SEECAT: Speech & Eye- tracking Enabled Computer Assisted Translation

Mercedes Garcia-Martinez, Karan Singla, Aniruddha Tammewar, Bartolomé Mesa-Lao, Ankita Thakur, M.A. Anusuya, Srinivas Banglore, Michaël Carl

▶ **To cite this version:**

Mercedes Garcia-Martinez, Karan Singla, Aniruddha Tammewar, Bartolomé Mesa-Lao, Ankita Thakur, et al.. SEECAT: Speech & Eye- tracking Enabled Computer Assisted Translation. European Association for Machine Translation: EAMT, 2014, Dubrovnik, Croatia. pp.81–88. hal-01433258

HAL Id: hal-01433258
https://hal.science/hal-01433258

Submitted on 24 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SEECAT: ASR & Eye-Tracking Enabled Computer-Assisted Translation

Mercedes García-Martínez[1], **Karan Singla**[2], **Aniruddha Tammewar**[2], **Bartolomé Mesa-Lao**[1], **Ankita Thakur**[3], **Anusuya M.A.**[4], **Srinivas Bangalore**[5], **Michael Carl**[1]

[1]CRITT - Copenhagen Business School, Denmark, [2]IIIT Hyderabad, India, [3]JSSATE, Noida, India
[4]SJCE, Mysore,India, [5]AT&T Research Labs, United States

Abstract

Typing has traditionally been the only input method used by human translators working with computer-assisted translation (CAT) tools. However, speech is a natural communication channel for humans and, in principle, it should be faster and easier than typing from a keyboard. This contribution investigates the integration of automatic speech recognition (ASR) in a CAT workbench testing its real use by human translators while post-editing machine translation (MT) outputs. This paper also explores the use of MT combined with ASR in order to improve recognition accuracy in a workbench integrating eye-tracking functionalities to collect process-oriented information about translators' performance.

1 Introduction

Human-aided machine translation is gradually becoming a common practice for language service providers (LSPs) as opposed to machine-aided human translation. Depending on the nature of the text, more and more LSPs pre-translate the source text using existing translation memories (TMs) and then automatically translate the remaining text using an MT engine. Then human translators correct and adapt, i.e. post-edit, the output from both TMs and MT to produce different levels of translation quality. Improving and maximizing the potentials of a post-editing workbench is thus one of the priorities set by both the industry and the research community (Mesa-Lao, 2012). The motivation behind this paper comes from a desire to know how different input modalities in a computer-assisted translation (CAT) workbench can be of greater support to translation professionals.

Keyboards are the most widely used input device for text production and they seem to be the easiest input method when only minor changes are needed. However, in the context of post-editing, when the text requires major changes (e.g. editing larger segments of text), typing could be optimized using other input modalities. Moreover, if the post-editor is not a touch typist, then she has to switch visual attention back and forth between the screen and the keyboard making the task more complex. A possible solution for this profile of users could be the use of other input methods, such as ASR or hand-writing, in addition to traditional typing (Hauptmann and Rudnicky, 1990).

The comparison between ASR and typing as input methods can be done based on task duration, i.e. measuring the time needed to type against the ASR rate including possible corrections to fix recognition inaccuracies. Studies on input durations have shown that ASR input can be faster (Chen, 2006; Vidal et al., 2006).

This paper is structured along the following lines: The first section presents the CASMACAT[1]

© 2014 The authors. This article is licensed under a Creative Commons 3.0 licence, no derivative works, attribution, CC-BY-ND.

[1]CasMaCat: Cognitive Analysis and Statistical Methods for Advanced Computer Aided Translation. Project co-funded by the European Union under the Seventh Framework Programme Project 287576 (FP7 ICT-2011.4.2). URL: http://casmacat.eu. Demo: http://casmacat.prhlt.upv.es/mail-demo/askdemo.php

workbench as an introduction to the SEECAT[2] project. A description follows on how new ASR modules for English, Spanish and Hindi have been added to the SEECAT workbench. The last section presents the experimental data collected in two pilot studies with human translators performing a series of tasks using ASR and keyboard as input methods.

2 Background: The CASMACAT workbench

The CASMACAT (Alabau et al., 2014a) project aims at developing the next-generation translation workbench to improve productivity, quality, and work practices in the translation industry. The current CASMACAT prototype (version 2) allows users to upload documents and work with a first MT draft for post-editing. In its current implementation the workbench only supports keyboard and mouse as input modes, but it will also support e-pen(Alabau et al., 2014b) in the next prototype. A diagram of the major components of the CASMACAT workbench is shown in Figure 1.

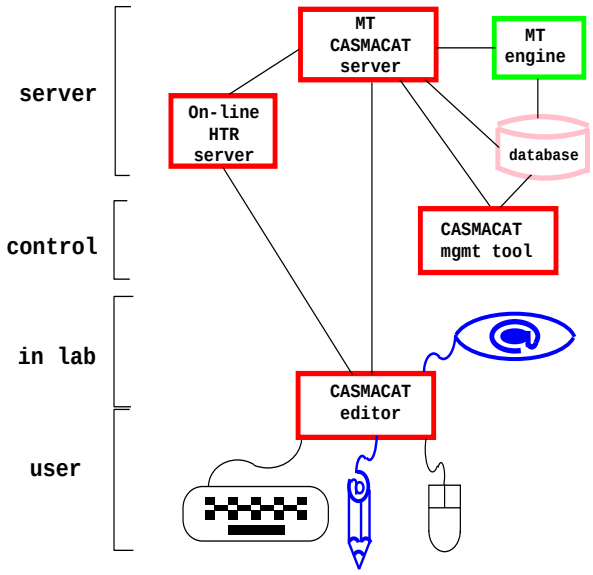

Figure 1: Major components of the CASMACAT workbench v.3

2.1 Machine translation server

The machine translation server converts the source text (in the form of XLIFF files) into a target text. The output is provided by the MATECAT (Bertoldi et al., 2012) component. This server works in parallel with Translation Memories (TMs) to retrieve the data from the translation server. TMs are basically a repository of previously translated segments. During the translation process, the translation server queries a TM to search for exact or fuzzy matches of the current source segment and these matches are then proposed to the translator as translation suggestions. When no matches are found in the TM, suggestions from the MT engine are supplied to the translator.

2.2 The editor

The CASMACAT editor is a web-based client with configurable visualization options for interactive translation prediction and interactive editing. The editor has several interfaces to communicate with a remote MT system via the CASMACAT MT server and it will be able to interface with an e-pen (Alabau et al., 2014b) in the next prototype. The editor features logging functions to record translator's keystrokes and mouse clicks as well as gaze activity captured by an eye-tracking device (i.e. EyeLink 1000).

Taking the CASMACAT workbench as a starting point, the SEECAT project aimed at testing the potential of speech recognition for translator-computer interaction. A description of the SEECAT workbench is provided in the next section.

3 The SEECAT workbench

The main aim of the SEECAT (Speech & Eye-Tracking Enabled Computer-Assisted Translation) project was to provide ASR as an input method for post-editing MT using the GUI of CASMACAT. The SEECAT workbench is able to recognize speech in English, Hindi and Spanish from any user without previous training.

User interaction is triggered after pressing the record button in the GUI to dictate text. The text to be replaced in the editor has to be selected before pressing the record button. The audio signal is then sent to the SEECAT server and the recognized text is sent back to the GUI. An example is shown in the figure 2.

Figure 3 shows the communication architecture between the client, the browser/GUI, the audio plug-in, the SEECAT server and the ASR server. It shows how the integration process is done with the server, and the client through the GUI. "Click

[2]SEECAT Project: Speech & Eye-Tracking Enabled Computer Assisted Translation. URL: http://bridge.cbs.dk/platform/?q=SEECAT

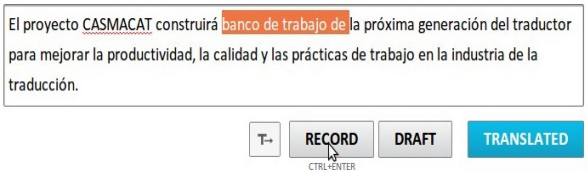

Figure 2: Recording functions in the SEECAT GUI.

RECORD" stands for clicking the record button to capture the speech signal, and "Click STOP" stands for clicking the stop button to stop the recording.

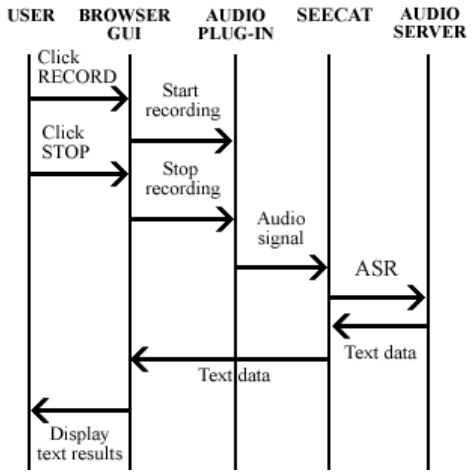

Figure 3: Case diagram for the interaction between browser, audio plug-in and server.

The data flow diagram proposed in the SEECAT project can be represented at a higher level as shown in Figure 4. The left part of the Figure 4 (CASMACAT) is represented in more detail in the previous Figure 1 and the right part of the Figure 4 (SEECAT) is described in more detail in Figure 5. CASMACAT sends the cursor position plus the recorded audio file to the SEECAT server. The SEECAT server sends back the ASR transcription.

3.1 Automatic Speech Recognition (ASR)

In SEECAT, AT&T Watson Toolkit has been trained for ASR in three languages, namely English, Hindi and Spanish.

3.1.1 English and Spanish

The English and Spanish ASR systems were provided by AT&T Labs-Research. Table 1 shows the details about the data. The data was recorded by female native speakers of the language.

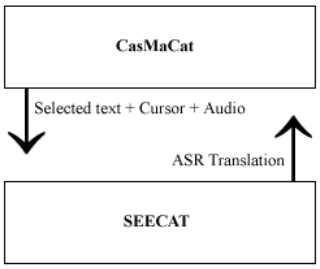

Figure 4: High level diagram of the proposed system.

Data Statistics	EN-ES	
	EN	ES
#sentences	7,792,118	7,792,118
#words	98,347,681	111,006,109
Vocabulary	501,450	516,906

Table 1: English and Spanish ASR data in SEECAT.

3.1.2 Hindi

The Hindi ASR was trained on more than 20 hours of audio data (7k training sentences) with transcriptions. The data was collected from various sources as described in table 2. The training details of Hindi ASR can be found in a previous work (Pandey et al., 2013).

Contributed By	Domain
KIIT	General text messages
McGill University	News
IIIT Hyderabad	Wikipedia Articles
SEECAT workshop	Text messages, Tourism

Table 2: Hindi ASR training data in SEECAT

For a test set of 67 sentences of a general domain, the recognition accuracy for Hindi ASR was 69.0.

3.2 SEECAT modules

SEECAT captures the speech signal using the WebRTC API(Bergkvist et al., 2012) (Web Real time communication) from the browser. WebRTC API is a browser API that enables browser to browser applications for voice calling/streaming, video streaming and peer-to-peer file sharing in real time communication without plug-ins. We have used this API for audio data. SEECAT uses SoX conversion for down sampling the speech signal from 16 khz to 8 khz. This signal is passed to

the ASR server for the recognition of the speech signal uttered by the user.

The CASMACAT logging functions have been extended with the information coming from ASR in order to be able to check when the ASR input starts and finishes.

Figure 5 describes how the SEECAT components interact.

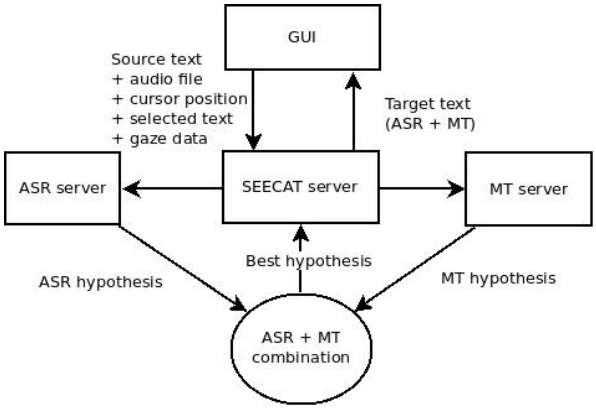

Figure 5: SEECAT architecture.

The SEECAT server receives from the GUI: The source text, an audio file, the cursor position, the selected text (optionally) and the gaze data whenever an eye-tracker device is connected. In order to improve the accuracy of the ASR module, the ASR in SEECAT is trained following previous work in this field (Paulik et al., 2005a) and (Paulik et al., 2005b). As described in Figure 5, in a post-editing task the ASR n-best hypotheses are generated using the audio file provided by the GUI to the SEECAT server. Also, the MT hypotheses are generated for the source string. These MT hypotheses are used to rescore ASR hypotheses. The experiments related to ASR MT integration are described in the following section.

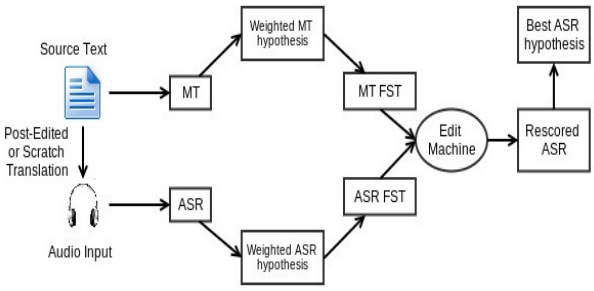

Figure 6: SEECAT combination of MT+ASR for better recognition.

In the future, we would also like to improve the translation output combining gaze data with ASR+MT output. The data from the eye-tracker will not only provide information about the translation process, but will also help to improve the output provided by the MT server base in gaze information coming from the user.

4 Experiments and results

This section presents experimental data using the current version of the SEECAT workbench.

4.1 Integration of ASR and MT

MT can improve ASR (Khadivi et al., 2006; Lecouteux et al., 2006) in a computer-assisted translation scenario. The same technique used to improve ASR through MT can be used with semantic information (Tammewar et al., 2013). In SEECAT, the hypotheses produced by ASR and MT are converted into lattices and are then composed using Edit Machine with the help of OpenFst toolkit (Allauzen et al., 2007). The synset information from WordNet is used while composing for the semantic matching of words. According to the edit distance scores, ASR hypotheses are rescored. We further extend this approach for the two language pairs Hindi-English and Spanish English, where the target language is English along with incorporating semantic information from English WordNet (Miller, 1995).

4.1.1 Experiments

In the Hindi-English MT system, it was found that the translated sentences were very poor and hence the POS tagger could not assign correct POS tags to the words. So we modified the technique to merge the senses not only from the predicted POS category but from all the four POS categories. This way the wrong POS tag will not affect the sense selection. Then this technique was also extended to Spanish-English system. This approach reduced the processing time, as now the POS tagger is not needed and time complexity is a very important factor in a real-time system such as SEECAT.

4.1.2 Results

For the evaluation, we used a test dataset of 132 sentences for Hindi-English and 96 sentences for Spanish-English. Table 3 enumerates the results for various experiments. Overall the word accuracy increased by *3.4%* for Hindi-English and *2.0%* for Spanish-English system over the baseline ASR. We performed the integration taking MT

hypotheses as sequence (Seq.) of words and unweighted (Unw.) bag of words and found that the latter strategy performs better (Tammewar et al., 2013).

Experiment		Language Pair		
		Hin-Eng		Span-Eng
POS		Yes	No	No
Only ASR		68.3		79.1
ASR+MT	Seq.	68.7		80.2
	Unw.	69.7		80.3
ASR+MT+Synset	Seq.	71.1	70.7	80.8
	Unw.	71.4	71.7	81.1

Table 3: ASR Word Accuracy in SEECAT.

There was not much difference in ASR word accuracy for experiments with POS and without POS tag, so we performed experiments without POS information for Spanish-English as the system performs faster in a real translation task without assigning POS tag for each word in the hypothesis.

4.2 Integration of ASR and Gaze

An eye-tracker plug-in has been integrated to the SEECAT interface to collect gaze information while a human translator interacts with the workbench. In a previous work (Kulkarni et al., 2013) was provided information on the use of gaze data to map gaze fixations to source words to improve ASR. For the integration, lattices were created and composed using the same ASR-MT composition.

Experiments showed that ASR as weighted bag-of-words and gaze as unweighted bag-of-words improved by 4.6% word accuracy in ASR for the English-Hindi pair.

4.3 Post-editing typing and using ASR

In this section the results of a pre-pilot and a pilot study assessing the potential of integrating ASR in a post-editing workbench are presented.

4.3.1 Pre-pilot test

Two native Spanish speakers volunteered to interact with the SEECAT workbench across the following six tasks:

- Task 1: Translation from scratch through typing (only using keyboard interaction)
- Task 2: Translation from scratch through ASR (only using speech interaction)
- Task 3: Post-editing through typing (only using keyboard interaction)
- Task 4: Post-editing through ASR (only using speech interaction)
- Task 5: Translation from scratch through typing + ASR
- Task 6: Post-editing through typing + ASR

Participant 1 was a professional translator while participant 2 did not have previous experience in translation. In each of these six tasks, the two participants worked from English into Spanish and the text domain involved in the experiments was tourism (the domain for which the ASR had previously been trained). Time to complete the task was considered as the dependent variable in order to measure the productivity gains derived from incorporating ASR as an input method for both translation from scratch and post-editing of MT.

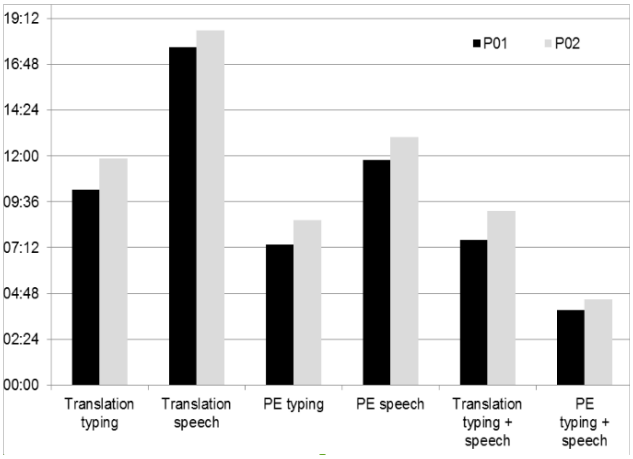

Figure 7: Time in minutes per tasks for participants 1 and 2.

Figure 7 shows overall times per task for the two participants in this pre-pilot test.

These preliminary results show that combining ASR and typing in post-editing tasks can produce faster turnaround when considering the task time overall as opposed to just providing ASR or typing as input method for the same tasks.

When looking at the time spent across individual segments in each of the six tasks for the two participants (see Figures 8 and 9), it can be seen that the majority of segments with the fastest turnaround belong to the post-editing task combining both ASR and typing.

In Figures 8 and 9, it is observed that the combination of ASR and typing requires the shortest time when compared to other tasks. This combination

of input methods could still benefit from enhancing the ASR module adding new vocabulary and new domains.

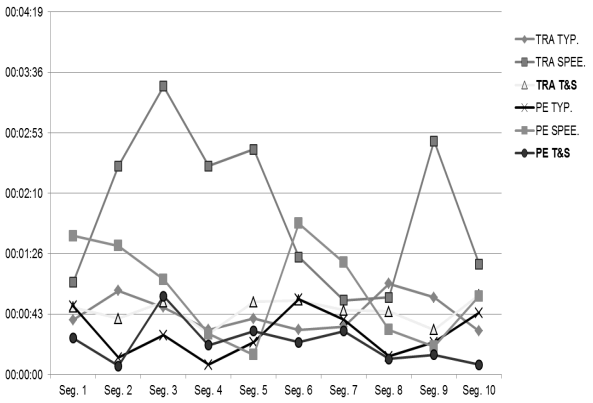

Figure 8: Time in minutes across segments and tasks for participant 1.

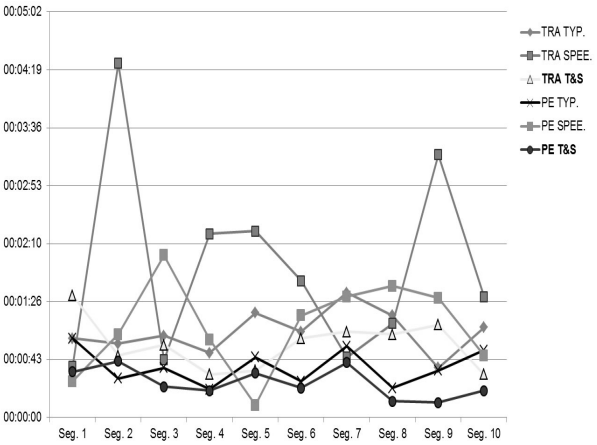

Figure 9: Time in minutes across segments and tasks for participant 2.

The results of this pre-pilot test encouraged the pilot test reported in the next section, where only post-editing tasks were included.

4.3.2 Pilot test

A group of 10 professional translators (7 women and 3 men) aged between 24 and 32 volunteered to perform the evaluation of the SEECAT workbench described in section 3. All participants had a degree in translation studies and were regular users of computer-aided translation tools (mainly SDL Trados and Déjà Vu X2). None of them had ever used ASR technology, but 90% of them claimed to have previous experience in post-editing MT as a professional service.

The pilot text involved two different texts (T1 and T2), of ten segments each, in the following two tasks:

1. Post-editing through typing (only using keyboard interaction)

2. Post-editing through typing + ASR

Task and text order were counterbalanced across participants. The language pair involved was English into Spanish and the text domain was tourism, the domain for which the ASR was trained. Following the design tested in the pre-pilot, this pilot study involved time to complete the task as the dependent variable and the input methods used while post-editing as the two independent variables, i.e. *i*) only typing or *ii*) typing and dictating (ASR).

Looking at the overall time spent to complete the task across participants (see Figure 10), 6 out of 10 benefited from integrating ASR as an input method, being able to complete the task faster than only typing. Participants P02, P03, P08 and P09 needed more time to complete the task when working with ASR. These four participants are also the ones who registered a greater time difference when comparing both tasks (up to an extra time-span of 4 minutes between task 1 and task 2 in the case of P02). There are no big time differences between the two tasks for the rest of the participants (12:30 minutes on average for the task involving only keyboard and 13:02 minutes for the task involving keyboard and ASR).

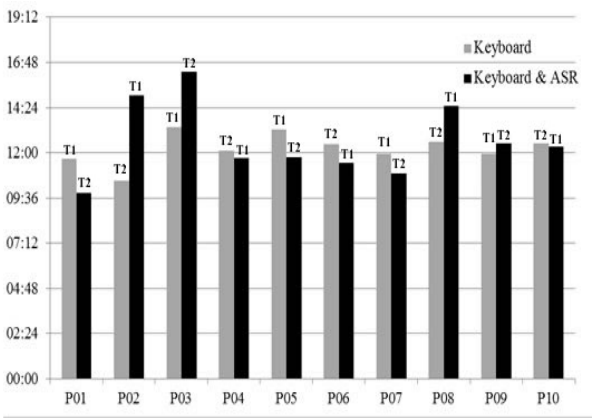

Figure 10: Time in minutes across tasks and texts using SEECAT.

When asked to provide feedback about the experimental tasks in a retrospective interview, all

participants stated that ASR seems to be a promising feature for a CAT workbench, but they all also underpinned that they would need more time to get acquainted with this technology in the context of post-editing.

5 Conclusions and future enhancements

As a result of the SEECAT project, ASR has been integrated to a computer-assisted translation tool as an additional input method. From these preliminary experiments, it seems reasonable to assume that working both with ASR and typing in post-editing tasks can be of help to boost translators' productivity. More experiments with a larger sample will have to be run in order to further explore the benefits of multimodal interaction both in translation and post-editing tasks. In addition, lab experiments showing that ASR can benefit from MT and semantic information for better re-scoring of ASR hypotheses have been presented.

Since WebRTC API has been used, future investigations will explore possibilities for online audio streaming of the data making the events synchronous rather than asynchronous. By doing this, we want to minimize the delay while the user receives the response from the system.

Future enhancements are foreseen integrating interactive machine translation and hand-written recognition using e-pen for the benefit of the human translator. More experiments in the context of professional translation over a longer period of time will be done to measure if productivity results increase after more hours of interaction with the workbench.

Acknowledgments

Work supported by the European Union 7^{th} Framework Program (FP7/2007-2013) under the CASMACAT project (grants agreement n$^{\text{o}}$ 287576), Copenhagen Business School, JSS Mahavidyapeetha and AT&T Labs-Research.

References

Alabau, Vicent, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, Luis Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Hervé Saint-Amand, Germán Sanchis-Trilles, and Chiara Tsoukala. 2014a. Casmacat: A computer-assisted translation workbench. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Software demonstrations.

Alabau, Vicent, Alberto Sanchis, and Francisco Casacuberta. 2014b. Improving on-line handwritten recognition in interactive machine translation. *Pattern Recognition*, 47(3):1217–1228.

Allauzen, Cyril, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. Openfst: A general and efficient weighted finite-state transducer library. In *Implementation and Application of Automata*, pages 11–23. Springer.

Bergkvist, Adam, Daniel C Burnett, Cullen Jennings, and Anant Narayanan. 2012. Webrtc 1.0: Real-time communication between browsers. *Working draft, W3C*.

Bertoldi, Nicola, Alessandro Cattelan, and Marcello Federico. 2012. Machine translation enhanced computer assisted translation. First report on lab and field tests. Available from: http://www.matecat.com/wp-content/uploads/2013/01/MateCat-D5.3-V1.2-1.pdf.

Chen, Fang. 2006. *Designing human interface in speech technology*. Springer.

Hauptmann, Alexander G and Alexander I Rudnicky. 1990. A comparison of speech and typed input. In *Proceedings of the Speech and Natural Language Workshop*, pages 219–224.

Khadivi, Shahram, Richard Zens, and Hermann Ney. 2006. Integration of speech to computer-assisted translation using finite-state automata. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 467–474. Association for Computational Linguistics.

Kulkarni, Rucha, Kritika Jain, Himanshu Bansal, Srinivas Bangalore, and Michael Carl. 2013. Mutual disambiguation of eye gaze and speech for sight translation and reading. In *Proceedings of the 6th workshop on Eye gaze in intelligent human machine interaction: gaze in multimodal interaction*, pages 35–40. ACM.

Lecouteux, Benjamin, Georges Linares, Pascal Nocéra, and Jean-François Bonastre. 2006. Imperfect transcript driven speech recognition. In *InterSpeech*.

Mesa-Lao, Bartolomé. 2012. The next generation translator's workbench: post-editing in casmacat v. 1.0. In *Proceedings of the 34th Translating and the Computer Conference, ASLIB*.

Miller, George A. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Pandey, Dipti, Tapabrata Mondal, SS Agrawal, and Srinivas Bangalore. 2013. Development and suitability of indian languages speech database for building watson based asr system. *Corpus*, 41282(90140):67522.

Paulik, Matthias, Christian Fügen, Sebastian Stüker, Tanja Schultz, Thomas Schaaf, and Alex Waibel. 2005a. Document driven machine translation enhanced asr. In *INTERSPEECH*, pages 2261–2264. Citeseer.

Paulik, Matthias, S Stuker, C Fugen, Tanja Schultz, Thomas Schaaf, and Alex Waibel. 2005b. Speech translation enhanced automatic speech recognition. In *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, pages 121–126. IEEE.

Tammewar, Aniruddha, Karan Singla, Srinivas Banglore, and Michael Carl. 2013. Enhancing asr by mt using semantic information from hindiwordnet.

Vidal, Enrique, Francisco Casacuberta, Luis Rodriguez, Jorge Civera, and Carlos D Martnez Hinarejos. 2006. Computer-assisted translation using speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(3):941–951.