Complex Linguistic Annotation – No Easy Way Out! A Case from Bangla and Hindi POS Labeling Tasks

Sandipan Dandapat[1] Dublin City University Ireland

Priyanka Biswas[1] LDCIL CIIL-Mysore, India

Monojit Choudhury **Kalika Bali** Microsoft Research Labs India Bangalore, India

E-mail: sdandapat@computing.dcu.ie, biswas.priyanka@gmail.com, monojitc@microsoft.com, kalikab@microsoft.com

Abstract

Alternative paths to linguistic annotation, such as those utilizing games or exploiting the web users, are becoming popular in recent times owing to their very high benefit-to-cost ratios. In this paper, however, we report a case study on POS annotation for Bangla and Hindi, where we observe that reliable linguistic annotation requires not only expert annotators, but also a great deal of supervision. For our hierarchical POS annotation scheme, we find that close supervision and training is necessary at every level of the hierarchy, or equivalently, complexity of the tagset. Nevertheless, an *intelligent annotation tool* can significantly accelerate the annotation process and increase the inter-annotator agreement for both expert and non-expert annotators. These findings lead us to believe that reliable annotation requiring deep linguistic knowledge (e.g., POS, chunking, Treebank, semantic role labeling) requires expertise and supervision. The focus, therefore, should be on design and development of appropriate annotation tools equipped with machine learning based predictive modules that can significantly boost the productivity of the annotators.

1 Introduction

Access to reliable annotated data is the first hurdle encountered in most NLP tasks be it at the level of Parts-of-Speech (POS) tagging or a more complex discourse level annotation. The performance of the machine learning approaches which have become de rigueur for most NLP tasks are dependent on accurately annotated large datasets. Creation of such databases is, hence, a highly resource intensive task both in terms of time and expertise.

While the cost of an annotation task can be characterized by the number of man-hours and the level of expertise required, the productivity or the benefit can be measured in terms of the reliability and usability of the end-product, i.e., the annotated dataset. It is thus no surprise that considerable effort has gone into developing techniques and tools that can effectively boost the benefit-to-cost ratio of the annotation process. These include, but are not limited to:

(a) exploiting the reach of the web to reduce the effort required for annotation (see, e.g., Snow et al. (2008) and references therein)

(b) smartly designed User Interfaces for aiding the annotators (see, e.g., Eryigit (2007); Koutsis et al. (2007); Reidsma et al. (2004))

(c) using supervised learning to bootstrap a small annotated dataset to automatically label a larger corpus and getting it corrected by human annotators (see, e.g., Tomanek et al. (2007); Wu et al. (2007))

(d) Active Learning (Ringger et al. 2007) where only those data-points which are directly relevant for training are presented for manual annotation.

Methods exploiting the web-users for linguistic annotation are particularly popular these days, presumably because of the success of the ESP-Game (von Ahn and Dabbish, 2004) and its successors in image annotation. A more recent study by (Snow et al., 2008) shows that annotated data obtained from non-expert anonymous web-users is as good as those obtained from experts. However, unlike the game model, here the task is distributed among non-experts through an Internet portal such as Amazon Mechanical Turk, and the users are paid for their annotations.

This might lead to an impression that the expert knowledge is dispensable for NLP annotation tasks. However, while these approaches may work for more simple tasks like those described in (Snow et al., 2008), most NLP related annotation tasks such as POS tagging, chunking, semantic role labeling, Treebank annotation and

[1] This work has been done during the authors' internship at Microsoft Research Lab India.

Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP 2009, pages 10–18,
Suntec, Singapore, 6-7 August 2009. ©2009 ACL and AFNLP

discourse level tagging, require expertise in the relevant linguistic area. In this work, we present a case study of POS annotation in Bangla and Hindi using a hierarchical tagset, where we observe that reliable linguistic annotation requires not only expert annotators, but also a great deal of supervision. A generic user interface for facilitating the task of hierarchical word level linguistic annotation was designed and experiments conducted to measure the *inter-annotator agreement* (IA) and annotation time. It is observed that the tool can significantly accelerate the annotation process and increase the IA. The productivity of the annotation process is further enhanced through bootstrapping, whereby a little amount of manually annotated data is used to train an automatic POS tagger. The annotators are then asked to edit the data already tagged by the automatic tagger using an appropriate user interface.

However, the most significant observation to emerge from these experiments is that irrespective of the complexity of the annotation task (see Sec. 2 for definition), language, design of the user interface and the accuracy of the automatic POS tagger used during bootstrapping, the productivity and reliability of the expert annotators working under close supervision of the dataset designer is higher than that of non-experts or those working without expert-supervision. This leads us to believe that among the four aforementioned approaches for improving the benefit-to-cost ratio of the annotation tasks, solution (a) does not seem to be the right choice for involved linguistic annotations; rather, approaches (b), (c) and (d) show more promise.

The paper is organized as follows: Section 2 provides a brief introduction to IL-POST – a hierarchical POS Tag framework for Indian Languages which is used for defining the specific annotation tasks used for the experiments. The design and features of the data annotation tool are described in Section 3. Section 4 presents the experiments conducted for POS labeling task of Bangla and Hindi while the results of these experiments are discussed in Section 5. The conclusions are presented in Section 6.

2 IL-POST

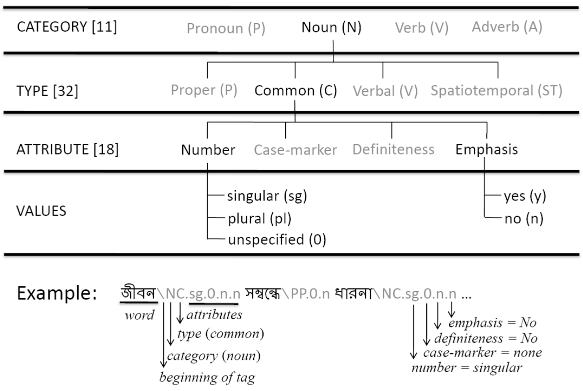
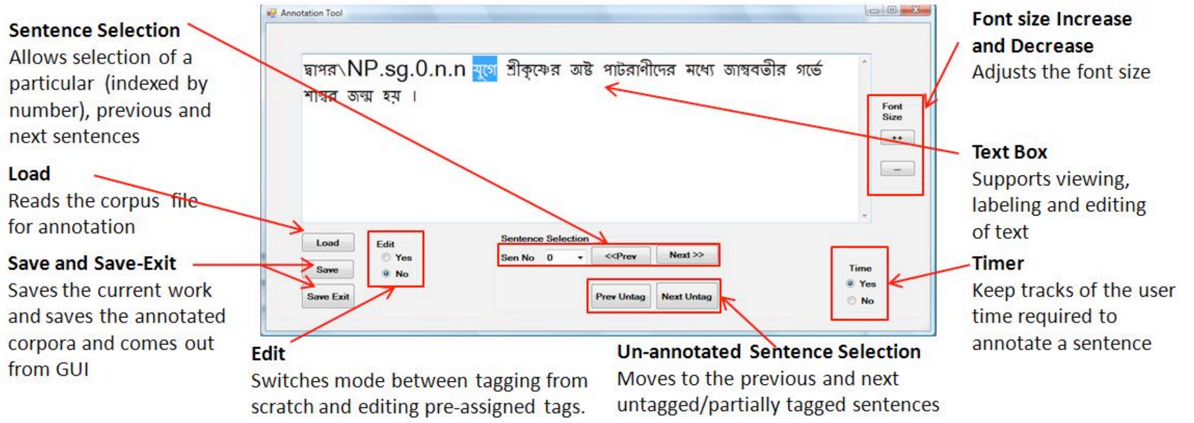


Figure 1: A schematic of IL-POST framework

IL-POST is a POS-tagset framework for Indian Languages, which has been designed to cover the morphosyntactic details of Indian Languages (Baskaran et al. 2008). It supports a three-level hierarchy of Categories, Types and Attributes that provides a systematic method to annotate language specific categories without disregarding the shared traits of the Indian languages. This allows the framework to offer flexibility, cross-linguistic compatibility and reusability across several languages and applications. An important consequence of its hierarchical structure and decomposable tags is that it allows users to specify the morpho-syntactic information applicable at the desired granularity according to the specific language and task. The complete framework supports 11 categories at the top level with 32 types at the second level to represent the main POS categories and their sub-types. Further, 18 morphological attributes or features are associated with the types. The framework can thus, be used to derive a flat tagset of only 11 categories or a complex three level tagset of several thousand tags depending on the language and/or application. Figure 1 shows a schematic of the IL-POST framework. The current framework has been used to derive maximally specified tagsets for Bangla and Hindi (see Baskaran et al. (2008) for the descriptions of the tagsets), which have been used to design the experiments presented in this paper.

3 Annotation Tool

Though a number of POS annotation tools are available none are readily suitable for hierarchical tagging. The tools from other domains (like discourse annotation, for example) that use hierarchical tagsets require considerable customization for the task described here. Thus, in order to facilitate the task of word-level linguistic annotation for complex tagsets we developed a generic annotation tool. The annotation tool can be customized to work for any tagset that has up to



দ্বাপর\NP.sg.0.n.n যুগে শ্রীকৃষ্ণের অষ্ট পাটরাণীদের মধ্যে জাম্ববতীর গর্ভে শাম্বর জন্ম হয় ।

Sentence Selection
Allows selection of a particular (indexed by number), previous and next sentences

Load
Reads the corpus file for annotation

Save and Save-Exit
Saves the current work and saves the annotated corpora and comes out from GUI

Edit
Switches mode between tagging from scratch and editing pre-assigned tags.

Un-annotated Sentence Selection
Moves to the previous and next untagged/partially tagged sentences

Font size Increase and Decrease
Adjusts the font size

Text Box
Supports viewing, labeling and editing of text

Timer
Keep tracks of the user time required to annotate a sentence

Figure 2: The basic Interface Window and Controls. See the text for details.

three levels of hierarchy and for any word level linguistic annotation task, such as *Named Entity* annotation and *Chunk boundary* labeling. In this section we describe the design of the user interface and other features of the annotation tool.

3.1 Interface Design Principles

The annotation scheme followed for linguistic data creation is heavily dependent on the end-application the data will cater to. Moreover, annotations are often performed by trained linguists who, in the Indian context, are either novice or intermittent users of computer. These observations led us to adopt the following principles: (1) customizability of the interface to any word level annotation task; (2) mouse driven selection of tags for faster and less erroneous annotation; and (3) display of all possible choices at every stage of the task to reduce memorization overload.

3.2 Basic Interface

Figure 2 depicts the basic interface of the annotation tool

3.2.1 Automatic Handling

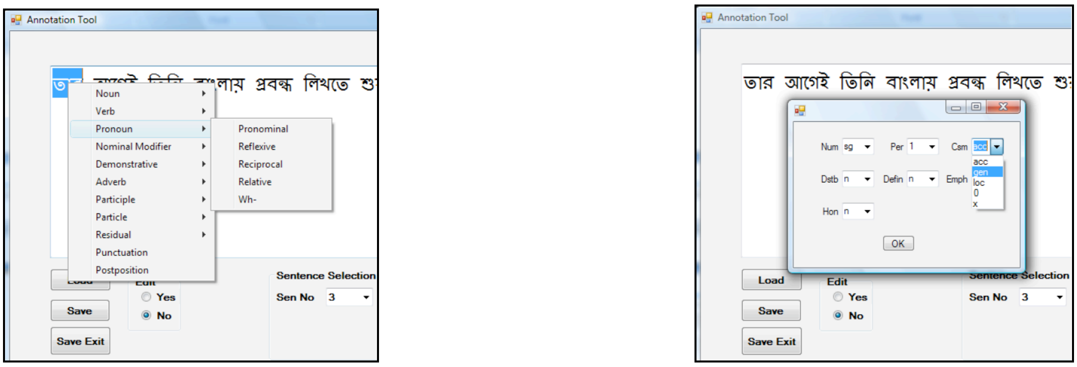
Apart from the surface controls, the interface also supports automatic selection facility that highlights the next unlabeled word that needs to be annotated. After loading the task (i.e., a sentence) it automatically highlights the first unlabeled word. Once a tag is assigned to the highlighted word, the next unlabeled word is automatically selected. However, the automatic selection module can be stopped by selecting a particular word through a mouse click.

3.2.2 Handling Hierarchical Annotation

The first two levels of the IL-POST hierarchy are displayed (on a right mouse click) as a two level context menu. This is illustrated in Fig. 3(a). On selection of the category and type by left clicks, a window is dynamically generated for the assignment of the attribute values, i.e., the third level of the hierarchy. A drop down box is associated with each attribute for selecting the appropriate values. This is shown in Fig. 3(b). The default values for each of the attributes are set based on the frequency of occurrence of the values in a general corpus. This further reduces the time of tag assignment. When the user clicks "OK" on the attribute assignment window, the system automatically generates the tag as per the user's selection and displays it in the *Text-box* just after the selected word.

3.3 Edit Mode Annotation

While performing the annotation task, human annotators need to label every word of a sentence. Instead of annotating every word from scratch, we incorporate machine intelligence to automatically label every word in a sentence. Suppose that we have an automatic POS tag prediction module that does a fairly accurate job. In that case, the task of annotation would mean editing the pre-assigned tags to the words. We hypothesize that such an editing based annotation task that incorporates some intelligence in the form of a tagger will be much faster than purely manual annotation, provided that the pre-assigned tags are "sufficiently accurate". Thus, human annotators only need to edit a particular word whenever machine assigns an incorrect tag making the process faster. We also make certain changes to the basic interface for facilitating easy editing. In particular, when the corpus is loaded using the interface, the predicted tags are shown for each word and the first *category-type* is highlighted automatically. The user can navigate



তার আগেই তিনি বাংলায় প্রবন্ধ লিখতে শু

Edit: ○ Yes ● No

Sen No 3

(a)

Num sg; Per 1; Csm acc; Dstb n; Defin n; Emph; Hon n

(b)

Figure 3: Annotation at a) Category-Type level, b) Attribute level

to the next or pervious editable positions (*Category-Type* or *Attributes*) by using the *Shift* and the *Ctrl* keys respectively. The user may edit a particular pre-assigned tag by making a right mouse click and choosing from the usual context menus or attribute editing window. The user also has the provision to choose an editable location by left mouse-click.

3.3.1 Automatic POS Tagger

We developed a statistical POS tagger based on *Cyclic Dependency Network* (Toutanova et al., 2003) as an initial annotator for the Edit mode annotation. The tagger was trained for Bangla and Hindi on the data that was created during the first phase of annotation (i.e. annotation from scratch). We developed taggers for both *Category+Type* level (CT) and *Category+Type+ Attribute* level (CTA). We also developed two versions of the same tagger with *high* and *low* accuracies for each level of the annotation by controlling the amount of training data. As we shall see in Sec. 4 and 5, the different versions of the tagger at various levels of the hierarchy and accuracy will help us to understand the relation between the *Edit mode* annotation, and the complexity of the tagset and the accuracy of the tagger used for initial annotation. The taggers were trained on 1457 sentences (approximately 20,000 words) for Bangla and 2366 sentences (approximately 45,000 words) for Hindi. The taggers were tested on 256 sentences (~ 3,500 words) for Bangla and 591 sentences for Hindi, which are disjoint from the training corpus. The evaluation of a hierarchical tagset is non-trivial because the error in the machine tagged data with respect to the gold standard should take into account the level of the hierarchy where the mismatch between the two takes place. Clearly, mismatch at the category or type level should incur a higher penalty than one at the level of the attributes. If for a word, there is a mismatch between the type assigned by the machine and that present in the gold standard, then it is assumed to be a full error (equivalent to 1 unit). On the other hand, if the type assigned is correct, then the error is 0.5 times the fraction of attributes that do not agree with the gold standard.

Table 1 reports the accuracies of the various taggers. Note that the attributes in IL-POST correspond to morphological features. Unlike Bangla, we do not have access to a morphological analyzer for Hindi to predict the attributes during the POS tagging at the CTA level. Therefore, the tagging accuracy in the CTA level for Hindi is lower than that of Bangla even though the amount of training data used in Hindi is much higher than that in Bangla.

4 Experiments

The objective of the current work is to study the *cognitive load* associated with the task of linguistic annotation, more specifically, POS annotation. Cognitive load relates to the higher level of processing required by the working memory of an annotator when more learning is to be done in a shorter time. Hence, a higher cognitive load implies more time required for annotation and higher error rates. The time required for annotation can be readily measured by keeping track of the time taken by the annotators while tagging a sentence. The timer facility provided with the annotation tool helps us keep track of the annotation time. Measuring the error rate is slightly trickier as we do not have any ground truth (gold standard) against which we can measure the accuracy of the manual annotators. Therefore, we measure the IA, which should be high if the error rate is low. Details of the evaluation metrics are discussed in the next section.

Language	CT		CTA	
	High	*Low*	*High*	*Low*
Bangla	81.43	66.73	76.98	64.52
Hindi	87.66	67.85	69.53	57.90

Table 1: Tagging accuracy in % for Bangla and Hindi

The cognitive load of the annotation task is dependent on the complexity of the tagset, (un)availability of an appropriate annotation tool and bootstrapping facility. Therefore, in order to quantify the effect of these factors on the annotation task, annotation experiments are conducted under eight different settings. Four experiments are done for annotation at the Category+Type (CT) level. These are:

- **CT-AT**: without using annotation tool, i.e., using any standard text editor[2].
- **CT+AT**: with the help of the basic annotation tool.
- **CT+ATL**: with the help of the annotation tool in the edit mode, where the POS tagger used has low accuracy.
- **CT+ATH**: in the edit mode where the POS tagger used has a high accuracy.

Similarly, four experiments are conducted at the Category+Type+Attribute (CTA) level, which are named following the same convention: CTA-AT, CTA+AT, CTA+ATL, CTA+ATH.

4.1 Subjects

The reliability of annotation is dependent on the expertise of the annotators. In order to analyze the effect of annotator expertise, we chose subjects with various levels of expertise and provided them different amount of training and supervision during the annotation experiments.

The experiments for Bangla have been conducted with 4 users (henceforth referred to as B1, B2, B3 and B4), all of whom are trained linguists having at least a post-graduate degree in linguistics. Two of them, namely B1 and B2, were provided rigorous training in-house before the annotation task. During the training phase the tagset and the annotation guidelines were explained to them in detail. This was followed by 3-4 rounds of trial annotation tasks, during which the annotators were asked to annotate a set of 10-15 sentences and they were given feedback regarding the correctness of their annotations as judged by other human experts. For B1 and B2, the experiments were conducted in-house and under close supervision of the designers of the tagset and the tool, as well as a senior research linguist.

The other two annotators, B3 and B4, were provided with the data, the required annotation tools and the experimental setup, annotation guidelines and the tool usage guidelines, and the task were described in another document. Thus, the annotators were self-trained as far as the tool usage and the annotation scheme were concerned. They were asked to return the annotated data (and the time logs that are automatically generated during the annotation) at the end of all the experiments. This situation is similar to that of linguistic annotation using the Internet users, where the annotators are self-trained and work under no supervision. However, unlike ordinary Internet users, our subjects are trained linguists.

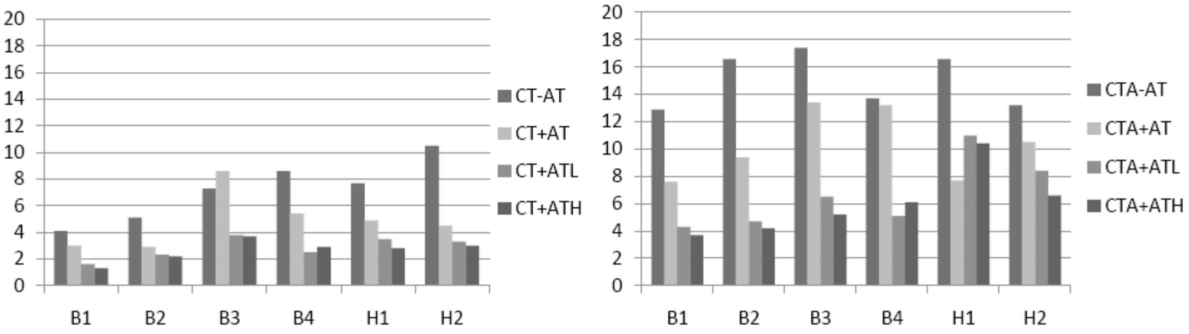
Experiments in Hindi were conducted with two users (henceforth referred to as H1 and H2), both of whom are trained linguists. As in the case of B1 and B2, the experiments were conducted under close supervision of a senior linguist, but H1 and H2 were self-trained in the use of the tool.

The tasks were randomized to minimize the effect of familiarity with the task as well as the tool.

4.2 Data

The annotators were asked to annotate approximately 2000 words for CT+AT and CTA+AT experiments and around 1000 words for CT-AT and CTA-AT experiments. The *edit mode* experiments (CT+ATL, CT+ATH, CTA+ATL and CTA+ATH) have been conducted on approximately 1000 words. The amount of data was decided based primarily on the time constraints for the experiments. For all the experiments in a particular language, 25-35% of the data was common between every pair of annotators. These common sets have been used to measure the IA. However, there was no single complete set common to all the annotators. In order to measure the influence of the pre-assigned labels on the judgment of the annotators, some amount of data was kept common between CTA+AT and CTA+ATL/H experiments for every annotator.

[2] The experiments without the tool were also conducted using the basic interface, where the annotator has to type in the tag strings; the function of the tool here is limited to loading the corpus and the timer.



(a) (b)

Figure 4: Mean annotation time (in sec per word) for different users at (a) CT and (b) CTA levels

Level	Mean Time (in Sec)			
	-AT	+AT	+ATL	+ATH
CT	6.3	5.0 (20.7)	2.6 (59.4)	2.5 (59.8)
CTA	15.2	10.9 (28.1)	5.2 (66.0)	4.8 (68.3)

Table 2: Mean annotation time for Bangla experiments (%reduction in time with respect to –AT is given within parentheses).

Level	IA (in %)			
	-AT	+AT	+ATL	+ATH
CT	68.9	79.2 (15.0)	77.2 (12.2)	89.9 (30.6)
CTA	51.4	72.5 (41.0)	79.3 (54.2)	83.4 (62.1)

Table 3: Average IA for Bangla experiments (%increase in IA with respect to –AT is given within parentheses).

5 Analysis of Results

In this section we report the observations from our annotation experiments and analyze those to identify trends and their underlying reasons.

5.1 Mean Annotation Time

We measure the mean annotation time by computing the average time required to annotate a word for a sentence and then average it over all sentences for a given experiment by a specific annotator. Fig. 4 shows the mean annotation time (in seconds per word) for the different experiments by the different annotators. It is evident that complex annotation task (i.e., CTA level) takes much more time compared to a simple one (i.e., CT level). We also note that the tool effectively reduces the annotation time for most of the subjects. There is some variation in time (for example, B3) where the subject took longer to get accustomed to the annotation tool. As expected, the annotation process is accelerated by bootstrapping. In fact, the higher the accuracy of the automatic tagger, the faster is the annotation. Table 2 presents the mean time averaged over the six subjects for the 8 experiments in Bangla along with the %reduction in the time with respect to the case when no tool is present (i.e., "-AT"). We observe that (a) the tool is more effective for complex annotation, (b) on average, annotation at the CTA level take twice the time of their CT level counterparts, and (c) bootstrapping can significantly accelerate the annotation process. We also note that experts working under close supervision (B1 and B2) are in general faster than self-trained annotators (B3 and B4).

5.2 Inter-annotator Agreement

Inter-annotator agreement (IA) is a very good indicator of the reliability of an annotated data. A high IA denotes that at least two annotators agree on the annotation and therefore, the probability that the annotation is erroneous is very small. There are various ways to quantify the IA ranging from a very simple percentage agreement to more complex measures such as the *kappa statistics* (Cohen, 1960; Geertzen and Bunt, 2006). For a hierarchical tagset the measurement of IA is non-trivial because the extent of disagreement should take into account the level of the hierarchy where the mismatch between two annotators takes place. Here we use percentage agreement which takes into consideration the level of hierarchy where the disagreement between the two annotators takes place. For example, the difference in IA at the category level between say, a Noun and a Nominal Modifier, versus the difference at the number attribute level between singular and plural. The extent of agreement for each of the tags is computed in the same way as we have evaluated our POS tagger (Sec.3.2.1). We have also measured the Cohen's Kappa (Cohen, 1960) for the **CT** level experiments. Its behavior is similar to that of percentage agreement.

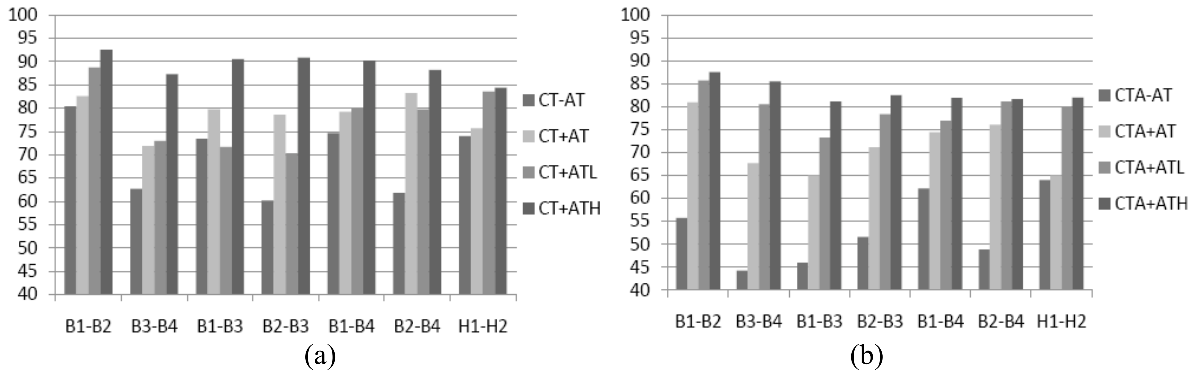


Figure 5: Pair-wise IA (in %) at (a) CT and (b) CTA levels

Fig. 5 shows the pair-wise percentage IA for the eight experiments and Table 3 summarizes the %increase in IA due to the use of tool/bootstrapping with respect to the "-AT" experiments at CT and CTA levels. We observe the following basic trends: (a) IA is consistently lower for a complex annotation (CTA) task than a simpler one (CT), (b) use of annotation tool helps in improvement of the IA, more so for the CTA level experiments, (c) bootstrapping helps in further improvement in IA, especially when the POS tagger used has high accuracy, and (d) IA between the trained subjects (B1 and B2) is always higher than the other pairs.

IA is dependent on several factors such as the ambiguity in the tagset, inherently ambiguous cases, underspecified or ambiguously specified annotation guidelines, and errors due to carelessness of the annotator. However, manual inspection reveals that the factor which results in very low IA in "-AT" case that the tool helps improve significantly is the typographical errors made by the annotators while using a standard text editor for annotation (e.g., NC mistyped as MC). This is more prominent in the CTA level experiments, where typing the string of attributes in the wrong order or missing out on some attributes, which are very common when annotation tool is not used, lead to a very low IA. Thus, memorization has a huge overload during the annotation process, especially for complex annotation schemes, which the annotation tool can effectively handle. In fact, more than 50% errors in CTA level are due to the above phenomenon. The analysis of other factors that lower the IA is discussed in Sec. 5.4.

We would like to emphasize the fact that although the absolute time difference between the trained and un-trained users reduces when the tool and/or bootstrapping is used, the IA does not decrease significantly in case of the untrained users for the complex annotation task.

Level	Tagger	Subjects			
		B1	B2	B3	B4
CT	*Low*	89.6	89.8	74.2	81.8
	High	90.8	90.1	64.8	77.8
CTA	*Low*	85.4	85.1	68.2	76.1
	High	86.4	85.4	59.1	73.4

Table 4: Percentage agreement between the edit and the normal mode annotations (for Bangla).

5.3 Machine Influence

We have seen that the IA increases in the edit mode experiments. This apparent positive result might be an unacceptable artifact of machine influence, which is to say that the annotators, whenever in confusion, might blindly agree with the pre-assigned labels. In order to understand the influence of the pre-assigned labels on the annotators, we calculate the percentage agreement for a subject between the data annotated from scratch using the tool (+AT) and that in the edit mode (+ATL and +ATH). The results are summarized in Table 4.

The low agreement between the data annotated under the two modes for the untrained annotators (B3 and B4) shows that there is a strong influence of pre-assigned labels for these users. Untrained annotators have lower agreement while using a high accuracy initial POS tagger compared to the case when a low accuracy POS tagger is used. This is because the high accuracy tagger assigns an erroneous label mainly for the highly ambiguous cases where a larger context is required to disambiguate. These cases are also difficult for human annotators to verify and untrained annotators tend to miss these cases during edit mode experiments. The trained annotators show a consistent performance. Nevertheless, there is still some influence of the pre-assigned labels.

5.4 Error Patterns

In order to understand the reasons of disagreement between the annotators, we analyze the confusion matrix for different pairs of users for the various experimental scenarios. We observe that the causes of disagreement are primarily of three kinds: (1) unspecified and/or ambiguous guidelines, (2) ignorance about the guidelines, and (3) inherent ambiguities present in the sentences. We have found that a large number of the errors are due to type (1). For example, in attribute level annotation, for every attribute two special values are '0' (denotes '*not applicable for the particular lexical item*') and 'x' (denotes '*undecided or doubtful to the annotator*'). However, we find that both trained and untrained annotators have their own distinct patterns of assigning '0' or 'x'. Later we made this point clearer with examples and enumerated possible cases of '0' and 'x' tags. This was very helpful in improving the IA.

A major portion of the errors made by the untrained users are due to type (2). For example, it was clearly mentioned in the annotation guidelines that if a borrowed/foreign word is written in the native script, then it has to be tagged according to its normal morpho-syntactic function in the sentence. However, if a word is typed in foreign script, then it has to be tagged as a foreign word. However, none of the untrained annotators adhered to these rules strictly.

Finally, there are instances which are inherently ambiguous. For example, in noun-noun compounds, a common confusion is whether the first noun is to be tagged as a nouns or an adjective. These kinds of confusions are evenly distributed over all the users and at every level of annotation.

One important fact that we arrive at through the analysis of the confusion matrices is that the trained annotators working under close supervision have few and consistent error patterns over all the experiments, whereas the untrained annotators exhibit no consistent and clearly definable error patterns. This is not surprising because the training helps the annotators to understand the task and the annotation scheme clearly; on the other hand, constant supervision helps clarifying doubts arising during annotation.

6 Conclusion

In this paper we reported our observations for POS annotation experiments for Bangla and Hindi using the IL-POST annotation scheme under various scenarios. Experiments in Tamil and Sanskrit are planned in the future.

We argue that the observations from the various experiments make a case for the need of training and supervision for the annotators as well as the use of appropriate annotation interfaces and techniques such as bootstrapping. The results are indicative in nature and need to be validated with larger number of annotators. We summarize our salient contributions/conclusions:

- The generic tool described here for complex and hierarchical word level annotation is effective in accelerating the annotation task as well as improving the IA. Thus, the tool helps reducing the cognitive load associated with annotation.
- Bootstrapping, whereby POS tags are pre-assigned by an automatic tagger and human annotators are required to edit the incorrect labels, further accelerates the task, at the risk of slight influence of the pre-assigned labels.
- Although with the help of the tool and techniques such as bootstrapping we are able to bring down the time required by untrained annotators to the level of their trained counterparts, the IA, and hence the reliability of the annotated data for the former is always poorer. Hence, training and supervision is very important for reliable linguistic annotation.

We would like to emphasize the last point because recently it is being argued that Internet and other game based techniques can be effectively used for gathering annotated data for NLP. While this may be suitable for certain types of annotations, such as word sense, lexical similarity or affect (see Snow et al. (2008) for details), we argue that many mainstream linguistic annotation tasks such as POS, chunk, semantic roles and Treebank annotations call for expertise, training and close supervision. We believe that there is no easy way out to this kind of complex linguistic annotations, though smartly designed annotation interfaces and methods such as bootstrapping and active learning can significantly improve the productivity and reliability, and therefore, should be explored and exploited in future.

Acknowledgements

We would like to thank the annotators Dripta Piplai, Anumitra Ghosh Dastidar, Narayan Choudhary and Maansi Sharma. We would also like to thank Prof. Girish Nath Jha for his help in conducting the experiments.

References

S. Baskaran, K. Bali, M. Choudhury, T. Bhattacharya, P. Bhattacharyya, G. N. Jha, S. Rajendran, K. Saravanan, L. Sobha and K.V. Subbarao. 2008. A Common Parts-of-Speech Tagset Framework for Indian Languages. In *Proc. of LREC 2008*.

J. Cohen. 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement, **20** (1):37-46

J. Geertzen, and H. Bunt. 2006. Measuring annotator agreement in a complex hierarchical dialogue act annotation scheme. In *Proc. of the Workshop on Discourse and Dialogue*, ACL 2006, pp. 126-133.

G. Eryigit. 2007. ITU Treebank annotation tool. In *Proc. of Linguistic Annotation Workshop*, ACL 2007, pp. 117-120.

I. Koutsis, G. Markopoulos, and G. Mikros. 2007. Episimiotis: A Multilingual Tool for Hierarchical Annotation of Texts. In *Corpus Linguistics*, 2007.

D. Reidsma, N. Jovanovi, and D. Hofs. 2004. Designing annotation tools based on the properties of annotation problems. *Report, Centre for Telematics and Information Technology*, 2004.

E. Ringger, P. McClanahan, R. Haertel, G. Busby, M. Carmen, J. Carroll, K. Seppi, and D. Lonsdale. 2007. Active Learning for Part-of-Speech Tagging: Accelerating Corpus Annotation. In *Proc. of Linguistic Annotation Workshop*, ACL 2007, pp. 101-108.

R. Snow, B. O'Connor, D. Jurafsky and A. Y. Ng. 2008. Cheap and Fast — But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proc of EMNLP-08*

K. Tomanek, J. Wermter, and U. Hahn. 2007. Efficient annotation with the Jena ANotation Environment (JANE). In *Proc. of Linguistic Annotation Workshop*, ACL 2007, pp. 9-16.

L. von Ahn and L. Dabbish. 2004. Labeling Images with a Computer Game. In *ACM Conference on Human Factors in Computing Systems, CHI 2004*.

Y. Wu, P. Jin, T. Guo and S. Yu. 2007. Building Chinese sense annotated corpus with the help of software tools. In *Proc. of Linguistic Annotation Workshop*, ACL 2007, pp. 125-131.