Kunji : A Resource Management System for Higher Productivity in Computer Aided Translation Tools

Priyank Gupta, Rashid Ahmad, Manish Shrivastava, Dipti Misra Sharma
LTRC, KCIS
IIIT-Hyderabad
{priyank.gupta, rashid.ahmadpg08}@research.iiit.ac.in
{m.shrivastava,dipti}@iiit.ac.in

Abstract

Complex NLP applications, such as machine translation systems, utilize various kinds of resources namely lexical, multiword, domain dictionaries, maps and rules etc. Similarly, translators working on Computer Aided Translation workbenches, also require help from various kinds of resources - glossaries, terminologies, concordances and translation memory in the workbenches in order to increase their productivity. Additionally, translators have to look away from the workbenches for linguistic resources like Named Entities, Multiwords, lexical and lexeme dictionaries in order to get help, as the available resources like concordances, terminologies and glossaries are often not enough. In this paper we present Kunji, a resource management system for translation workbenches and MT modules. This system can be easily integrated in translation workbenches and can also be used as a management tool for resources for MT systems. The described resource management system has been integrated in a translation workbench Transzaar. We also study the impact of providing this resource management system along with linguistic resources on the productivity of translators for English-Hindi language pair. When the linguistic resources like lexeme, NER and MWE dictionaries were made available to translators in addition to their regular translation memories, concordances and terminologies, their productivity increased by 15.61%.

1 Introduction

NLP applications - machine translation systems and translation workbenches (which can have multiple MT systems integrated), are complex in nature as they are built with various complex heterogeneous NLP modules. These complex NLP applications are compute and knowledge intensive in nature and require various types of resources at different levels of processing.

The translators using the translation workbenches have to look for various resources - glossaries, terminologies, concordances and translation memory in order to supplement their translation tasks. It is still hard to maintain consistency while maintaining high productivity. Sometimes, the aid the translators get from the resources is not enough and they have to look for linguistic resources - Named Entities, lexical, bilingual, Multiwords and lexeme dictionaries - offline or online for the correct and appropriate meaning of a given phrase or word leading to additional cognitive load on translators.

In MT system, when MT modules execute on the given input, various types of resources namely lexical, bilingual, domain, multiword dictionaries, paradigms, maps files and rules are required at various stages of the processing. With the exploration of Sampark and Anusaaraka MT systems (ILMT, 2009; Chaudhury et al., 2010), it has been found that the resources can be used in a more efficient way in the both systems. These resources are developed by by NLP researchers who face resource management issues like exporting, importing, storage *etc*. For a large scale NLP application development, inefficient storage and management of these resources can become a bottleneck for productivity.

For the translation tasks and MT applications' development for Indian languages, the above mentioned issues become more critical as the digital content available in Indian languages are way less compared to other languages. As Hindi is one of the top 5 spoken languages of the world but still its digital content available on the web is less than 0.05% while the content of the other Indian languages are still lower. Hence the need of a resource management system arises which can ad-

D M Sharma, P Bhattacharyya and R Sangal. Proc. of the 16th Intl. Conference on Natural Language Processing, pages 184–192
Hyderabad, India, December 2019. ©2019 NLP Association of India (NLPAI)

dress the issues of both translators as well as MT modules' development.

In this paper, we present a resource management system named Kunji, which addresses the issues related to resources of both translation workbenches and MT system development for Indian languages. It can be used in translation workbenches to facilitate the translators to use additional linguistic resources - domain, lexical, named entities, multiword and WSD dictionaries - along with the terminologies, concordances and glossaries. It facilitates the translators in two ways - i) reduction of their repetitive load by providing searching and filter mechanism and ii) reuse of their previously translated words, phrases and sentences by providing the provision of personal and domain dictionaries. It shifts the problem of recall to recognition which causes less cognitive load on translators by providing them provision to search the corresponding terms and recognize. It also facilitates the NLP researchers to manage the resources for MT modules in a robust manner while addressing their issues of management, export, import, formats and efficient search etc. It also lets them evaluate and verify their work from other senior expert. Such system facilitates the large scale MT applications development by making the resources available to multiple modules and their facilitation to process and reuse them efficiently in an MT system.

Additionally we performed an experiment to study the impact of providing this resource management system along with linguistic resources on the productivity of translators. The described resource management system, Kunji, has been integrated in the translation workbench Transzaar (Ahmad et al., 2018). The resources with linguistic features like lexeme, NER and MWE dictionaries have been provided in Kunji to facilitate translators in addition to their regular translation memories, concordances and terminologies for English-Hindi language pair. We observed that the productivity of the translators increased by 15.61% when the resources with linguistic features were provided to the translators in addition to their regular resources (translation memories, glossaries, concordances and terminologies).

In this paper, background and motivation for Kunji is described in Section-2 followed by related work in Section-3. The detailed architecture and functionality of Kunji is described in Section-4. Section-5 describes the experiment performed on Kunji followed by its results and discussion in Section-6. In Section-7, we conclude our work.

2 Background and Motivation

In current scenario, various translation workbenches(CAT tools) are available. Some of them offer the resources like translation memories(TM), concordances, terminologies and glossaries to translators but there is a lack of the resource management systems which provide the provision by which a translator can use additional linguistic resources like NE, MWE and lexeme dictionaries. Additionally there is a lack of the provisions of reuse, management and verification of these resources which can lead a process to improvement and development of the MT systems.

There are several tools for resource development but they are built with separate purposes and languages as they are for task specific. Like for annotators they have separate tools in Indian languages like Sanchay (Singh, 2008) but it is desktop based application. Some tools like GATE (Cunningham, 2002) addresses the some of the problems faced by NLP researchers while developing NLP resources but it is desktop based application.

It motivated us to develop Kunji, a resource management tool, which facilitates the translators in translator workbenches as well as MT system development process. It tries to address the issues faced by both translators as well as MT module developers. It allows the translators to use and manage the linguistic resources along with their terminologies and glossaries.

3 Related Work

There are some translation workbenches with separate resource management systems such as: Smartcat [1], Memsource [2] and Matecat (Federico et al., 2014) are web based CAT tools, provide central resources management for a project but they do not provide the provision for the linguistic resources. CASMACAT (Alabau et al., 2014) is a translation workbench which is web based and offers advanced functionality for computer-aided translation. It offers TMs but it also lacks the provision for the linguistic resources. PET (Aziz et al., 2012) is a tool to postedit to for evaluating

[1] https://www.smartcat.ai

[2] https://www.memsource.com

the quality of translations. It evaluates the efforts required in translations in order to be fixed. SDL Trados, MemoQ[3] and Anubis (Jaworski, 2013) are CAT tools which are not web based and both lack the provision to use linguistic resources.

Brat (Stenetorp et al., 2012) is basically a web based annotation tool which focuses mainly on text annotations to enhance annotators productivity by closely integrating NLP technology into the annotation process. It doesnt address the issues in resources management with respect to translators and large scale NLP application development. Creating language resources for NLP in Indian languages (Sangal and Sharma, 2001) defines a novel idea in which the development of lexical resources is linked with an example NLP application like MT then it can act as a test bed for the developing resources and provide constant feedback. Sanchay (Singh, 2008) provides an extra layer of easily customizable language encoding support for less computerized languages along with an editor named Sanchay with different types of fonts and language encoding supports but it is desktop based application. GATE (Cunningham, 2002) enables users to develop and deploy the language engineering tools and resources in a robust manner. It is a desktop based tool which supports many NLP and information extraction tasks in multiple languages. Nancy et al (Ide and Romary, 2009) presents an abstract data model and its implementation for linguistic annotations. Annomarket (Dimitrov et al., 2014) described a cloud-based open platform for text mining, which aims to assist the development and deployment of robust, large-scale text processing applications.

We see that in some translation workbenches, the resource management systems for TM, Terminologies and glossaries are available but they don't provide the provision for linguistic resources as well.

4 Kunji : Resource Management System

We propose Kunji-a Resources management System, which is a web based system based on microservices architecture. It can be easily integrated to a translation workbench and provides a mechanism to facilitate translators to use and manage the various types of resources (i.e. like terminologies, glossaries and domain dictionaries along with linguistic resources(lexeme dictionaries, NERs, MWEs etc.)). It does not only facilitate the translators but also facilitates the MT modules development by facilitating language researchers to manage the language engineering resources in a robust manner for various language processing tasks and evaluate them which boosts the process of large scale MT development.

4.1 Microservices Based Architecture and NLP Applications

Microservices (Thönes, 2015) based approach is an architectural concept of developing an application or software systems according to which a functionality or process can be developed, deployed and tested independently. An application is divided into set of such modular components or functionalities in which each functionality is an independent or disjoint from the others' in the application. In this approach, each functionality or component has a proper structured interface which is called API and is used to communicate with the corresponding module. It helps to overcome the drawbacks of monolithic approach.

The complex NLP applications- especially MT systems or translation workbenches are knowledge and compute intensive in nature. So management and development of the resources for them requires deep knowledge of corresponding domain, nature of language while implementing algorithm for it requires knowledge of both streams NLP and computer science. For the development of the resource management tools for a translation workbench, the requirements of translators need to be understood. Many existing NLP applications for resource management follow monolithic design and hence static in nature. In NLP, for building scalable, distributed resource management systems , microservices based architecture can overcome the barriers of the monolithic architecture.

4.2 Architectural requirements

Various NLP applications - CAT tools have been analyzed for their resource management, their procedures to utilize different kinds of resources. For the translation workbenches, various aspects like their reuse, verification and sharing of resources in a projects among various translators which are linked to a project. In MT modules, the procedures to access, management and reuse of the resources have been analyzed along with their functional and development requirements. We figured out the fol-

[3] https://www.memoq.com/en

lowing Architectural requirements for the system which are given below:

4.2.1 Microservices based architecture and Web based tool:

The need of microservices based architecture arise to overcome the barriers imposed by monolithic architecture which also facilitates the creation of the dynamic web based tool for the same. In such paradigm, every functionality of the resource management system will be exposed as an independent service which can be utilized by the resource management in translation workbenches in order to facilitate the translators. It can also facilitate the NLP resource management for large scale MT development.

Today is the era of the web, everything we see and work on is on web or is exposed as a web based service or tool. Due to mobile and internet evolution, many people are internet friendly even non-technical people. So in order to make people use and develop resources in more efficient way it is a major need to make it available in web as a form of web-tool which will also makes the application platform independent.

4.2.2 Facilitation in translation workbench:

The resources management system should be designed in such a way so that it should be able to integrate in the translation workbench in order to facilitate translators to increase their productivity. So that translators can manage and use their own private dictionaries, domain dictionaries, terminologies among with the project or task specific dictionaries.

4.2.3 Support for linguistic resources in translation workbenches:

It should support the translators to use, reuse and manage the linguistic resources like NE, MWE and lexeme dictionaries dictionaries as well.

4.3 Our Architecture

We explored the various aspects of NLP resources management and the ways in which they can be utilized in a translation workbenches to aid the translation tasks to translators as well as their facilitation to NLP researchers and annotators towards development of large Scale MT systems. Taking such requirements in consideration, we designed the architecture of the system such that it is a collection of independent microservices for each of the functionality which can be deployed independently. We resolve the problem of monolithic architecture by exposing each functionality as a microservice which can be executed and interacted via RESTful API.

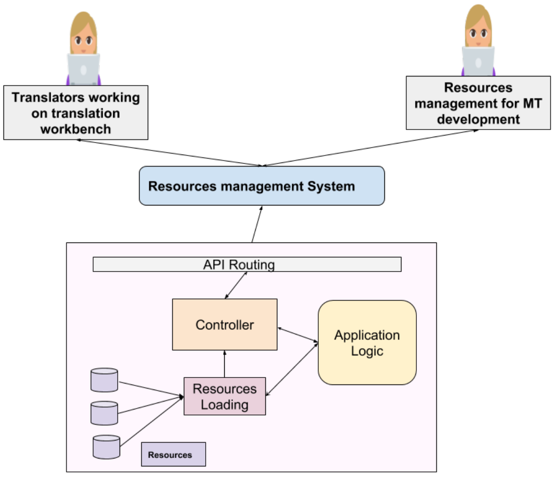
Translators working on translation workbench
Resources management for MT development
Resources management System
API Routing
Controller
Application Logic
Resources Loading
Resources

Figure 1: Kunji : Architecture

4.3.1 Kunji : Architecture Explanation

The architecture has been shown in Fig. 1 following with it's explanation.

- **Application logic :** It is the block or portion containing the complete functionality of the system. Hence the system's functionalities are structured, designed and implemented in the functional way independently.
- **In-Memory Resource Loading :** The different types of resources are required in different modules of MT as well as for providing help in the translation workbenches. Their loading from database at each execution process makes process slow as well as it imposes an unnecessary load to the system. Hence the services of the system are designed in such a way that resources load at the time of the starting of the service not at the time of invocation of the service.
- **REST APIs :** REST(Representational state transfer) API is basically an approach or technology for communications which is used in developing web services. It provides many methods such as HTTP or HTTPs for implementation. We expose each of the functionality of the resource management system as a

microservice with REST API interface to interact with. The APIs are created for create, add, import, export, delete, link, update and verify.

- **UI** The simple UI has been created with the HTML5 and JavaScript

4.3.2 Structure of Resources

First the configuration of a resource is defined followed by the corresponding entries of the data inside the resource as follows:

- Configuration of a resource : It contains the following fields as given below in the Table-1:

Resource Configuration Fields
resourceId
resourceName
resourceType
domain
subDomain
srcLang
tgtLang
project
client
description
createdBy
modifiedBy
creationDatetime
lastModifiedDateTime

Table 1: Resource configuration.

First the configuration of the resource is created as the configuration mentioned in Table-1:

- A data record of a resource will look like as mentioned in Table-2 :

  It contains the record with its target value and category and other required fields.

- **Explanation:** First the configuration of a particular resource type is created then its corresponding data entries can be created. So after the resources are created then we read the corresponding configuration of the resource and load it in memory in order to make process efficient.

Resource Record Fields
resourceId
source
target
category
gender
subCategory
createdAt
modifiedAt
createdBy
modifiedBy
description
verfiedFlag

Table 2: Resource Record Details.

4.3.3 Technologies used

For the controller part, for complete in-memory based architecture and for creating the microservices we used Java servlets with Apache-Tomcat-9 web server.

For the storage of the resources we used MongoDB-3.6.

For UI portion of the management system HTML5 along with CSS and Ajax have been used.

4.3.4 Features:

This resource management system has been integrated with two translation workbenches Transzaar (Ahmad et al., 2018). The features of the resource management system are given as follows:

- Addition of the resources in single and bulk mode.
- Editing and updating of the resources and their corresponding meta data.
- Provision to export and reuse the resources in various tools.
- Search: N-gram dynamic search mechanism.
- Easy import the resource in a translation workbench.
- Provision to develop and manage resources with linguistic features.
- Provision to verify them with a senior language expert of the corresponding language.

4.3.5 Process Flow

By using Kunji, translators can get aid in their translation tasks while using translation workbench. We demonstrated it with facilitating the translation workbench transzaar with Kunji. In kunji, they can create or import and use various kinds of resources like terminologies, bilingual and glossary dictionaries. Consider a scenario where a translator working on the workbench has some domain terminology dictionaries and wants to get help in his/her translation tasks. Then he can simply take that dictionary text file with source and target tab separated in each line in it. These resources can be imported/uploaded in the Kunji with the user metadata i.e. username, domain, language pair, and project etc.. So when translator opens his translation tasks and do editing then he can get the appropriate meanings of the terms or words of source text from terminology dictionary uploaded in Kunji. Those words are highlighted in source pane and on clicking them we can get appropriate meanings of the terms. Kunji facilitates a user working on workbench to add, import, export, update, n-gram search and edit the resources.

Possible types of the resources which can be uploaded are terminologies, glossaries, domain and bilingual dictionaries etc. Kunji also facilitates translator to import and use linguistic resources namely lexemes, named entities(NE), multiword dictionaries which can be helpful in selecting the correct form of the meaning of a word they should use.

4.4 Benefits of Kunji

It helps the translators by allowing them to search and look-up into their corresponding user specific or system specific resources, terminologies, glossaries which they can link with their translation tasks of a project to increase their productivity. Repetitive tasks are also facilitated by search or look up of a term which repeats many times in the task from the dictionaries and hence their consistency for given translation task would not be affected.Translators can import and use the linguistic resources like lexeme, NE and MWE dictionaries which can be helpful to decrease their cognitive load like selecting the correct form of the meaning of a word they should use. An experiment has been performed and is explained in Section-5 which would show how the linguistics resources can be beneficial for translators when we use this resource management system with a translation workbench.

Kunji also provides various aspects for managing and developing resources for MT application development. For NER and MWE, it provides provisions of category and subcategories like person/location/organization or Idiom/multi-word phrases/Domain terms etc. Similarly, it also provides the ways to create different types of domain dictionaries according to the need. So the development process of MT can be hugely facilitated by using this NLP researchers-linguists, annotators and and NLP module developers can be benefited. It addresses issues like encodings, fonts, usage of the common resources and lexical dictionaries for multiple tools, import and export etc so using such tool can enhance their productivity.

5 Experiment

The described resource management system - Kunji has been integrated in translation workbenches Transzaar (Ahmad et al., 2018). We study the impact of providing this resource management system in CAT tool-Transzaar along with linguistic resources on the productivity of translators for English-Hindi language pair. The various types of resources namely terminologies, glossaries, TM along with linguistic resources like lexeme, NE and MWE dictionaries were made available in the resources management tool in the workbench.

5.1 Experimental setup

We setup the experiment of translation tasks from the news data of "The Hindu" news website [4].

5.1.1 Data Set

For the experiment, 20 news stories from The Hindu news paper From national domain have been taken. The data stories have been divided into 4 sets with each set containing 5 stories. The details of the data sets are given in Table 3 and Table 4.

We integrated the Kunji, the resource management system to the Tanszaar and integrated the resources namely terminologies, glossaries, TM along with linguistic resources like lexeme, NE and MWE dictionaries were made available in the resources management tool in the workbench. We divided the experiment of translation tasks into 6 scenarios which are described in Table 5.

[4] https://www.thehindu.com

Set No	Paras	Words	Sentences	Avg words per para	Avg paras per story	Avg sentences per para	Avg sentences per story
Set1(5 stories)	42	1180	65	28	8.5	1.54	13
Set2(5 stories)	64	1722	87	26.9	12.8	1.36	17.4
Set3(5 stories)	59	1553	83	26	11.8	1.4	16.6
Set4(5 stories)	54	1238	74	23	10.8	1.37	14.8

Table 3: Data sets description-1

Total words in dataset	5693
Total sentences in dataset	309
Total paras in dataset	219
Average no. of sentences in each data set	15.45
Average sentence length	18.34 tokens
Average paras in complete data set	11
Average para length	26 tokens

Table 4: Data sets description-2

Scenarios	Description
Scenario-1	Manual Translation
Scenario-2	Post-editing in GT on Text editor
Scenario-3	GT with Transzaar without Kunji
Scenario-4	GT with Transzaar and Kunji (onlyTM, Terms, Glossaries)
Scenario-5	GT with Transzaar and Kunji (only linguistic resources)
Scenario-6	GT with Transzaar and Kunji (TM, Terms, Glossaries and linguistic resources)

Table 5: Description of Various scenarios : Here GT refers to Google translation output

We have chosen the English-Hindi language pair for our experiment and created the translation tasks from the dataset for each of the mentioned scenarios in Table 5 and assigned them to the four translators. The translation tasks have been assigned in such a way that no translator would repeat the same set again in a given scenario.

6 Results and Discussion

The results of experiments are given in Table 6 and Table 7. Each cell in the Table 6 presents the total time taken(in hours) by corresponding translator in the post-editing of the each set. The cells in the last row present the average time taken on all sets in the corresponding scenario. Table 7 presents per sentence time taken by each translator(in minutes) in the 6 different scenarios given in Table 5.

From Table 6 and Table 7 we see the impact of a translation workbench with and without the availability of different kinds of resources. We observed the translation data with professional translators in Table 6 and Table 7. Scenario-3 is when we use a workbench without any resources. Compared to Scenario-3, in Scenario-4 we use the workbench with translation resources but exclude linguistic resources. We observe that the productivity is significantly improved by 29.93%. When we contrast this with Scenario-5 where only linguistic resources are used then the productivity is only improved by 17.12% over Scenario-3. When we use all the resources including linguistic resources then the productivity is improved by 45.54% as compared to Scenario-3. Hence, we see that the productivity is improved additionally by 15.61% over Scenario-4 when the resources with linguistic features are made available for the translators in the workbench along with TM, Terms and Glossaries. From the above mentioned results, it is evident that the translators' productivity is positively affected when they get help from the utilization of resources with linguistic features in CAT tool.

The scenario-wise time consumed per sentence translation by each of the translators is presented in Table 7. It is observed that the productivity of each translator increases when we go from Scenario-1 to Scenario-6. For each translator, it is observed that their productivity increases as they use the translation workbench with resources. Their productivity significantly increase by more than 66% in scenario-6 i.e. when the linguistic resources are made available for the translators in the workbench along with TM, Terms and Glossaries. In scenario-6, the corresponding productivity of the translators are increased by 66%, 71.6%, 71.3% and 68.2% respectively. We also see the significant improvement in productivity when we go from scenario-4(workbench with translation resources but exclude linguistic resources) to scenario-6((workbench with translation resources but including linguistic resources).

There are some observations of the translators about the data set. The Set-1 is the toughest of all four sets. It is evident from table-7 that, translators took most time in translation of this set in each of the scenario. The Set-4 is the easiest of the sets as less time is consumed in its translation. Translators found the linguistic dictionaries (lex-

Set No	**Scenario-1 (Total Time in hours)**	**Scenario-2 (Total Time in hours)**	**Scenario-3 (Total Time in hours)**	**Scenario-4 (Total Time in hours)**	**Scenario-5 (Total Time in hours)**	**Scenario-6 (Total Time in hours)**
Set1	1.986(U1)	1.211(U2)	1.16(U3)	0.826(U2)	0.96(U4)	0.702(U21)
Set2	1.733(U4)	1.12(U1)	1.064(U2)	0.734(U1)	0.858(U3)	0.563(U4)
Set3	1.464(U3)	0.8431(U4)	0.835(U1)	0.5925(U4)	0.6864(U2)	0.421(U3)
Set4	1.158(U2)	0.702(U3)	0.64(U4)	0.443(U3)	0.565(U1)	0.33(U2)
Avg Time Taken in the scenario	1.585	0.97	0.9236	0.647	0.766	0.503

Table 6: Results of Experiment

Scenarios	Total Time taken by U1 per sent. (in min.)	Total Time taken by U2 per sent. (in min.)	Total Time taken by U3 per sent. (in min.)	Total Time taken by U4 per sent. (in min.)
Scenario-1	1.84(S1)	0.938(S4)	1.058(S3)	1.195(S2)
Scenario-2	0.78(S2)	1.03(S1)	0.569(S4)	0.61(S3)
Scenario-3	0.6(S3)	0.733(S2)	1.07(S1)	0.52(S4)
Scenario-4	0.506(S2)	0.76(S1)	0.359(S4)	0.43(S3)
Scenario-5	0.41(S4)	0.49(S3)	0.59(S2)	0.88(S1)
Scenario-6	0.64(S1)	0.267(S4)	0.304(S3)	0.38(S2)
Improvement (in %)	66%	71.6%	71.3%	68.2%

Table 7: Scenario-wise Results of Experiment with Professional Translators

eme, named entities and MWE) very helpful in the translation. For example the meaning of the word ”dreaded” contains two senses in different contexts in Hindi. The lexeme dictionary aided the translators in disambiguation of word senses.

7 Conclusion

We have proposed a resource management system which addresses the issues of the translators as well as MT system development. Using Kunji, a translator can utilize, reuse and manage his resources which can be shared across projects and can be used further. Also the NLP researchers get benefited using it in their resources management in MT system development so that the resources can be utilized and managed in an efficient way.

It also provides the mechanism which allows a translator to get help from the linguistic resources-lexeme, NER and MWE dictionaries in addition to their regular translation memories, concordances and terminologies. We performed experiments which show the effect of the linguistic resources in the productivity of the translators. We see that their productivity increased by 15.61% when they use linguistic resources along with their regular TM, terminologies and glossaries as mentioned in Table 6 and Table 7. We hope to extend this system by analyzing the impact of more complex NLP module like Morphological analyzer in the translation pipeline.

References

Rashid Ahmad, Priyank Gupta, Nagaraju Vuppala, Sanket Kumar Pathak, Ashutosh Kumar, Gagan Soni, Sravan Kumar, Manish Shrivastava, Avinash K Singh, Arbind K Gangwar, et al. 2018. Transzaar: Empowers human translators. pages 1–8.

Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis Leiva, et al. 2014. Casmacat: A computer-assisted translation workbench. pages 25–28.

Wilker Aziz, Sheila Castilho, and Lucia Specia. 2012. Pet: a tool for post-editing and assessing machine translation. pages 3982–3987.

Sriram Chaudhury, Ankitha Rao, and Dipti M Sharma. 2010. Anusaaraka: An expert system based machine translation system. pages 1–6.

Hamish Cunningham. 2002. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

Marin Dimitrov, Hamish Cunningham, Ian Roberts, Petar Kostov, Alex Simov, Philippe Rigaux, and Helen Lippell. 2014. Annomarket–multilingual text analytics at scale on the cloud. pages 315–319.

Marcello Federico, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico Lupinetti, Andrea Martines, et al. 2014. The matecat tool. pages 129–132.

Nancy Ide and Laurent Romary. 2009. Standards for language resources. *arXiv preprint arXiv:0909.2719*.

GROUP ILMT. 2009. Sampark: Machine translation system among indian languages. `http://sampark.iiit.ac.in`.

Rafał Jaworski. 2013. Anubis-speeding up computer-aided translation. pages 263–280.

Rajeev Sangal and Dipti Misra Sharma. 2001. Creating language resources for nlp in indian languages 1. background.

Anil Kumar Singh. 2008. A mechanism to provide language-encoding support and an nlp friendly editor.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. pages 102–107.

Johannes Thönes. 2015. Microservices. *IEEE software*, 32(1):116–116.