the present file and in the whole corpus, and also the speed of annotation of each user (in terms of sentences per minute) is included here.

2.2 *Using the Tool: POS Annotation in ILCI*

There are three levels of users of this tool:

1. **Administrator (Admin)**: For the purpose of management, each language is assigned an administrator user account or the Admin

account. The Admin has a username and password, which he or she uses to access his/her account. It is in the Admin's jurisdiction to assign annotation work to as many Users as is required, the language in which annotation work will be carried out as well as up to 3 sets within each language group. The tasks of the Admin include maintaining the log of user details, tagging status and downloading completed files.

2. **User**: The User is assigned a username, password and language. The User, on entering this information in the Login page is directed to the main Home page of the tool, wherein the sets that he or she is assigned are displayed. The User selects the set number and the sentence ID which (s)he wants to work on. In case there is a need for correction within the displayed sentence, the User uses the Segment button to insert or delete additional information, such as white space removal, hyphen insertion etc. Once the sentence is ready for tagging, the User clicks on Tag the Sentence button. On clicking the button, each word of the sentence is displayed separately with the tagset in a drop-down box format beside each word. The User selects the appropriate tag for each word and tags the sentence. On completion, the sentences, along with the tags, are saved with the help of Save button. On completion of work, the User logs out using the Logout button.
3. **Master Admin**: The Master Admin also has a Username and password, which he or she uses to access his/her account. In addition to the normal tasks of the Admin, the Master Admin can also maintain the time log of the user accounts and create, delete, or change passwords of user accounts.

3 EFFICIENCY AND RELIABILITY OF THE TOOL

In order to understand the efficiency and reliability of the tool, an experiment was conducted with the help of three annotators. Each annotator was given two sets of data, each containing around 500 words (a total of 45 sentences). These sentences were taken from the ILCI corpora and contained almost equal number of words from both the health and tourism domain. The annotators were required to annotate the words manually (in a text file, without using any kind of tool), using the tool without intelligence and using the tool with intelligence. While the first set of 500 words were same across all these methods of annotation, the second set of 500 words were different

across all these methods. As is common practice in such experiments, the annotators were not allowed to consult each other during the annotation period. The experiments were conducted over a period of 6 days, with a gap of one day in between the annotation by each method (to reduce the bias in the common set). The time taken by each annotator in annotating each set by each method was noted down. Also the tagged data is being used to calculate the inter-annotator agreement in order to see if the tool also increases the reliability of the annotation process.

3.1 *Calculating the Efficiency*

Table 1 gives the time taken by each annotator in annotating each set by each method.

Table 1. Comparison of time taken in annotation (in minutes)

	Manual		Not intelligent		Intelligent	
Sets	A	B	A	B	A	B
Annotator A	55	50	30	35	15	15
Annotator B	32	36	22	25	18	17
Annotator C	125	97	29	33	24	16

As we could clearly see the tool (without any intelligence) has led to almost 100% increase in the efficiency of annotator A (for set A). While for others also there is an increase of around 50% in the efficiency of annotator A and B. While for annotator C, we see that the speed (which was very slow when the annotation was carried out manually) has increased tremendously and has come at par with the other two annotators. Moreover when we impart some intelligence to the system, we again see an increase of almost 50% in the efficiency of annotator A; while there is a marked increase in the speed of other annotators also. This efficiency could be further increased by imparting more intelligence to the machine. It must be noted that the intelligence, at present, is given to the machine by way of an *autotag* file which consists of a list of word with the tag that should be given to it. This file is prepared manually and contains those words which always takes only one tag irrespective of the context (mainly function words; but it also has some content words). At a later stage the tool will be equipped

with machine learning algorithms so that it becomes a POS tagger in effect and it could auto-tag most of the words and the user's effort remains only in revising the annotated data.

3.2 *Calculating the Reliability*

Several methods (discussed in detail) are used to compute the reliability (or, inter-annotator agreement) of any annotation work. Some of the major ones include the following.

Percentage Agreement (also called observed agreement, defined by Scott, 1955) is one of the simplest and earliest measures of inter-annotator agreement where the percentage of agreements between two annotators is calculated.

Cohen's kappa coefficient [1] is one of the best-known statistical measures of inter-rater agreement or *inter-annotator agreement* (IAA) for qualitative items. It is generally thought to be a more robust measure than simple percent agreement calculation since K takes into account the agreement occurring by chance. Cohen's kappa measures the agreement between two raters and each classifies N items into C mutually exclusive categories. The equation for K is

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)},$$

where Pr(a) is the relative observed agreement among raters, and Pr(e) is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly saying each category. If the raters are in complete agreement then K = 1. If there is no agreement among the raters other than what would be expected by chance, K = 0.

Scott's pi [5] is a statistic for measuring inter-rater reliability for nominal data. Scott's pi is similar to Cohen's kappa in that they improve on simple observed agreement by factoring in the extent of agreement that might be expected by chance. On the other hand Scott's pi makes the assumption that annotators have the same distribution of responses, which makes Cohen's kappa slightly more informative. The equation for Scott's pi, as in Cohen's kappa is:

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)};$$

however, Pr(e) is calculated using joint proportions.

Fleiss' Kappa [2] is a generalization of Scott's pi statistic. It is a statistical measure for assessing the reliability of agreement between a fixed number of raters when assigning categorical ratings to a number of items or classifying items. It works for any number of raters giving categorical ratings to a fixed number of items unlike Cohen's kappa and Scott's pi. It can be interpreted as expressing the extent to which the observed amount of agreement among raters exceeds what would be expected if all raters made their ratings completely randomly. Fleiss' kappa specifically assumes although there are a fixed number of raters (e.g., three), different items are rated by different individuals (Fleiss, 1971, p.378). If a fixed number of people assign numerical ratings to a number of items then the kappa will give a measure for how consistent the ratings are. The kappa, K, can be defined as:

$$\kappa = \frac{\overline{P} - \overline{P}_e}{1 - \overline{P}_e}.$$

The factor gives the degree of agreement that is attainable above chance, and $\overline{P} - \overline{P}_e$ gives the degree of agreement actually achieved above chance. If the raters are in complete agreement then K = 1. If there is no agreement among the raters then K = 0.

For the present purposes, Cohen's Kappa and Scott's Pi are not very relevant since the experiment involved more than two annotators. However we have calculated both the percentage and the Fleiss' Kappa so that the agreement measure of both kinds (taking chance into account and without taking chance into account) is calculated.

3.3 *Calculating Percentage Agreement*

The simple percentage of agreements among the three pairs of annotators is summarised in Table 2. It is calculated using the simple formula of percentage: sum of agreed instances × 100 / total number of instances.

While the inter-annotator agreement between annotators A and B is already on the higher side of the spectrum, it does not improve much with the use of the tool and it seems that the other factors (like the tagset itself, the guidelines, annotators' expertise, etc.) are playing a vital role here. However the situation is quite different in case of

agreement between annotators B and C and that between A and C where the inter-annotator agreement in case of manual annotation is pretty low. The agreement between the annotators improves quite considerably with the use of the tool. The intelligence of the tool also seems to be playing some role in the improvement of the inter-annotator agreement.

Table 2. Percentage of agreement among three pairs of annotators (%)

	Manual		Not intelligent		Intelligent	
Sets	A	B	A	B	A	B
Annotators A and B	85	87	84	83	90	87
Annotators B and C	66	77	81	81	83	85
Annotators A and C	67	72	76	81	81	80

3.4 *Calculating Fleiss' Kappa*

As mentioned earlier Fleiss' Kappa is a generalization over Scott's pi to calculate the inter-annotator agreement among more than 2 annotators. Since the present experiment involved three annotators, Fleiss' Kappa was also calculated (which is generally considered more reliable and accurate than percentage calculation). In order to arrive at a better picture vis-a-vis the percentage agreement as well as see if the overall agreement is affected by one annotator, both the inter-annotator agreement in between each pair of annotators as well as the overall agreement is also estimated. The values of Fleiss' Kappa for each pair of annotator in each set and also the general values for all the sets taken together is summarised in Table 3.

These values of Fleiss' reaffirm the facts that were shown by the percentage calculation of the agreements. The tool seems to be making only a small contribution to an increase in the reliability of the annotation at the present stage. However when we look at the overall result, we see a steady increase in the reliability (or, inter-annotator agreement) of the annotation efforts as we move from manual annotation to annotation using the tool to annotation using the tool with some limited intelligence.

Table 3. Calculated values of Fleiss' Kappa

Annotators	Manual		Not intelligent		Intelligent	
Sets:	A	B	A	B	A	B
A and B	0.852	0.871	0.829	0.820	0.895	0.881
B and C	0.698	0.786	0.794	0.796	0.814	0.867
A and C	0.719	0.732	0.731	0.803	0.789	0.819
A, B and C	0.757	0.797	0.785	0.806	0.833	0.856
A, B and C	0.777		0.797		0.845	

4 CONCLUSIONS

In the present paper, we have described the working of an online annotation tool, ILCIANN, which is meant not only to facilitate the task of manually annotating the data but also increase the overall efficiency (by considerably reducing the time taken in the annotation work) and the reliability (by increasing the inter-annotator agreement) of the annotation task. The experiments conducted to know the exact nature of efficiency and reliability has clearly shown that both of these attributes increase as the intelligence of the tool increases. Since the tool is developed in such a way that it could become more intelligent as more annotation takes place, the tool is expected to work in a much better way as the time passes and it could prove to be a very useful resource for the development of language resources for all kinds of language, especially the less-resourced ones.

REFERENCES

1. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement. 20(1), 37–46 (1960)
2. Fleiss, J. L.: Measuring nominal scale agreement among many raters. Psychological Bulletin. 76(5), 378–382 (1971)
3. Jha, G. N.: Indian Language Corpora Initiative (ILCI). Invited talk, 4th Intern. Language and Technology Conf. (4th LTC), Poland (2009).
4. Jha, G. N.: The TDIL program and the Indian Language Corpora Initiative (ILCI). In: Proceedings of the Seventh International conference on Language Resources and Evaluation (LREC'10), pp. 982-985 (2010)
5. Scott, W.: Reliability of content analysis: The case of nominal scale coding. Public Opinion Quarterly. 19(3), 321–325 (1955)
6. Bird, S., David D., Garofolo, J. S., Henderson, J., Laprun, C., Liberman, M.: Atlas: A flexible and extensible architecture for linguistic annotation. CoRR, cs.CL/0007022 (2000)

7. Kaplan, D., Iida, R., Tok, T.: Annotation Process Management Revisited. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10), pp. 3654-3661 (2010)

RITESH KUMAR
CENTRE FOR LINGUISTICS,
JAWAHARLAL NEHRU UNIVERSITY,
NEW DELHI, INDIA.
E-MAIL: <RITESH78_LLH@JNU.AC.IN>

SHIV KAUSHIK
SPECIAL CENTRE FOR SANSKRIT STUDIES,
JAWAHARLAL NEHRU UNIVERSITY,
NEW DELHI, INDIA
E-MAIL: <SHIVKAUSHIK.ENGG@GMAIL.COM>

PINKEY NAINWANI
CENTRE FOR LINGUISTICS,
JAWAHARLAL NEHRU UNIVERSITY,
NEW DELHI, INDIA
E-MAIL: <PINKEYBHU39@GMAIL.COM>

ESHA BANERJEE
CENTRE FOR LINGUISTICS,
JAWAHARLAL NEHRU UNIVERSITY,
NEW DELHI, INDIA
E-MAIL: <ESHA.JNU@GMAIL.COM>

SUMEDH HADKE
CENTRE FOR INDIAN LANGUAGES,
JAWAHARLAL NEHRU UNIVERSITY,
NEW DELHI, INDIA
E-MAIL: <SUMEDHKHADKE@GMAIL.COM>

GIRISH NATH JHA
SPECIAL CENTRE FOR SANSKRIT STUDIES,
JAWAHARLAL NEHRU UNIVERSITY,
NEW DELHI, INDIA
E-MAIL: <GIRISHJHA@GMAIL.COM>