PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 102 OCTOBER 2014 57–68

A Set of Annotation Interfaces for Alignment of Parallel Corpora

Anil Kumar Singh

IIT (BHU), Varanasi, India

Abstract

Annotation interfaces for parallel corpora which fit in well with other tools can be very useful. We describe a set of annotation interfaces which fulfill this criterion. This set includes a sentence alignment interface, two different word or word group alignment interfaces and an initial version of a parallel syntactic annotation alignment interface. These tools can be used for manual alignment, or they can be used to correct automatic alignments. Manual alignment can be performed in combination with certain kinds of linguistic annotation. Most of these interfaces use a representation called the Shakti Standard Format that has been found to be very robust and has been used for large and successful projects. It ties together the different interfaces, so that the data created by them is portable across all tools which support this representation. The existence of a query language for data stored in this representation makes it possible to build tools that allow easy search and modification of annotated parallel data.

1. Introduction

Machine translation, at least for certain language pairs, has reached a point where it is now being practically used by professional translators as well as users. Many, perhaps a majority of these machine translation systems are based on the statistical approach (Koehn et al., 2007). The general architecture of these machine translation systems is easily portable to other language pairs than those for which they were made. However, the one major drawback of these systems is that they need a large quantity of aligned parallel text. While the rule-based approach (Corbí-Bellot et al., 2005) may provide a feasible solution in many cases where such corpora are lacking, the other alternative, i.e., creating the needed parallel corpora for language pairs that lack them can still be an effective solution under certain conditions such as the availability of sufficient quantity of parallel text in electronic form. Still, even when such

 Corresponding author: nlprnd@gmail.com
Cite as: Anil Kumar Singh. A Set of Annotation Interfaces for Alignment of Parallel Corpora. The Prague Bulletin of Mathematical Linguistics No. 102, 2014, pp. 57–68. doi: 10.2478/pralin-2014-0014.

text is available, there is a lot of work that needs to be done if the text is not already sentence aligned. Therefore, sentence alignment tools are one of the first enablers for creating machine translation systems based on the statistical approach in cases where sentence alignment accuracy is not close to 100%. If the text can be further aligned at the word level, it becomes a valuable resource for many other purposes.

Tools for automatic alignment of text, both at the sentence level (Brown et al., 1991; Gale and Church, 1991) and at the word level (Gale and Church, 1993; Och and Ney, 2003) are available. However, word alignment tools do not have an accuracy that will allow them to be used directly without further correction, except as part of a statistical machine translation system. Sentence alignment tools reportedly have very good accuracies, but on closer inspection we find that they do break down under certain conditions which are especially likely to occur for language pairs which lack sentence aligned parallel corpora. The languages should not be very different in phylogenetic terms and, more importantly, the text should not be 'noisy' (Singh and Husain, 2005), which practically means that it should be almost aligned already.

Beyond the word level, we could also have corpora which are partially aligned for specific grammatical elements such as nominal compounds, or even corpora which are syntactically aligned, e.g. parallel treebanks (Li et al., 2012). Such corpora are even more valuable for not only linguistic analysis and extraction of linguistic patterns but also for machine learning of Natural Language Processing (NLP) tasks.

2. The Workflow

The above are the reasons why we need annotation interfaces for parallel corpora. We need a set of interfaces that enable the complete workflow from sentence alignment to word alignment to syntactic (or even semantic) alignment. There are not many such interfaces available, at least in the open source domain. One that is available is called Uplug[1] (Tiedemann, 2003). It allows sentence alignment, word alignment and with some extensions such as in UplugConnector[2], even treebank alignment.

In the following paragraphs we describe a set of interfaces that aim to implement the complete process of parallel corpora alignment up to the syntactic (treebank) level. All these interfaces are part of the same suite of tools and APIs called Sanchay[3].

The workflow starts with some parallel text that is not sentence aligned, but is tokenized into sentences and words. An automatic sentence alignment tool can optionally be run on this parallel text. The text is then fed into the sentence alignment interface, where a user manually corrects the alignments marked by the sentence alignment tool. Alternatively, the user can directly mark the alignments completely manually. The output of the sentence alignment interface can then be run through an

[1] http://sourceforge.net/projects/uplug/

[2] http://www2.lingfil.uu.se/personal/bengt/uconn113.html

[3] http://sanchay.co.in

automatic word alignment tool, and then to the word alignment interface. A user then corrects the word alignments errors. The user can also start from scratch at the word alignment level and mark all the alignments manually. It is possible at this stage to create a shallow parsed aligned parallel corpus. Such a corpus will have chunks or word groups marked, (optionally) tagged and aligned.

The output of the word alignment interface can be run through an automatic syntactic analysis tool such as a parser. If the goal is ambitious, it may be possible to create parallel treebanks using the syntactic annotation interface available in the same suite. This interface is meant for creating syntactico-semantic resources such as treebanks.

Finally, there is a parallel syntactic annotation tool that will allow users to align the parallel treebanks at the deep syntactic level. Aligned parallel treebanks can allow a wealth of information to be extracted from them.

3. The Representation

For all the interfaces mentioned here except one, the data is stored in memory as well in files using a representation called the Shakti Standard Format or SSF (Bharati et al., 2014). This representation is a robust way of storing data which is the result of linguistic analysis, particularly syntactico-semantic analysis.

An example sentence in SSF is show below:

```
Address Token       Category       Attribute-value pairs
  ------------------------------------------------
  1     ((          NP
  1.1   children    NNS  <fs af=child,n,m,p,3,0,,>
        ))
  2     ((          VG
  2.1   are         VBP  <fs af=be,v,m,p,3,0,,>
  2.2   watching    VBG  <fs af='watch,v,m,s,3,0,,' aspect=PROG>
        ))
  3     ((          NP
  3.1   some        DT   <fs af=some,det,m,s,3,0,,>
  3.2   programmes  NNS  <fs af=programme,n,m,p,3,0,,>
        ))
  4     ((          PP
  4.1   on          IN   <fs af=on,p,m,s,3,0,,>
  4.1.1 ((          NP
  4.1.2 television  NN   <fs af=television,n,m,s,3,0,,>
        ))
        ))
  ------------------------------------------------
             Shakti Standard Format
```

In this representation, sentences are the nodes of a tree at the document level and the constituents are the nodes at the sentence level. Each node has a possible lexical

item, a tag (such as a part-of-speech or POS tag or a phrase tag) and a feature structure associated with it. SSF allows not only features of nodes to be stored, but also features across nodes, which represent relationships across words, chunks, phrases, nodes of a dependency tree or even sentences and documents. We have used an attribute called 'alignedTo' for storing the alignments, whether of words, chunks, wordgroups, phrases or of sentences or documents. This attribute takes a string value and multiple alignments can be given by using semi-colon as the separator.

Figures 1 and 2 show the same sentence analyzed according to phrase structure grammar and dependency grammar, respectively. SSF can encode both of them in the same place. For example, the underlying tree can be a phrase structure tree and the dependency tree can be encoded over the phrase structure tree using an attribute like 'drel' (dependency relation) and 'name' (a unique identifier for the node). The value of the 'drel' attribute is in two parts, separated by a colon, e.g. 'k1:children'. The first part is the dependency relation and the second part is the unique name.

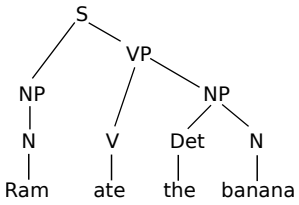

Figure 1. Phrase structure tree

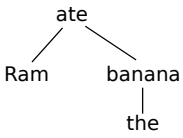

Figure 2. Dependency tree

We have opted for SSF because it allows us to preserve many different kinds of linguistic information about the text being aligned. The alignment information is added as an extra layer in the form of the 'alignedTo' attribute. Moreover, since we are storing alignments on both sides, the interfaces have to ensure that the values of the 'alignedTo' attribute are kept synchronized on the two sides. This makes it possible to get the alignments from either side to the other side. SSF allows us to keep all the information in one place and to have the data in a readable form so that even without a visualizer it can still be made sense of by a human user.

4. Sentence Alignment Interface

The sentence alignment interface (Figure 3) takes as input the text which has been tokenized into sentences. It shows one sentence per row in each table: one table on the source side and one on the target side. It includes an implementation of a sentence alignment algorithm which is based on sentence lengths. Such algorithms are based on the intuition that the lengths of translations are likely to be roughly proportional to lengths of sentences in the source language. The length can be calculated in terms

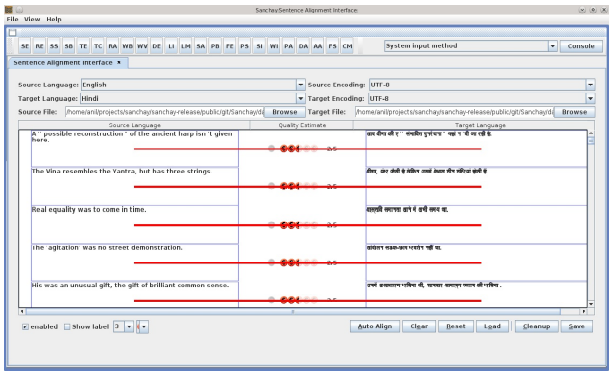

Figure 3. Sentence Alignment Interface: Initial alignments have been marked automatically

of words (Brown et al., 1991) or in terms of characters (Gale and Church, 1991). The latter has been shown to work better, perhaps because more data is available. In our implementation, we use both of them as features with weights. The alignment algorithm is a dynamic programming based algorithm. This implementation is available as the default, but with some minimal effort it is possible to plug-in any other available implementation of some sentence alignment algorithm, such as that in (Moore, 2002), which has been shown to be quite robust (Singh and Husain, 2007).

The user can mark manual alignments or correct automatic alignments by the simple mechanism of drag-and-drop. Deleting an alignment also uses this mechanism: If an alignment already exists, drag-and-drop deletes it. Multiple alignments to the same sentence are allowed on either sides. The interface tries to keep the sentences displayed according to the alignments, i.e., the display changes based on the alignments. This ensures that the user does not have to drag-and-drop too far because the potential alignment is as close to the source language sentence as possible.

The interface design is based on the assumption that sometimes the input to the interface can come from a machine translation system. In such a case, it is possible to extend the interface to allow the annotation of quality estimation measurement as well as to allow it to be used as a post-editing tool. Some work has already been done in this direction. Completing it is one of the future directions of this work.

The first step in using all the interfaces is to start Sanchay (see `http://sanchay.co.in`). Then the respective interfaces can be opened either by clicking on the buttons in the toolbox or from the 'File' menu. The toolbox buttons have two letter symbols for each interface. For example, the symbol for the word alignment interface is 'WI'. These could perhaps better be replaced by suitable icons. Hovering the cursor on the symbols displays their full name. Clicking on 'WI' will open the word alignment in-

terface. The symbol for the sentence alignment interface is 'SI', for the parallel corpus markup interface it is 'PA', for the syntactic annotation interface it is 'SA' and for the parallel syntactic annotation alignment interface it is 'PS'.

To start the alignment process, the user has to first browse to the source and the target files, where the target file should be the translation of the source file. Then clicking on the button 'Load' opens these files and displays them in the interface. The 'Auto Align' button is for running the automatic sentence aligner on the opened files, so that the work is reduced and only automatic alignment errors need to be corrected. Now the user can start marking the alignments by drag-and-drop.

The output is in the form of two files: source and target. Both files have sentences in the Shakti Standard Format. Each aligned sentence has an attribute 'alignedTo', whose value points to the unique id of the aligned sentence in the other file:

```
<Sentence id='1' alignedTo='1'>
```

5. Word Alignment Interface

Since the data representation (SSF) is common, the word alignment interface (Figure 4) can directly take in the output of the sentence alignment interface. This interface is actually more than a word alignment interface. It allows words to be grouped together on both sides, to be tagged, and then allows these groups to be aligned. Just like in the sentence alignment interface, a drag-and-drop over an existing alignment deletes that alignment.

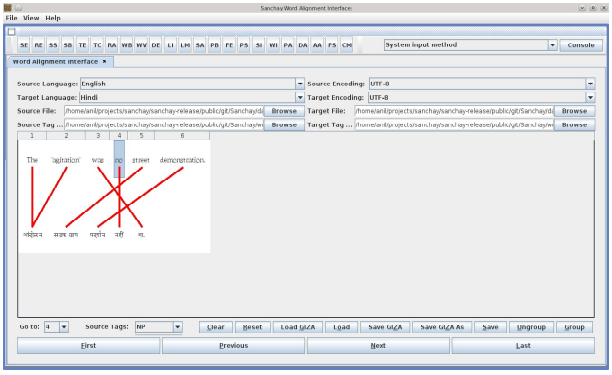

Figure 4. Word Alignment Interface

There is an implementation of a word alignment algorithm based on the first three IBM models (Brown et al., 1993), but that is not yet connected to the interface (another possible future direction for this work). However, the interface can read the output of

the most popular word alignment tool, namely GIZA++[4]. Therefore, it is possible to use this interface to correct the output of GIZA++. It can also save data in the same (GIZA++) format for those who are not comfortable with SSF or because the user wants to further process the corrected alignments in this format. This connectivity with GIZA++ makes the tool more useful.

While the sentence alignment interface uses the 'alignedTo' attribute at the sentence level, this interface uses the same attribute at the word or the chunk level to mark the alignments in the data representation.

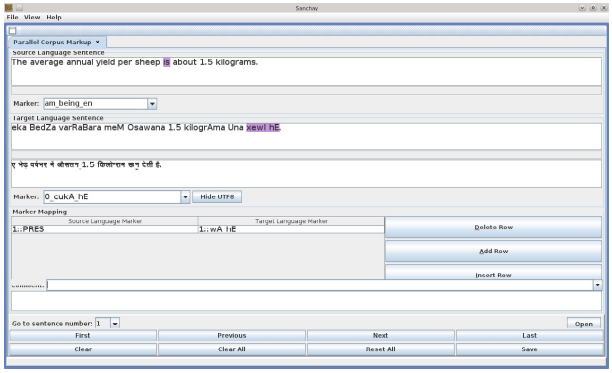

Figure 5. Parallel Markup Interface

The user can start the alignment process in two ways. One is by loading the source and target files (whether containing automatic alignments or not) and start working on them, either from scratch or by correcting errors. The other is to load the output of GIZA++ (in '*.A3.*' format) into the interface and correct the errors in automatic alignment. Since the interface allows grouping and tagging also, the source and target tag files may also need to be loaded.

The output from the interface is again in the form of two files, one source and one target, and both files have sentences in the Shakti Standard Format. Each aligned node has an attribute 'alignedTo', whose value points to the unique name of the aligned node in the corresponding sentence in the other file:

```
<Sentence id='1' alignedTo='1'>
...
8       ancient         <fs name='ancient' alignedTo='prAcIna'>
9       flute           <fs name='flute' alignedTo='vInA'>
...
```

[4] https://code.google.com/p/giza-pp

6. Parallel Markup Interface

Unlike the other interfaces in the suite, this one (Figure 5) does not use SSF. Instead, it uses a sentence level stand-off based representation. As a result, the alignments are stored in a file separate from the source and target data files. Otherwise, this interface has almost the same functionality as the the word alignment interface. It allows words to be grouped together. It allows them (or the groups) to be assigned some tags. And it allows their alignments to be marked. It does differ, however, in the way these alignments are marked. Instead of drag-and-drop, it requires the user to select the contiguous text to be grouped as a unit and then to select a tag from a drop-down list. A table below allows the alignments to be marked. The selected units appear in this table. To make the process easier, the cells on the target side only list (as a drop-down list) the possible alignments, out of which the user has to select one. This interface has been used for creating a parallel corpus of tense, aspect and modality (TAM) markers (how they are translated) and also for marking translations of nominal compounds.

A tool associated with this interface makes it possible to gather some statistics. This tool can be used to find out, for example, what are the possible translations of a word group and how many times do they occur in the corpus. The statistics are about the source side, the target side and about the alignments (or translations). Such statistics were used for extracting features for training a CRF based model for automatically finding the translations of TAM markers (Singh et al., 2007).

This interface runs by default in what we call the *task mode*. In this mode, the user does not browse to input files directly. Instead, the user can create task groups and task lists within each group. Out of these, one task can be selected by the user to work on when the interface is started. The task configuration files are stored in the 'workspace' directory of Sanchay. The file listing the task groups is stored in 'workspace/parallel-corpus-markup' directory and is named 'task-groups.txt'. Within this are listed the files which, in turn, list the specific tasks. The description of each task includes the source and the target text files which are to be aligned as well as the lists of tags on the two sides. These tags are used to tag the aligned units. The input text files are plaintext files with one sentence per line and they are assumed to be sentence aligned with one-to-one mapping. These files are not changed during the annotation process. The output is in the form of three extra files. Two of them (the '.marked' files) contain the sentence level offsets and the tag indices of the units (words or word groups) that are marked for alignment. The third (the '.mapping' file) contains the actual alignments.

7. Syntactic Annotation Interface

It is not possible to describe all the features of this interface here as it is the interface with the most functionality in Sanchay. We briefly describe it here. We plan to prepare detailed manuals for this and other interfaces and post them on the Sanchay website.

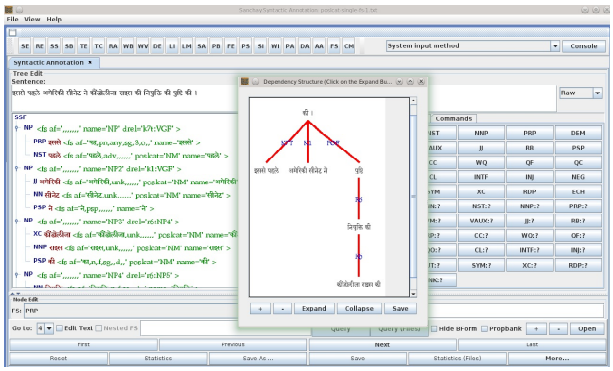

Figure 6. Syntactic Annotation Interface

This interface is meant for marking up the complete syntactico-semantic analysis of sentences. It can be used for annotating a sentence according to a phrase structure framework or a dependency structure framework. This includes information about the morphological analysis, POS tags, chunks, phrases, dependency relations and even certain kinds of semantic features. (There is also a separate but untested ProbBank annotation interface in the suite).

It is an easy to use interface that has evolved over the years based on the feedback of annotators and other kinds of users. The underlying representation is SSF. There is an API available for data in SSF and there are also various tools in the API to work on such data. There is a query language implementation in the API that allows easy search and modification of annotated data (Singh, 2012).

Figure 6 shows an example sentence in this interface, analyzed up to the dependency structure level. Dependency markup in this interface can be performed in a pop-up window by using the drag-and-drop mechanism, although there is a longer and more tedious method available too. The wealth of functionality in this interface makes it natural for us to use it for creating a parallel syntactic annotation interface.

8. Work In Progress: Parallel Syntactic Annotation Alignment Interface

This is an interface (Figure 7) that is still under construction. Only an initial prototype is ready. It uses the syntactic annotation interface on the source as well as the target side. The 'alignedTo' attribute here is used first at the document level and defines the scope of the alignments marked at the sentence and lower levels, and then at the setence and node levels. The input to this interface can be the output of either the sentence alignment interface or the word alignment interface.

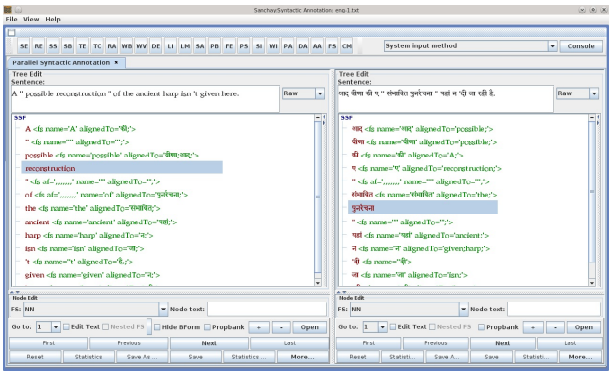

Figure 7. Parallel Syntactic Annotation Alignment Interface

The initial version of the interface simply has two panels, each containing the syntactic annotation interface, one for the source side and one for the target side. Alignments are marked again by simple drag-and-drop across the two panels. It can potentially be used to create parallel aligned treebanks such as the GALE Arabic-English Parallel Aligned Treebank[5].

The completion of this interface is the major future direction of this work. It will require resolving many issues that make it difficult to create such an interface. For example, how do we align dependency structures in this interface? Since the underlying representation is SSF, which has shallow parsed sentences at the core and has dependency relations marked by using attributes such as 'drel' (dependency relation), it is not directly possible to mark alignment of dependency nodes in the data representation (although it can be done in a visualizer). As the annotation of dependency relations is marked at the feature or attribute level, we essentially have to mark alignments also at the same level. This problem does not arise for phrase structure annotation, because such annotation is represented at the node level in SSF.

The above issue can be generalized to any information that is not directly represented at the node level. For example, alignments of parts of constituents even in the phrase structure framework poses the problem of not only representation, but also visualization in the interface. We do not yet have solutions for these issues. There may be other issues which we have not yet discovered and might find out once we try to complete the implementation of this interface.

The input to this interface is currently in the form of two files (source and target) in the Shakti Standard Format. The output is also stored in these same files. No additional files are created, but this may change as the interface develops.

[5] `http://catalog.ldc.upenn.edu/LDC2014T08`

9. Conclusion

Parallel aligned corpus can be a precious resource, especially if such corpus is annotated with linguistic analysis, as in a parallel aligned treebank. Even just sentence aligned parallel corpus is valuable enough to be the main resource needed for creating a statistical machine translation system. Therefore, user friendly annotation interfaces for creating parallel aligned corpora can be immensely useful. We described a set of annotation interfaces that ultimately aim to implement the entire process of the creation of parallel aligned treebanks, right from sentence alignment to word alignment to treebank alignment. This set of interfaces probably goes further than any other similar tool available in the open source domain. However, we find that, using our representation scheme (Shakti Standard Format or SSF), there are many issues which make it difficult to implement the entire treebank alignment process. One of them is that anything that is not represented directly at the node (document, sentence or word/chunk/constituent) level is difficult to mark up for alignment. Alignment of the dependency structure and non-node parts of phrase structure constituents are examples of this. The parallel markup interface, which uses the stand-off notation, does not have this problem, but it may be less user friendly.

Completing the implementation of the parallel syntactic annotation alignment interface is the main goal for the future. We also mentioned some other future directions such as completing the implementation of machine translation quality estimation measurement and post-editing functionality to the sentence alignment interface. To the word alignment interface, we can add the facility to tag alignment links as in the GALE Arabic-English Parallel Aligned Treebank. It can also be explored whether there is a better alternative to drag-and-drop.

Bibliography

Bharati, Akshar, Rajeev Sangal, Dipti Sharma, and Anil Kumar Singh. SSF: A Common Representation Scheme for Language Analysis for Language Technology Infrastructure Development. In *Proceedings of the COLING Workshop on Open Infrastructures and Analysis Frameworks for HLT (To Appear)*, 2014.

Brown, Peter F., Jennifer C. Lai, and Robert L. Mercer. Aligning Sentences in Parallel Corpora. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 169–176, 1991.

Brown, Peter F., Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19:263–311, 1993.

Corbí-Bellot, Antonio M., Mikel L. Forcada, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez, Iñaki Alegria, Aingeru Mayor, and Kepa Sarasola. An Open-source Shallow-transfer Machine Translation Engine for the Romance Languages of Spain. In *Proceedings of the Tenth Conference of the European Association for Machine Translation*, pages 79–86, May 2005.

Gale, William A. and Kenneth Ward Church. A Propgram for Aligning Sentences in Bilingual Corpora. In *Proceedings of the 29th Annual Meeting of the Association of Computational Linguistics (ACL)*, 1991.

Gale, William A. and Kenneth W. Church. A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, 19(1):75–102, Mar. 1993. ISSN 0891-2017.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2007.

Li, Xuansong, Stephanie Strassel, Stephen Grimes, Safa Ismael, Mohamed Maamouri, Ann Bies, and Nianwen Xue. Parallel Aligned Treebanks at LDC: New Challenges Interfacing Existing Infrastructures. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pages 1848–1855, 2012.

Moore, Robert C. Fast and Accurate Sentence Alignment of Bilingual Corpora. In *Proceedings of the 5th conference of the Association for Machine Translation in the Americas (AMTA)*, pages 135–144, 2002.

Och, Franz Josef and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

Singh, Anil Kumar. A Concise Query Language with Search and Transform Operations for Corpora with Multiple Levels of Annotation. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, Instanbul, Turkey, 2012. ELRA.

Singh, Anil Kumar and Samar Husain. Comparison, Selection and Use of Sentence Alignment Algorithms for New Language Pairs. In *Proceedings of the ACL 2005 Workshop on Parallel Text*, Ann Arbor, Michigan, 2005. Association for Computational Linguistics.

Singh, Anil Kumar and Samar Husain. Exploring Translation Similarities for Building a Better Sentence Aligner. In *Proceedings of the 3rd Indian International Conference on Artificial Intelligence*, Pune, India, 2007.

Singh, Anil Kumar, Samar Husain, Harshit Surana, Jagadeesh Gorla, Chinnappa Guggilla, and Dipti Misra Sharma. Disambiguating Tense, Aspect and Modality Markers for Correcting Machine Translation Errors. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, 2007.

Tiedemann, Jörg. *Recycling Translations – Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. PhD thesis, Uppsala University, Uppsala, Sweden, 2003. Anna Sågvall Hein, Åke Viberg (eds): Studia Linguistica Upsaliensia.

Address for correspondence:
Anil Kumar Singh
nlprnd@gmail.com
Dept. of CSE, IIT (BHU)
Varanasi-221005, U.P., India