Predicting Post-Editor Profiles from the Translation Process

Karan Singla karan.singla@students.iiit.ac.in
International Institute of Information Technology, Hyderabad, India
David Orrego-Carmona davidorregocarmona@gmail.com
Intercultural Studies Group, Universitat Rovira i Virgili, Tarragona, Spain
Ashleigh Rhea Gonzales ashleigh.gonzales@gmail.com
Department of Linguistics, Simon Fraser University, Burnaby, Canada
Michael Carl mc.ibc@cbs.dk
Copenhagen Business School, Copenhagen, Denmark
Srinivas Bangalore srini@research.att.com
AT&T Labs-Research, Bedminster, USA

Abstract

The purpose of the current investigation is to predict post-editor profiles based on user behaviour and demographics using machine learning techniques to gain a better understanding of post-editor styles. Our study extracts process unit features from the CasMaCat LS14 database from the CRITT Translation Process Research Database (TPR-DB). The analysis has two main research goals: We create n-gram models based on user activity and part-of-speech sequences to automatically cluster post-editors, and we use discriminative classifier models to characterize post-editors based on a diverse range of translation process features. The classification and clustering of participants resulting from our study suggest this type of exploration could be used as a tool to develop new translation tool features or customization possibilities.

1 Introduction

While significant strides have been made in statistical machine translation (MT) technology, the quality of fully automated MT systems is still a distant second to the quality of human translations. However, with the increasing demand for translation in the global market, the balance between quality and cost of translation is a trade-off many translation companies face. Human-in-the-loop translation techniques[1] aim to strike a balance between human and machine factors to optimize productivity. While the need for a human in the translation process loop is widely acknowledged, the possible techniques for improving the efficiency of the translator is largely open. In the ongoing CasMaCat project (Alabau 2013), there have been several techniques explored within the user interface designed for the translator to correct the MT output, such as automatic correction of the output based on the changes made by the post-editor, automatic replacement terminology when the post-editor corrects a term and active retraining of the MT model based on the changes made by the post-editor.

The task of post-editing is cognitively demanding; thus, it is expected that the post-editing tool factors in significantly to maximize end-user experience. A personalized post-editing tool

[1] also known as human-assisted MT, machine-assisted human translation and interactive MT

Proceedings of the Workshop on Interactive and Adaptive Machine Translation, pages 51–60
AMTA Workshop. Vancouver, Canada. September 22, 2014

that caters and adapts to a user's work style is bound to improve productivity metrics. To this end, we investigate techniques that help identify the post-editor behaviour profile using a multitude of factors tracked during the post-editing process. We study this process as a sequence of activity events that enable us to identify individual profiles. From the emergent patterns, we are then able to cluster post-editors into subgroups based on the commonalities of their individual process sequences. Our main motivation is that a higher level of granularity in the units that are analyzed would provide a more detailed account of the post-editing process. The identification of different post-editing styles and the definition of patterns in those styles at a fine-grained level provide insights for (a) the development and adaptation of translation tools, (b) classification of individual translators based on non-process factors (translator experience, translator personality, time constraints, etc.) and (c) the most salient skills required of post-editors, which can later be applied to translator training.

For the current study, we exploit the activity data tracked during the post-editing sessions to infer clustering and classification models. We investigate a range of machine learning (ML) techniques and validate the learned clusters against demographical metadata provided by the post-editors to demonstrate the veracity of the inferred models.

2 Related Work

The identification of translator and post-editor styles is an active field in Translation Process Research (TPR). To understand factors affecting translation workflow, researchers have explored activity data to identify patterns and define style taxonomies. This provides us with an understanding into the cognitive processes involved in translation tasks: It generates user-based knowledge for software development by considering the effects of training and experience (Carl and Schaeffer, forthcoming). However, no such study has applied a machine learning (ML) approach. Rather, the most widespread method to study translator style is the segmentation of the translation process into a limited number of subphases that broadly correspond to a preparation phase, a typing phase and a revision phase.

To identify factors and improve translation tools to better support users, Carl et al. (2011) establishes three phases in the translation process: Initial orientation, translation drafting and revision. Within each phase, the study further identifies different possible behaviours. Each translation phase and behaviour poses separate challenges, so a better choice of task support options for each phase can greatly benefit the end user.

Schrijver et al. (2009) also identifies three phases in the translation process: Pre-writing, writing and post-writing phase. The aim of the study is to explore transediting – the overlapping of translation and editing activities Stetting 1989. Considering the differences between the translation process and the transediting process, they configure two translation methods that vary dependent on where the first word of the target text originates. The second, more detailed method identifies nuances that prove important for the completion of the task and the product's adequacy with regards to the client's requirements.

Targeting post-editing specifically, Mesa-Lao (2013) suggests six steps that comprise the post-editing cycle, and identifies four cycles that are more common among post-editors, defining a more specific taxonomy for categorizing post-editing processes. Variation in post-editing styles is found to be dependent on the type of computer-assisted translation (CAT) tool GUI and the type of post-editor, which serves as an indicator of user adaptation to different conditions.

Lastly, Martínez-Gómez et al. (2014) employ a ML approach to translator activity sequence data to identify translator expertise. Surveying 800 translation sessions of an earlier version of the TPR database, they classify translators based on process features related to gaze fixations and keystroke activity. Notably, instead of defining translation activity subphases, their approach is to classify sequences of translation events (fixations and keystrokes) into distinct

activities to model the translation sessions. The error rate reduces when the analysis operates under the hypothesis of translator certification, and significantly when tasked with identifying translators' years of experience. In contrast with the current study, they focus on the prediction of expertise and years of experience, rather than the identification of translator profiles.

For the current research objectives, we implement generative and discriminative ML models to analyze the activity sequences in post-editing sessions. Profiling translators and post-editors based on fine-grained units of activity hint at different underlying cognitive processes that occur during translation; this analysis would provide grounds for further and deeper studies of the cognitive dimension of the translation process. The fact that our methods help identify relevant features for the post-editors classification can also provide the statring point to obtain actionable insights for developing better CAT tools.

3 Data

The data for the current study was extracted from the CasMaCat (Alabau 2013) longitudinal study (LS14) carried out during a six week period between April and May 2014 (CRITT TPR Database[2]). The training and adaptation factors are the most neglected aspects in post-editing research. Few TPR studies have addressed this issue (cf. Massey and Ehrensberger-Dow 2013), although it is commonly explored in research dealing with the development of translation competence and translator training in general (Pacte 2009, Göpferich 2009).

The LS14 study is the first of its kind that implements a longitudinal approach to assess how post-editors adapt to different GUI designs and work environments. The data collection includes five post-editors employed with a translation agency in Madrid, Spain. Participants used the CasMaCat workbench to perform the post-editing tasks (Ortiz-Martínez et al. 2012). Each week, each participant translated four texts under two conditions – Two texts with traditional post-editing (TPE) and two texts under interactive post-editing (IPE), which provides post-editors with real-time translation suggestions to aid in task completion – for a total of 24 texts and 120 translations sessions. All participants were native speakers of Spanish and translated from English into Spanish. The raw logging data included in the LS14 study is mainly derived from the post-editors' translation activities, extracted under the method detailed in Carl and Schaeffer (2013). Eye-tracking data was also collected for all post-editors, but only for the first and last weeks of the experiment, so only one-third of the files includes gaze data.

In order to identify the post-editor profiles and to conduct a benchmark study using ML techniques, we focus our analyses on the information logged in the post-editing session. We include three types of segmentation information derived from process unit file conventions extracted from the LS14 TPR-DB[3]): Activity units (CU), production units (PU), and translation segments (SG), which are detailed below.

CUid	Session	Time	Dur	TTseg	Type	Label
83	PE1_P1	480671	1839	1255	8	CU83-S:1255-T:8-D:1839
84	PE1_P1	482510	163	1255	4	CU84-S:1255-T:4-D:163
85	PE1_P1	482673	8202	1255	8	CU85-S:1255-T:8-D:8202
86	PE1_P1	490875	1526	1255	4	CU86-S:1255-T:4-D:1526

Table 1: Activity unit (CU) of post-editing activities from Participant 1 (PE1) in Segment 1255

[2]CRITT Translation Process Research (TPR) Database http://bridge.cbs.dk/platform/?q=node/18

[3]Carl and Schaeffer (2013) offer a detailed account of the data annotation methods and the different units used in the CRITT TPR Database

3.1 Activity Units (CU)

Features from the activity units serve as a baseline of user translation processes. The sequences within the translation session is a segmentation of typing, reading or pause activity recordings. We employ a dichotomous model: Activity is categorized as either Translation activity (Type 4) or No Activity (Type 8) to follow the conventions of Carl and Schaeffer (2013). To achieve finer-grained distinctions in the activity profile, we refine the activity labels with duration information of each event resulting in five additional classes centered around the median duration (in milliseconds). Furthermore, Part-of-speech (PoS) sequences extracted from the target text (TT) files are aligned with the CU data. There are in 68 unique PoS tags identified for Spanish in LS14, derived from TrEd/Treex (Pajas 2004, Popel and Žabokrtský 2010).

3.2 Production Units (PU)

Each production unit represents a coherent sequence of typing activity and includes information about the duration of the unit, duration of the preceding pause, number of edits, insertions and deletions, tokens involved in the source text and target text and average cross values. Cross values are the "relative local distortion of the reference text with respect to the output text, and indicate how many words need to be consumed in the reference to produce the next token(s) in the output" (Carl and Schaeffer 2013). A PU boundary is defined by a time lapse of more than one second between successive keystrokes.

3.3 Translation Segments (SG)

Translation segments provide sequence information of aligned source and target text segments detailing the segment production duration, character length, insertions and deletions and gaze data, when available. Average word entropy, cross values, perplexity, and source text literality were also calculated and appended to this file type, given the level of segmentation that our analysis required.

4 Experiments and Results

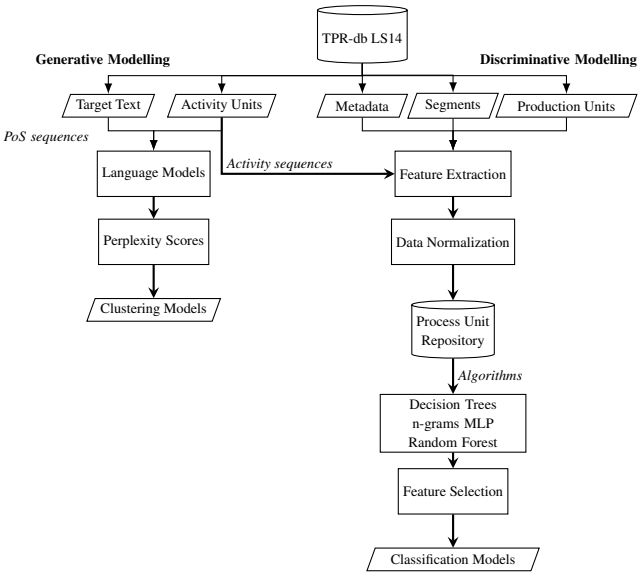

Figure 1: Basic pipeline of the study: Generative and discriminative models.

Category	Feature	Description
all	Participant	participant identifier
	Dur	duration of the unit
CU	Type	type of activity unit
	dur_cu	duration of activity units
	TokS	number of source tokens in the segment
	TokT	number of target tokens in the segment
	PoS	part of speech tag
CU, SG	LenS	character length of source segment
	LenT	character length of target segment
SG	Nedit	number of edits of the segment
	LenMT	character length of the machine translation segment
	Kdur	duration of coherent keyboard activity excluding keystroke pauses greater than or equal to five (5) seconds
	Fdur	duration of segment production time excluding keystroke pauses greater than or equal to 200 seconds
	Mins	Number of manually generated insertions
	Mdel	Number of manually generated deletions
	Ains	Number of automatically generated insertions
	Adel	Number of automatically generated deletions
	STent	average word translation entropy of the segment
	PP	perplexity score of the segment based on STent
	STlit	source text literality
	FixS	number of fixations on the source text unit
	FixT	number of fixations on the target text unit
	GazeS	total gaze time on source text unit
	GazeT	total gaze time on target text unit
	STcr2	cross value of source text token
	TTcr2	cross value of target text token
SG, PU	CrossS	cross value of source token
	CrossT	cross value of target token
	STseg	source segment identifier
	TTseg	target segment identifier
PU	Time	timestamp of the event
	ParalS	percentage of parallel source text reading activity during unit production
	ParalT	percentage of parallel target text reading activity during unit production
	Linear	degree of linear editing
	Pause	duration of production pause before typing onset

Table 2: Master process unit feature list

4.1 Toolkits

We use the Waikato Environment for Knowledge Analysis, WEKA 3.6 (Hall et al. (2009)) open-source toolkit for data mining and machine learning. Using several machine learning algorithms provided by the toolkit, we train various classification models. For the generative models, we use the SRI Language Modeling (SRILM) Toolkit (Stolcke 2002).

4.2 Claustering Post-Editors

Using the SRILM toolkit, we build n-gram models on Activity Unit sequences and target text PoS sequences of each post-editor. We use perplexity values as scores in a k-mean clustering to find similarity between post-editors, and then validate these clusters using the metadata.

Clustering Based on Activity Unit Sequences

The original CU files included in the TPR database contain eight types of activities. However, this classification of activity labels depends on the gaze information, which unfortunately is not

available across all points in our data. As such, we map the original eight categories into two:

- **Type 4 (Translation activity, T4)**: Activity units as defined by a sequence of coherent typing, which may also include gaze information; and,
- **Type 8 (No Activity, T8)**: Boundary between two activity units defined as a pause of 1000ms or more without any keyboard activity.

Under this modified categorization, because there are now only two types of activity, translation activity (Type 4) is always followed by a pause (Type 8). This creates a model in which only two transitions are possible (*T4-T8-T4* or *T8-T4-T8*). Therefore, we further subdivide Type 4 and Type 8 into five categories based on the duration of these events: Five buckets centered on the median duration, further partitioning the activity and pause units into five subgroups. Table 3 illustrates the generated sequences considering the duration of the translation and pause units. We create a standard trigram language model on the activity sequences of each post-editor. The

P01	T4,1	T8,3	T4,1	T8,2	T4,5	...
P02	T8,2	T4,3	T8,4	T4,1	T8,1	...
...						
P05	T4,1,,	T8,3	T4,2	T8,5	T4,2	...

Table 3: Sample user participant activity sequences bucketed by duration

language model of one post-editor is then used to calculate the perplexity scores of the activity sequences for all the other post-editors. Perplexity, PP, is often used for measuring the fit of a language model to a corpus of sequences. It can be interpreted as the average number of tokens that can be produced by a model at each point in the sequence. For a test set with tokens $W = w_1, w_2, ..., w_n$, the perplexity of a trigram model on the test set is

$$PP_W = \prod P(w_i \mid w_{i-1}, w_{i-2})^{-\frac{1}{n}}$$

where it can be noted that perplexity is normalized by the number of tokens in the test sequence.

Table 4 shows the perplexity scores of each post-editors language model on the other post-editor's activity sequences. It illustrates that the diagonal contains the smallest perplexity value since the dataset is the same as the one used to create the model.[4]

	PE1_LM	**PE2_LM**	**PE3_LM**	**PE4_LM**	**PE5_LM**
PE1	*4.09526*	4.3195	4.62186	4.84951	4.39231
PE2	4.30064	*4.06063*	4.41296	4.60593	4.40357
PE3	4.63742	4.39636	*4.06999*	4.30479	4.85385
PE4	4.47429	4.29059	3.99274	*3.80005*	4.88682
PE5	4.00879	4.09205	4.46445	4.81527	*3.80372*

Table 4: Perplexity scores for the Activity sequence LM model for all post-editors

We use the perplexity values as distance costs in a k-means clustering algorithm to produce two ($k = 2$) clusters. We obtain the following clusters: $Cluster1\{PE1, PE2, PE5\}$ and $Cluster2\{PE3, PE4\}$. When looking for possible explanations in the metadata, we found that $cluster1$ includes the most experienced post-editors. Based on the findings provided by this clustering, it seems to be the case that experienced post-editors produce similar kinds of activity sequences in contrast with the activity sequences of inexperienced post-editors.

[4]However, an exception as seen in Table 4, PE 3's activity model has a higher perplexity score on PE 3's sequence compared to that on PE 4's activity sequence.

Clustering Based on Target Text Part-of-Speech Sequences

We extract the PoS sequences for each segment in the target text and created a n-gram language model for each post-editor (PE). Then we use this model to calculate the perplexity values of the language model for all other post-editors to measure the appropriateness of the model. Using the perplexity scores as distance metrics, we grouped the post-editors into two clusters by applying standard k-means clustering: $Cluster1\{PE1, PE3, PE5\}$ and $Cluster2\{PE2, PE4\}$. To account for this clustering, we compare the results with the participant metadata. We find that Post-Editor 2 and Post-Editor 4 share a very negative response to the post-editing approach, whereas the other three participants did not indicate such apprehension towards the task. Considering our data, this seems to indicate that post-editors with similar negative response towards post-editing tend to have similar activity patterns.

4.3 Discriminating Post-Editors

Unlike generative modelling which clusters the post-editors based on their shared characteristics, discriminative modelling is done to determine if the ML models are able to identify the five post-editors based on their activity profiles. We carry out tests on the three types of data mentioned in Section 3: activity units, productions units and translation segments. We segment the data to analyze the effect of the GUI (traditional post-editing and interactive post-editing) in the analyses. We apply various ML algorithms with 10-fold cross validation for classification, but find that "multilayer perceptron" and "classification via regression" perform best for this task of identifying the post-editors. The baseline accuracy is 20% given that there are the same number of samples for the five participants.

	Algorithm	Traditional PE	Interactive PE	Combined
Activity Unit Profile	Multilayer Perceptron	40.58 %	35.51 %	41.54 %
	Classification via Regression	42.37 %	36.82 %	42.72 %
Production Unit Profile	Multilayer Perceptron	44.67 %	39.82 %	37.06 %
	Classification via Regression	45.83 %	47.69 %	46.48 %
Translation Segment Profile	Multilayer Perceptron	42.88 %	46.93 %	44.45 %
	Classification via Regression	44.64 %	47.51 %	45.71 %

Table 5: Results for the 5-way classification task to discriminate post-editors based on activity, production and translation segments profiles created at the segment level.

Activity Unit Profile

Table 5 shows results obtained from frequencies of unigrams and bigrams of activities as features for discriminating post-editors. It illustrates that the model is able to discriminate post-editors better when they use the traditional PE GUI, with 42.37% accuracy, compared to 36.82% in the Interactive PE environment. However, when the data is combined using the GUI as an additional feature, accuracy of the model remains almost the same at 42.72%.

Production Unit Profile

We create a features matrix using the PU features described in Table 2 to identify post-editors. In the matrix, all text dependent features have been normalized using *LenS* (character length of source sentence) to ensure that the system is not biased by differences in the length of the text. Considering there are multiple production units for each segment, and that the number of PUs vary per post-editor, we make a sparse vector to group together the different production units of each segment. As shown in Table 5, we achieve 46.48% accuracy while using the entire data set with GUI as a feature. When dividing the data set depending on the GUI, we achieve an

accuracy of 47.69% and 45.83% with the system discriminating post-editors in the traditional and interactive enabled GUI, respectively.

Translation Segment Profile

Translation segments have some features overlapping with production unit profiles as detailed in Table 2. Nevertheless, in this file, all the information is cumulative for a segment and dependent on the text, while in the production unit files, the information is created based on the post-editors' typing bursts. When testing the combined dataset including the data from the two GUIs, the model has an accuracy of 45.71%. When running the tests independently for the two GUIs, the TPE dataset achieves 44.64% of accuracy, while the IPE dataset reaches 47.51% of accuracy.

4.4 Feature Analysis

To serve as a clearer visualization of the features identified as salient by the classifiers, we present in Figure 2 a detail of a decision tree learned using a J4.8 classifier. The most relevant features to classify post-editors are related to different types of duration (*Fdur, Kdur, Dur*) and the post-editors' typing activity (*Mins, Nedit*).

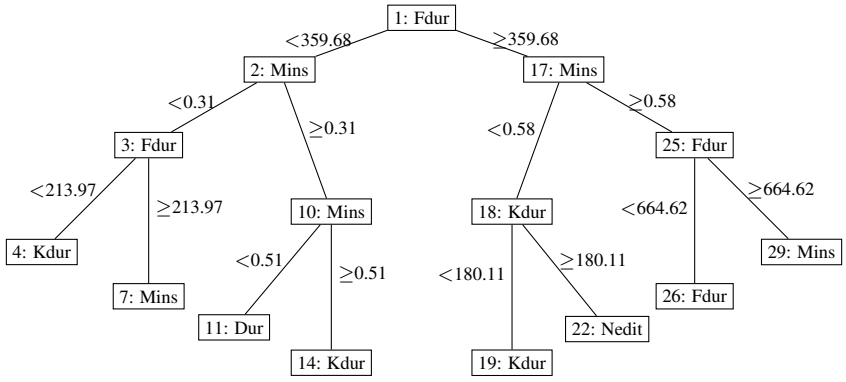

Figure 2: Decision tree of salient features from Translation Segments (SG)

5 Discussion

In this paper, we test the hypothesis that events that make up the translation process provide enough information for the individualization of post-editor profiles. By using machine learning models, we are able to not only find the post-editors' profiles, but also cluster and discriminate between post-editors. Classifying post-editors based on activity microunits, either dependent on the text or on the individual user, provides interesting results that are worth exploring in translation studies and computer science. However, since only a few post-editors participated in this pilot collection, the current study should be considered only as an initial exploration of such methods on translation process data. Considering our initial results, it would be beneficial to explore how additional features on a different segmentation level affect the models, and to what degree, if at all. For example, information related to user personality, user training and experience, testing conditions, genre of the text and other qualitative features can be added to the existing models to explore non-activity factors.

6 Conclusion

Computer assisted translation remains a progressive field of research, and there is an ever-growing interest in providing translators and post-editors with better software tools to facilitate their work and increase productivity. Identifying how translators interact with the tools and gain insights into features that have an impact on their performance can help in the development of a new generation of translation tools. TPR aims at uncovering the cognitive process that unfold in the translator's mind while performing the translation tasks. Although our sample is undoubtedly limited consisting of data from five participants only, our results can serve as indicator of an avenue that starts providing interesting consideration that could be further explored at a bigger and more comprehensive scale. The insights brought forth from this study are gathered under the goal of performance improvement through different channels: (1) Providing better tools and (2) uncovering training needs. The empirical methods of the current study provide the foundation for further exploration of the translation process in order to satisfy the needs in those areas. We believe our methods of user participant profiling can be adapted and extrapolated to analyze different translation processes and provide researchers with solid findings for multiple applications in the field.

7 Acknowledgments

This work was supported by EU's 7th Framework Program (FP7/2007-2013) under grant agreement 287576 (CASMACAT).

References

Alabau, V. (2013). Web technologies in casmacat. Interactive machine translation. Speech & Eye-Tracking Enabled CAT (SEECAT). Copenhagen, Denmark,.

Carl, M., Dragsted, B., and Jakobsen, A. (2011). A Taxonomy of Human Translation Styles. *Translation Journal*, 16(2):n.p.

Carl, M. and Schaeffer, M. J. (2013). The CRITT Translation Process Research Database v1.4. http://bridge.cbs.dk/resources/tpr-db/TPR-DB1.4.pdf.

Carl, M. and Schaeffer, M. J. (forthcoming). Processes of Literal Translation and Post-editing.

Göpferich, S. (2009). Towards a model of translation competence and its acquisition: the longitudinal study 'transcomp. In Göpferich, S., Jakobsen, A. L., and Mees, I. M., editors, *Behind the Mind: Methods, Models and Results in Translation Process Research*, Copenhagen Studies in Language 37, pages 11–37. Copenhagen.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations*, 11(1).

Martínez-Gómez, P., Minocha, A., Huang, J., Carl, M., Bangalore, S., and Aizawa, A. (2014). Recognition of Translator Expertise using Sequences of Fixations and Keystrokes. In Qvarfordt, P. and Hansen, D. W., editors, *Proceedings of Symposium on Eye Tracking Research and Applications*, pages 299–302, New York, USA.

Massey, G. and Ehrensberger-Dow, M. (2013). Evaluating translation processes: opportunities and challenges. In Kiraly, D., Hansen-Schirra, S., and Maksymski, K., editors, *New Prospects and Perspectives for Educating Language Mediators*, Translation Studies Series 10, pages 157–180. Gunter Narr, Tübingen.

Mesa-Lao, B. (2013). Eye-tracking Post-editing Behaviour in an Interactive Translation Prediction Environment. In *Proceedings of the 17th European Conference on Eye Movements*, Lund, Sweden.

Ortiz-Martínez, D., Sanchis-Trilles, G., Casacuberta, F., Alabau, V., Vidal, E., Benedí, J.-M., González-Rubio, J., Sanchis, A., and González, J. (2012). The CASMACAT Project: The Next Generation Translator's Workbench. In *Proceedings of the 7th Jornadas en Tecnología del Habla and the 3rd Iberian SLTech Workshop (IberSPEECH)*, page 326–334.

Pacte (2009). Results of the validation of the pacte translation competence model: Acceptability and decision making. *Across Languages and Cultures*, 10(2):207–230.

Pajas, P. (2004). Tred tree editor. http://ufal.mff.cuni.cz/tred/.

Popel, M. and Žabokrtský, Z. (2010). Tectomt: Modular nlp framework. In *Lecture Notes in Computer Science, Vol. 6233, Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL 2010)*, pages 293–304, Berlin/Heidelberg. Springer.

Schrijver, I., van Vaerenbergh, L., and van Waes, L. (2009). Transediting in students' translation processes. *Artesis VT working papers*, 1:1–31.

Stetting, K. (1989). Transediting: A new term for coping with the grey area between editing and translating. In Caie, G., Haastrup, K., and Arnt Lykke Jakobsen, e. a., editors, *Proceedings from the Fourth Nordic Conference for English Studies*, pages 371–382, Copenhagen: University of Copenhagen.

Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904, Denver, USA.