informatics

MDPI

Article

Improving the Translation Environment for Professional Translators

Vincent Vandeghinste [1,*,†], Tom Vanallemeersch [1,‡], Liesbeth Augustinus [1], Bram Bulté [1], Frank Van Eynde [1], Joris Pelemans [2,§], Lyan Verwimp [2,‖], Patrick Wambacq [2], Geert Heyman [3,¶], Marie-Francine Moens [3], Iulianna van der Lek-Ciudin [4], Frieda Steurs [4,†], Ayla Rigouts Terryn [5], Els Lefever [5], Arda Tezcan [5], Lieve Macken [5], Véronique Hoste [5], Joke Daems [5], Joost Buysschaert [5], Sven Coppers [6], Jan Van den Bergh [6] and Kris Luyten [6]

1 Centre for Computational Linguistics, KU Leuven, B-3000 Leuven, Belgium; tallem@ccl.kuleuven.be (T.V.); liesbeth.augustinus@kuleuven.be (L.A.); bram.bulte@ccl.kuleuven.be (B.B.); frank.vaneynde@kuleuven.be (F.V.E.)
2 Department of Electrical Engineering, KU Leuven, B-3001 Leuven, Belgium; joris.pelemans@esat.kuleuven.be (J.P.); lyan.verwimp@esat.kuleuven.be (L.V.); patrick.wambacq@esat.kuleuven.be (P.W.)
3 Department of Computer Science, KU Leuven, B-3001 Leuven, Belgium; geert.heyman@kuleuven.be (G.H.); sien.moens@kuleuven.be (M.-F.M.)
4 Department of Linguistics @ Antwerp, KU Leuven, B-2000 Antwerp, Belgium; iulianna.vanderlekciudin@kuleuven.be (I.v.d.L.-C.); frieda.steurs@kuleuven.be (F.S.)
5 Language and Translation Technology Team (LT3), Ghent University, B-9000 Ghent, Belgium; ayla.rigoutsterryn@ugent.be (A.R.T.); els.lefever@ugent.be (E.L.); arda.tezcan@ugent.be (A.T.); lieve.macken@ugent.be (L.M.); veronique.hoste@ugent.be (V.H.); joke.daems@ugent.be (J.D.); joost.buysschaert@ugent.be (J.B.)
6 Expertise Centre for Digital Media, Flanders Make–tUL–UHasselt, B-3590 Diepenbeek, Belgium; sven.coppers@uhasselt.be (S.C.); jan.vandenbergh@uhasselt.be (J.V.d.B.); kris.luyten@uhasselt.be (K.L.)
* Correspondence: vincent@ccl.kuleuven.be; Tel.: +32-16-325-089
† Current address: Instituut voor de Nederlandse Taal, 2311 GJ Leiden, The Netherlands.
‡ Current address: CrossLang, B-9050 Gentbrugge, Belgium.
§ Current address: Apple Inc., Cupertino, CA 95014, USA.
‖ Current address: Apple Inc., 52062-52080 Aachen, Germany.
¶ Current address: Nokia Bell Labs, B-2018 Antwerp, Belgium.

Received: 23 April 2019; Accepted: 13 June 2019; Published: 20 June 2019

Abstract: When using computer-aided translation systems in a typical, professional translation workflow, there are several stages at which there is room for improvement. The SCATE (Smart Computer-Aided Translation Environment) project investigated several of these aspects, both from a human-computer interaction point of view, as well as from a purely technological side. This paper describes the SCATE research with respect to improved fuzzy matching, parallel treebanks, the integration of translation memories with machine translation, quality estimation, terminology extraction from comparable texts, the use of speech recognition in the translation process, and human computer interaction and interface design for the professional translation environment. For each of these topics, we describe the experiments we performed and the conclusions drawn, providing an overview of the highlights of the entire SCATE project.

Keywords: computer-aided translation; machine translation; speech translation; translation memory-machine translation integration; user interface; domain-adaptation; human-computer interface

1. Introduction

The SCATE project (Smart Computer Aided Translation Environment) was a four year research project that ran from March 2014 to February 2018, in which a consortium of three Flemish universities investigated several aspects and stages in the professional translation workflow, aiming at improvements in each of them. This paper describes the highlights of our research.

Figure 1 provides an overview of most parts of the project, through a prototype user interface (which is described in detail in Section 6) of the *Smart Computer-Aided Translation Environment*. A demo version of the prototype is available at http://scate.edm.uhasselt.be/). A professional translates a sentence under translation (A) using a large text entry box centrally on the screen (B). The central placement provides space for context—(G) preceding and (H) subsequent sentences, overall translation progress ((I)) and configuration options in the top bar. Autocomplete (F) assists translators during their task. Suggestions come from multiple sources, all related to technologies developed or improved within the project. A translator can accept the default suggestion, choose an alternative term from the presented options (D) or start typing a different translation.

When starting the translation of a sentence, the default translation comes from hybrid machine translation. The complete translated sentence is presented immediately below the text box (C) and can be copied using a single shortcut. The hybrid machine translation builds on the research on both machine translation and fuzzy matching, discussed in Section 2. As other results from fuzzy matching can help during translation, the top results are also presented to the translator in (E). At the right-hand side of both the hybrid machine translation (C) and fuzzy matches (E) quality estimations are presented. Research on quality estimation is described in Section 3. The relevant terms of these fuzzy matches are also presented in the list of alternative (D), just as results from an automatically extracted term list, for which frequency information in the source is also presented. Results on the topic of term extraction are discussed in Section 4. The integration of speech recognition in the translation process is described in Section 5.

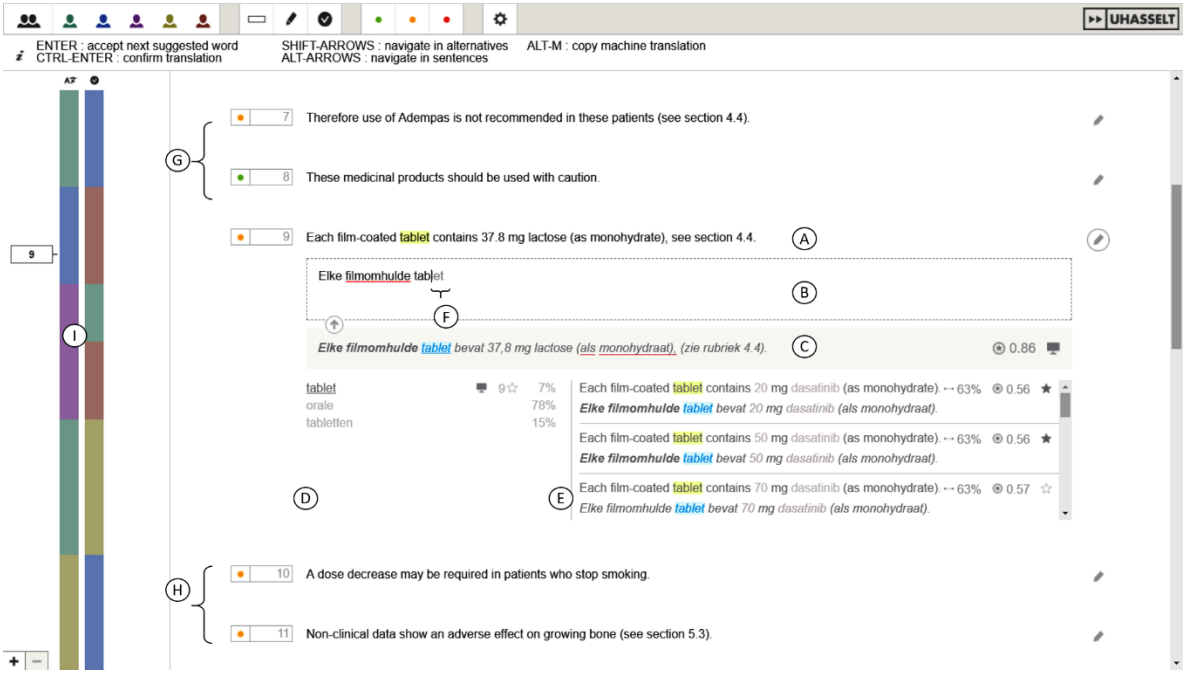


Figure 1. An overview of the SCATE interface. (A) The sentence to translate, (B) the editing field, (C) the hybrid MT that also includes pretranslations, (D) a list of translation alternatives coming from the term base, TM and MT, (E) fuzzy matches, (F) suggestion from autocomplete, (G) previous source sentences, (H) upcoming source sentences and (I) a progress bar.

2. Translation Technologies

Amongst the main translation technologies, besides a term-base (TB), that are accessible to most translators in their professional CAT environment are a TM system and an MT engine. Section 2.1 describes how a TM system can improve the matching of existing translations with the segment to translate. Section 2.2 investigates integrating TM and MT technologies. Section 2.3 describes our efforts in the creation and accessibility of parallel treebanks (i.e., syntactically annotated parallel sentences) for syntax-based MT.

2.1. Improved Fuzzy Matching

CAT tools have become indispensable in the environment of the modern translator. They help increase consistency, productivity and quality. One of the core components of a CAT tool is the TM system, which contains a database of already translated fragments, the TM. Given a sentence to be translated, the traditional TM system looks for source language sentences in a TM which are identical (*exact matches*) or highly similar (*fuzzy matches*) and, upon success, suggests the translation of the matching sentence to the translator.

Similarity calculation can be done in many ways. In current TM systems, fuzzy matching techniques mainly consider sentences as simple *sequences of words* and contain very limited linguistic knowledge, such as stop word lists. Few tools use more elaborate linguistic knowledge. We include *syntactic information* for detecting TM sentences which are not only similar when comparing words but also when comparing the syntactic information associated with the sentences. Such information can consist of lemmas, part-of speech tags or syntax parse trees. We investigate whether such abstract, syntax-based matching is able to assess the usefulness of matches in a better way than methods purely based on sequences of words. The fuzzy matching metrics we use are not only the string based metrics such as *Levenshtein distance* [1], *Translation Edit Rate* (TER) [2], *Percent Match* and *n-gram precision* [3] (a sentence-based metric very similar to BLEU [4]). We apply these metrics also on strings of lemmas and also use METEOR [5]. Furthermore we test *tree-based* metrics, such as *shared partial subtree matching* and *n-gram precision for head word chains*. We also experiment with fuzzy matching of flattened tree representations, such as *Prüfer sequences* [6].

We designed a flexible and time-efficient framework which applies and combines different metrics in the source and target language. We measure the correlation of fuzzy matching metrics scores with the evaluation score of the suggested translation to find out how well the usefulness of a suggestion can be predicted and we measure the difference in recall between fuzzy matching metrics by looking at the improvements in mean TER as the match score decreases.

Our comparison of the baseline matching metric, *Levenshtein distance* [1], with linguistically aware and unaware matching metrics, has shown that the use of linguistic knowledge in the matching process provides clear added value, especially when several metrics are combined into a new metric using a regression tree. The correlation of combined metrics with the evaluation score is much stronger than the correlation of the baseline. Moreover, there is significant improvement in mean evaluation score and the difference in recall with the baseline increases as match scores decrease. Full details of this study can be found in Reference [7].

The improved fuzzy matching system is implemented as a web service available through an application programming interface (API) and is used in the SCATE interface prototype, as shown in Figure 2. This prototype is discussed in more detail in Section 6.

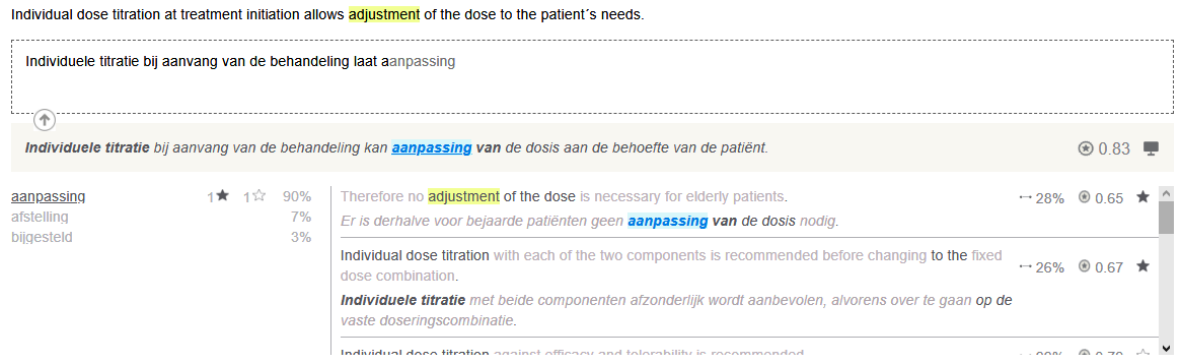


Figure 2. Fuzzy matches (bottom right) and integrated TM-MT suggestion (middle) in the prototype.

2.2. Integration of Translation Memory with Machine Translation

We test the integration of MT and TM, in order to increase the quality of, and potentially the confidence in, MT output, in a similar way as in Reference [8]. The TM-MT system consists of two main components: (1) fuzzy match repair, that is, the automatic editing of close matches found in the TM and (2) span pre-translation, in the context of which MT output is constrained by including certain consistently aligned subsegments coming from one or more TM matches. Both components use a TM with fuzzy matching techniques and a statistical MT (SMT) system with related alignment information. Different metrics are used for the retrieval and scoring of fuzzy matches, including the syntactic fuzzy matching metric described in Section 2.1. We performed experiments on ten language pairs (English ↔ German, French, Hungarian, Dutch and Polish) which involve multiple language families, using the DGT dataset [9]. We applied phrase-based SMT without span pre-translation [10], *pure* TM and a recurrent neural network (RNN) encoder-decoder neural MT (NMT) system [11] as baselines and evaluated the translations using several metrics. The tests show that this approach has potential. As shown in Figure 3, significantly higher BLEU scores [4] for nine of the ten language combinations were reported and also METEOR [5] and TER [2] scores show comparable patterns. More details are available in Reference [12]. The system is, as shown in Figure 2, also integrated in the SCATE prototype, which provides translators with informed MT output and which is described in Section 6.

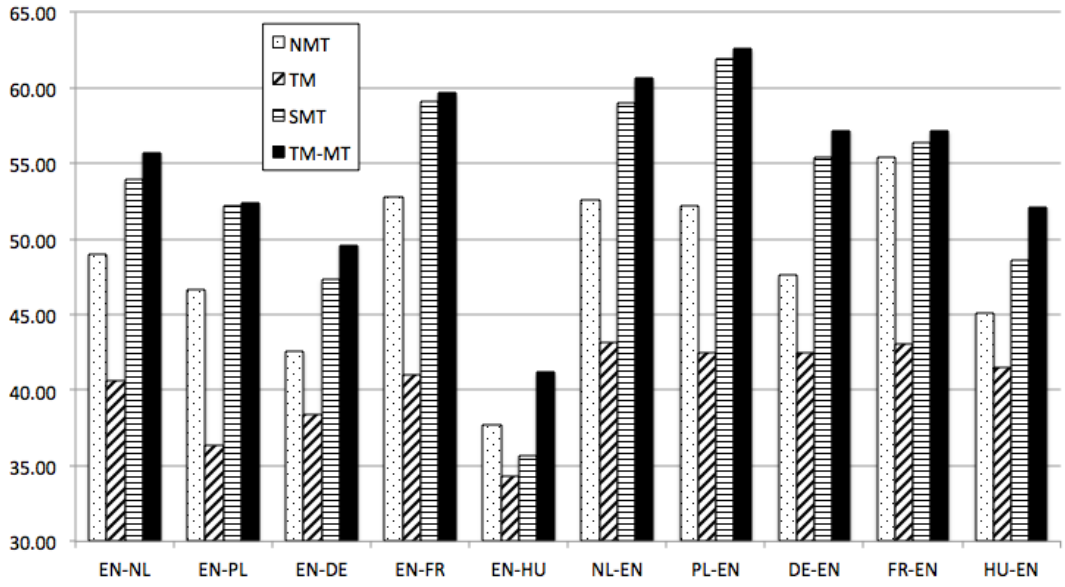


Figure 3. Overview BLEU scores TM-MT integration and baselines [12].

2.3. Building Resources for Syntax-Based Translation

The SCATE project was conceived before the change of the MT world towards the *neural* paradigm. As one of the goals of the SCATE project was to improve syntax-based MT engines, a substantial amount of work was dedicated to improving the data preprocessing of the resources for what, at the time, seemed to be one of the most promising approaches towards MT.

Parallel treebanks [13] are syntactically annotated versions of parallel corpora. While the latter are traditionally used in data-driven MT systems, such as *phrase-based SMT* or *NMT* [14], parallel treebanks can be used to improve syntax-based statistical MT ([15,16]) by taking advantage of linguistic information, allowing higher levels of abstraction than in phrase-based SMT.

Work on parallel treebanks also has potential to improve *tree*-based NMT, which is a very recent research topic. *Tree-to-string* approaches are, amongst others, described in References [17,18] and *string-to-tree* approaches in, amongst others, References [19,20]. While we are not aware of any *tree-to-tree* approaches in NMT (yet), we consider it only a matter of time before such approaches appear, as such techniques are already being used, for example, computer program translation between different programming languages [21].

Below, we explain the concept of alignment (Section 2.3.1), leading to results like parallel treebanks and the creation of MT rules from alignments (Section 2.3.2). We explain the SCATE work on enriching parallel treebanks with semantic information in order to bridge syntactic divergences and to facilitate MT rule creation (Section 2.3.3) and the work on allowing to search parallel treebanks (Section 2.3.4).

2.3.1. Sub-Sentential Alignment

Alignment consists of linking segments of a source text with translation-equivalent segments of the target text, that is, the translation of the source text. Starting at the *document level*, alignment is usually performed using an iterative refinement strategy. Alignment proceeds at the *sentence level* and may continue at the *sub-sentential level* and the *word level*.

Sentence alignment is more or less considered a solved problem, at least for parallel documents (cf. http://www.statmt.org/survey/Topic/SentenceAlignment for an overview). Sub-sentential alignment consists of aligning elements below the sentence level, such as words, chunks or constituents at deeper levels of syntactic hierarchy. Word alignment deals with issues such as NULL links (untranslated words or words added during translation), crossing links (changes of order of words during translation) and fuzzy links (e.g., translation of groups of words as a whole rather than as individual words). Word alignment in sentence pairs is typically produced using statistical tools such as GIZA++ [22], which also create a set of lexical probabilities based on the word alignments of a large set of sentence pairs. These probabilities indicate the likelihood a source word is translated by a target word or vice versa. The word alignment and lexical probabilities allow for the alignment of word groups, aligned groups being integrated into a so-called phrase table for SMT systems. Sub-sentential alignment may apply linguistic information by aligning chunks [23], which result from a superficial syntactic analysis of a sentence (detection of the boundaries of noun phrases and verb phrases) or by aligning nodes in parse trees, which provide a deep syntactic hierarchy of a sentence.

We focus on the alignment of nodes in syntactic parse trees (a.k.a. tree alignment), as this allows more flexible translation patterns for MT engines than mere word alignment. Several tree aligners exist ([24–26]) taking syntactic parse trees as input, using word alignments and lexical probabilities as input. Tree alignment leads to parallel treebanks. In other words, such treebanks [13] are syntactically annotated versions of parallel corpora.

2.3.2. Machine Translation Rules

Based on alignment results, translation rules can be created. Data-driven MT systems such as phrase-based SMT [10] and NMT [11], at least in its standard form, use parallel corpora without annotations. Parallel treebanks, on the other hand, can be used to create syntax-based MT rules and

hence to develop syntax-based statistical MT systems ([15,16]). The linguistic information incorporated in their rules allows for higher levels of abstraction and more flexible patterns than the rules derived from non-annotated corpora. Figure 4 shows a sub-sententially aligned pair of parallel trees.

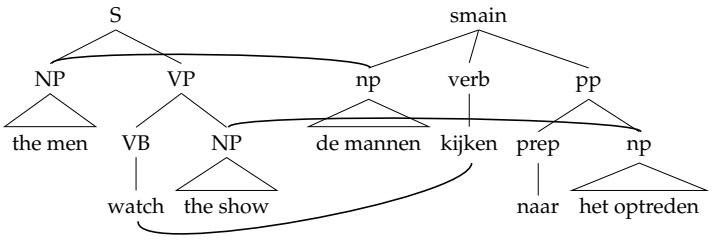


Figure 4. An example node-aligned parallel tree (Gloss of the Dutch sentence is "*the men look at the show*").

The translation-equivalent sentences in parallel corpora may show syntactic divergences, that is, use different syntactic means to convey the same meaning as a result of linguistic necessities or translators' choices. This makes alignment based on syntactic structure complex.

2.3.3. Semantic Information

While the syntactic structure of sentences often changes during translation, semantic information tends to remain constant. Therefore, we investigated whether aligning parse trees based on such information facilitates alignment and leads to higher quality MT rules with respect to alignment purely based on syntactic information. We focus on shallow semantics, in the form of predicates and roles. We apply a five-step approach in order to obtain semantically motivated MT rules:

Step 1: Creation of a semantic role labeler. As tools for automatically assigning semantic predicates and roles are scarce resources, we apply a crosslingual projection approach and train a semantic role labeler from the projected information. We annotate syntactic parse trees in the resource-rich language (English) with a semantic role labeler and project the predicate and role labels to the syntactic parse trees in the target language (Dutch) through a non-linguistic tree aligner, LENG (Lexically Equivalent Node Grouping) [27], which we developed in SCATE. Details of this aligner can be found below.

Step 2: From the projected labels, we train a semantic role labeler, requiring a minimum of manual intervention. The labeler contains a model with mappings between syntax and semantics.

Step 3: We align parse trees via semantic labels, word alignment and lexical probabilities.

Step 4: We derive translation rules based on the aligned parse trees.

Step 5: We extend a phrase-based SMT system with the translation rules.

Evaluation results for step 3 and 5 indicate that enriching parse trees with semantic predicate and role labels leads to more precise tree alignment results and that combining a phrase table with semantic translation rules helps in improving translation quality. While we performed tests on the language pair English-to-Dutch, our approach is sufficiently generic for tests on other language pairs. More details can be found in Reference [27].

The LENG tree aligner, being non-linguistic, may also be applied in a broader context, beyond semantically motivated MT. It combines the language pair and parser independence of Reference [26] with the higher performance of Reference [25]. It looks for pairs of isomorphic source and target subtrees in which pairs of nodes show a strong lexical equivalence. The tree alignment consists of linked subtree pairs that do not overlap with each other. As opposed to Reference [26], LENG does not only use lexical probabilities but also the word alignment of the sentence pair (similarly to [25]), imposes less well-formedness constraints and only links nodes to each other if there is strong evidence for doing so.

We compared LENG with References [25,26] on the last 35 sentences in the 125-sentence Lingua-Align gold standard, using the lexical probabilities and word alignment included with the

gold standard. Evaluation statistics are shown in Table 1. It shows that we clearly perform better than Reference [26] on precision, recall and F-score and also outperform Reference [25].

Table 1. Subtree alignment accuracy on English-Dutch gold standard. Best scores are set in boldface.

System	Precision	Recall	F-Score
SubTree Aligner [26]	69.30	71.55	70.40
Lingua-Align [25]	79.29	88.78	83.77
LENG	**83.48**	**89.96**	**86.60**

2.3.4. Searching Parallel Treebanks

Parallel treebanks can not only be used for creating MT rules but also as a resource for studying translation phenomena. We built an updated version (with improved parses and improved alignment) of the parallel Europarl treebank for Dutch and English [13]. This treebank is *tree aligned* (see also Section 2.3.1) and can be queried with Poly-GrETEL [28].

Poly-GrETEL, developed within the SCATE project, is an online tool (http://gretel.ccl.kuleuven.be/poly-gretel/) which enables example-based syntactic querying in parallel treebanks and which is based on the monolingual GrETEL (Greedy Extraction of Trees for Empirical Linguistics) environment [29]. The tool provides online access to the Europarl parallel treebank for Dutch and English, allowing users to query the treebank using either an XPath expression or an example sentence in order to look for similar constructions (Currently, this is limited to the years 2000 and 2001. After we speed up the process using [30,31], we expect to expand this to the entire Europarl corpus, version 7.). The treebank contains automatic alignments between the nodes. By combining example-based query functionality with node alignments, we limit the need for users to be familiar with the query language and the structure of the trees in the source and target language, thus facilitating the use of parallel corpora for comparative linguistics and translation studies. Poly-GrETEL is part of CLARIN (Common Language Resources for Research Infrastructure, http://www.clarin.eu).

In future versions, we expect to allow users to upload their TM files in TMX format, which would enable them to look up how certain syntactic constructions are translated in the available TM. Poly-GrETEL can hence be seen as an initial version of a *syntactic concordancer*.

3. Quality Estimation of Computer-Aided Translation

Quality Estimation (QE) is defined as the task of providing a quality indicator for machine-translated text without relying on reference translations. The aim of QE is to predict a quality score at sentence and/or document level or more fine-grained error labels at word level that indicate the need for post-editing. The general approach to QE consists of feature engineering, which is the task of finding informative predictors (or features) of MT quality and applying various Machine Learning (ML) algorithms to build prediction models, which associate features with quality labels.

Today, despite their widespread adoption, ML models of QE remain mostly *black boxes*, where no explanation for the predicted quality is provided [32–34]. In order to gain wide-spread acceptance, besides building more accurate systems, one of the main challenges of QE can be considered to build *white box* systems whose predictions can be justified. Based on the definition of the post-editing task, one way of doing this would be to take a two-step approach, by detecting different types of MT errors in the first step, which are then used in a second step to estimate a global score at sentence level. Such systems would not only be beneficial for MT developers and end users to make a meaningful analysis about the translation errors a certain MT system makes but they can also yield higher productivity gains in CAT workflows that utilise MT and can improve the acceptability of MT by post-editors, by filtering out the sentences with the more challenging error types and by highlighting errors. In the SCATE project, we use automatic error detection as a basis to two-step, informative quality estimation systems for MT, which are able to justify the reasons for estimated quality.

In Section 3.1, we first describe a new taxonomy and annotated data set of MT errors. Section 3.2 describes our approach to building informative quality estimation systems.

3.1. Taxonomy and Annotated Data Set of Machine Translation Errors

Despite the link between MT errors and post-editing effort, most QE systems predict overall post-editing effort, without making a distinction between error types. Automatic error detection is essential to build informative QE systems that are specialised in localizing different types of errors. To this end, in Figure 5, we present the SCATE MT error taxonomy, a fine-grained, hierarchical taxonomy, in which errors are classified according to the type of information that is needed to detect them. We refer to any error that can be detected in the target text alone as a *fluency error*. Fluency errors are concerned with the well-formedness of the target language, regardless of the content and meaning transfer from the source language. There are five main error subcategories under fluency errors: *grammar, lexicon, orthography, multiple errors* and *other fluency errors*. *Accuracy errors*, on the other hand, are concerned with the extent to which the source content and the meaning is represented in the target text and can only be detected when both source and target sentences are analyzed together. Accuracy errors are split into the following main subcategories: *addition, omission, untranslated, Do-Not-Translate (DNT), mistranslation, mechanical, bilingual terminology, source errors* and *other accuracy errors*.

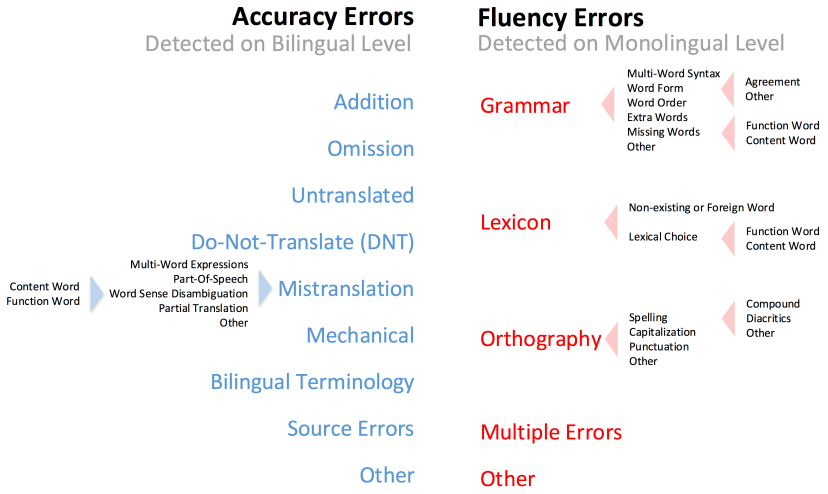


Figure 5. The SCATE MT error taxonomy.

Certain similarities can be observed between some of the accuracy and fluency error categories in the error taxonomy, such as *extra words* versus *addition*, *missing words* versus *omissions* or *orthography–capitalisation* versus *mechanical–capitalisation*. As the main distinction between accuracy and fluency errors in the taxonomy is based on the type of information that is needed to be able to detect them, accuracy errors do not necessarily imply fluency errors, or vice versa, for that matter [35].

Using the SCATE MT error taxonomy, for the English-Dutch language pair, we built corpora of MT errors consisting of output from three MT systems that are based on different MT paradigms: SMT, Rule-Based MT (RBMT) and NMT. In these corpora of MT errors, we obtained error annotations provided by multiple annotators, yielding high Inter-Annotator Agreement (IAA). We used Google Translate (2014) as SMT system, Systran Enterprise Edition, version 7.5 as RBMT system and Google Translate (2017) as NMT system to obtain MT output for all source sentences. The source sentences in the corpus of SMT errors are extracted from the Dutch Parallel Corpus [36] and consist of an equal number of sentences from three different text types: *external communication, non-fiction literature* and *journalistic texts* (698 sentences in total). Furthermore, we extended the corpus of SMT errors (2963 sentences in total) to analyze the relationship between MT error types and post-editing effort and to build automatic error detection systems, which are further explained in the next section. Further

information on the MT error taxonomy, the corpora of MT errors and the IAA analysis can be found in Reference [35].

3.2. Quality Estimation

We first discuss the predictive power of SMT errors in Section 3.2.1, before discussing automatic error detection in Section 3.2.2 and informative quality estimation in Section 3.2.3.

3.2.1. The Predictive Power of MT Errors on Temporal Post-Editing Effort

From a post-editor's perspective, MT quality can be considered of the highest level when the MT system makes no serious translation errors, in other words when the effort required to post-edit is minimal. Despite the obvious relationship between the cognitive effort involved in post-editing and the translation errors made by the MT system, the impact and the predictive power of different types of MT errors on post-editing effort are yet to be fully understood.

With the hypothesis that the different error types an MT system makes can explain the cognitive effort involved in correcting them, we investigate whether ML techniques can be used to estimate Post-Editing Time (PET), an indirect measure of cognitive effort, by using gold-standard MT errors as features. We analyzed the SCATE corpus of SMT errors in combination with post-edits obtained for each MT output by two post-editors, both native speakers of Dutch and Master's students in translation studies and the average PET calculated per sentence.

By using the gold-standard error annotations, we showed that PET can be estimated with high accuracy, provided that the types of errors in the MT output are known. We obtained these results by applying different ML techniques to the largest data set ever used in similar studies (The SCATE corpus of MT errors is available at https://github.com/ardate/SCATE.) [37].

While these findings suggest that building two-step, informative quality estimation systems is possible in theory, accurate detection of all MT error types can be considered to be a challenging task, considering the different linguistic properties they represent. On the task of predicting PET, we applied various feature selection methods not only to seek a minimal subset of MT error types without reducing QE performance but also to reveal the predictive power of different error types on PET. Our results show that high QE performance on SMT output can be achieved by using only eight error types (compared to all 33 error types) in the SCATE error taxonomy, corresponding to 31% of all gold-standard error annotations in the corpus. We observed the *Accuracy–Mistranslation* and *Fluency–Grammar* errors as two main error categories, whose sub-categories correspond to error types with high predictive power. Our findings suggest that we do not need to detect all error types to estimate PET successfully and error detection systems that focus only on error types with high predictive power on PET can lead to high quality sentence-level QE performances. For the details of our findings, we refer to Reference [37].

3.2.2. Automatic Error Detection

Considering the informativeness of the different types of MT errors on PET, we propose novel RNN architectures for word-level automatic error detection for *Fluency* and *Accuracy* errors as bases to building informative QE systems for predicting PET on sentence level.

In order to train Neural Networks (NNs) on the task of detecting fluency errors, which are concerned with the well-formedness of the target text alone, we propose a new word representation method, in which we transform each word in a given MT output into a feature vector using multi-hot encoding, which consists of three types of information: Part-of-Speech (PoS), morphology and dependency relation, which we extract by using the Alpino dependency parser for the Dutch language [38]. In each word vector, all elements are assigned the value of 0, except the elements representing the linguistic features of each word, which are assigned 1. Unlike word embeddings, the morpho-syntactic representation strips out semantic features from words. One difficulty of using dependency parsing on MT output is that the generated parse trees can be unreliable when the MT

output itself contains errors. On the other hand, multiple studies demonstrated that parse trees obtained on MT output nevertheless provide useful information in terms of MT quality [39,40].

Figure 6 shows an example source sentence (EN), its machine-translated version (NL) and the morpho-syntactic representation for the word *zijn (are)*. The MT output in this figure contains a *Fluency–Grammar* error in the form of subject-verb agreement in number between the words *zijn (are)* (plural) and *kans (chance)* (singular).

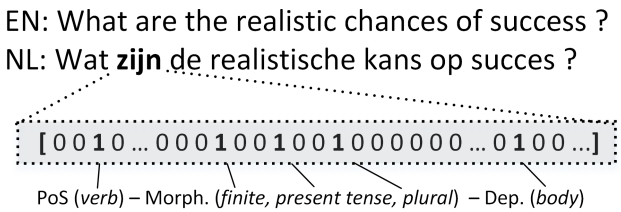


Figure 6. Binary vector for *zijn (are)* consisting of 1s for its PoS, morphology and dependency features and 0s for the remaining items in the vocabulary.

Besides surface context windows (*n*-grams), we utilised syntactic context windows for each given target word, which we extracted from the dependency parse tree for each given MT output. Syntactic n-grams enable us to capture long-distance dependencies in MT output, which can be considered as an important piece of information especially for detecting *Fluency–Grammar* errors. Combining morpho-syntactic features with surface and syntactic n-grams, we propose an RNN architecture, which is illustrated in Figure 7.

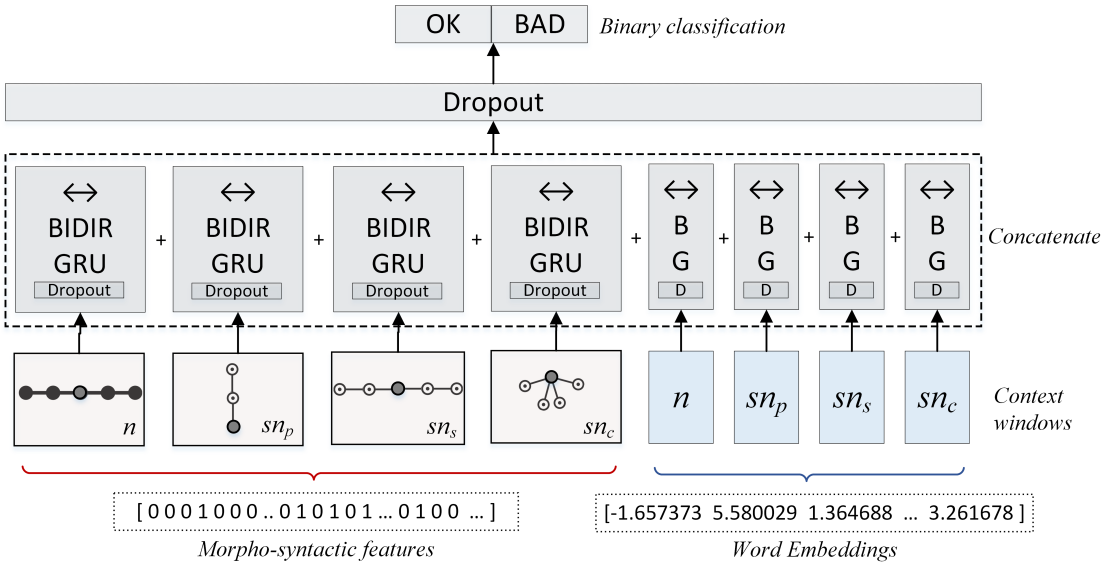


Figure 7. The proposed neural network architecture for detecting fluency errors. While n represents a surface n-gram, sn_p, sn_s and sn_c represent syntactic n-grams obtained around the target word by considering its parents, siblings and children as context in a given dependency tree.

In the proposed RNN architecture, we provide morpho-syntactic feature vectors and word-embedding vectors of a target word in the form of surface and syntactic n-grams into eight parallel Gated Recurrent Unit (GRU) layers, whose output is concatenated before they are connected to the output layer. This network, as a result, predicts if a given word contributes to a fluency error or not as a binary classification task. The combination of morpho-syntactic features and word-embeddings achieved better QE performance on the task of detecting all fluency errors, than using either type of information as input in isolation. Moreover, on the task of detecting *Fluency–Grammar* errors in

SMT output, we achieved a marked improvement in performance by using accurate morpho-syntactic features over word-embeddings [41]. An approach to grammatical error detection as in [42], but adapted to MT output instead of learners output, was not yet available at the time of our research but could be tried in future research for detecting different types of fluency errors in MT output.

To detect accuracy errors, we modify the proposed RNN architecture and instead of using morpho-syntactic features of the target text, we use word-embedding information obtained on the source and target texts as input. Our approach additionally incorporates automatic word alignment techniques to extract relevant information from the source text. We show that the proposed method achieves the best results compared to other NN configurations that utilise morpho-syntactic features as additional input. For the details of our experiments on automatic error detection, we refer to Reference [41].

3.2.3. Informative Quality Estimation

Automatic error detection of fine-grained error categories remains a highly challenging task. However, the predictions obtained from the error detection systems on more coarse-grained error categories, such as dedicated systems for all accuracy and all fluency errors perform relatively well and serve as valuable features for building informative QE systems to predict PET. Furthermore, additional experiments show that the predictive power of such informative sentence level QE systems could be maximised with additional sentence-level features obtained on a given source/MT output pair, yielding 96% of the Pearson's correlation score of the upper boundary we observed on this task by using gold-standard error annotations as features [41].

One of the aims for building informative QE systems is to inform the users about the reasons for the estimated quality. Figure 8 shows how informative QE is presented to the user on the SCATE platform. Words that are underlined in red are the words that correspond to fluency errors, which are detected automatically. The score to the right of the MT output (0.56) corresponds to the predicted sentence-level quality, which is calculated as $1 - TER$. The model that predicts the TER score is a Support Vector Machine (SVM) model, which has been trained with 17 sentence-level features obtained by the QuEst++ toolkit [33,43]; such as the number of tokens in source and target texts and language model probability scores of source and target texts and the number of tokens with fluency errors predicted by the neural model described above, as an additional feature. As illustrated in this figure, the SCATE platform not only highlights the type and location of errors in a given MT output but also uses this information to predict its sentence-level quality.

Figure 8. Quality estimation output in the SCATE user interface.

Even though predicting the exact location of MT errors remains a challenging task, we observe that the proposed systems approximate the location of errors with greater success. Moreover, despite the given challenges, our findings confirm that using automatic error detection systems as a basis for sentence-level QE is a promising approach to build informative QE systems. We demonstrate that the proposed methods deliver QE systems that perform well on estimating temporal post-editing effort, while providing meaningful predictions about the type and location of the translations made by a given MT system. For further details, see Reference [41].

4. Terminology Extraction

We first describe our observations of translator's methods for acquiring terminology (Section 4.1), before we describe our approach to automatic term extraction from comparable text (Section 4.2).

4.1. Studying Translator's Methods of Acquiring Domain-Specific Terminology

To identify translators' terminology strategies of acquiring new domain knowledge, we launched an online questionnaire and visited language professionals at their workplaces. The questionnaire contained a total of 46 questions out of which 13 concerned demographics and professional experience, 9 concerned the translation work environment and 9 concerned terminology activities. The questionnaire was answered by 187 language professionals worldwide, out of which more than 70% were freelance translators and the rest were in-house translators/revisers, terminologists, interpreters, post-editors and project managers. The questionnaire was online between December 2014 and February 2015.

In the field, we observed 13 translators and 3 terminologists in their authentic professional work environment (freelance, commercial and institutional settings) by applying the Contextual Inquiry [44] and Think Aloud Protocol (TAP) [45] research methods. The workplace visits took place in Belgium, the Netherlands and Luxembourg and were spread over a period of 6 months between November 2014 and June 2015. For more details we refer to Reference [46].

The study reveals information about translators/terminologists' terminology acquisition and management practices, web search behaviour and usage of online linguistic resources to solve terminological problems. Out of 187 survey respondents, about 139 indicated performing terminology activities. About 88% collected terms manually, while 22% used semi-automatic term extraction programs. More than half (about 52%) stored their terms in their CAT termbase, while 43% in a spreadsheet. The rest preferred a text processor (27%) and standalone translation management systems (15%). More than half stored only the language equivalents in their termbases. As for term research activities, the online resources were most exploited, followed by personal resources and client's resources. Finally, the survey helped us identify needs and shortcomings of the terminology management component integrated in CAT tools, related to the integration with online databases and exchange of terminological data. For more information see Reference [47].

During the contextual inquiries at translators' workplaces we noticed the following types of terminology problems that occurred during translation:

(1) Related to specialised terminology: the translator does not know the meaning of the source term; the translator does understand the source term but does not know how to translate it in the target language; the translator does not know which target language equivalent to select from several translation alternatives coming from a large database.
(2) Related to general language.
(3) Related to the translation of named entities, acronyms, ambiguity, low quality of the source text and punctuation.

To find a solution, translators used various tools, search and retrieval strategies both from local and online resources. We summarise the main findings below:

Both the survey and the field observations revealed that translators rely more on their TMs than on termbases to retrieve translation solutions. When no matches are found, the translator can perform a bilingual concordance search, in which the source term is highlighted and a target sentence is shown as such, with no highlight of the translation equivalents. The translators has to copy/paste the preferred translation from the concordance result window into the target sentence. We saw that the concordance feature was the second preferred CAT tool feature, after the TM match retrieval functionality. The over-reliance on TMs is signalled and discussed in early studies as well, for example, [48,49]. While parallel corpora can be very useful for analysing translation equivalents in their context, Reference [50] warns that they can have a major drawback in the fact that *"they require*

the existence of a translation history" and they are not *"faithful to linguistic uses in the target language."* She further emphasises that comparable corpora (collections of original texts in two or more languages assembled on the basis of similarity) can also be a good alternative to acquire specialised knowledge and terminology for under-resourced languages and emerging fields. Despite its proven usefulness [51] the SCATE survey shows that comparable corpora are hardly exploited for terminology and knowledge acquisition, the only resource mentioned being Wikipedia. SketchEngine that contains the TenTen Corpus Family (https://www.sketchengine.eu/documentation/tenten-corpora/) was mentioned only by one participant out of 139 who indicated performing terminology activities.

Besides the concordance feature, the translator can also use the term extraction feature incorporated in their CAT tool to quickly retrieve term candidates from their TMs and reference corpora, validate the term and add them to their termbase for future use. Most tools incorporate a monolingual term extraction component, whereas our research shows that there is also a need for bilingual and multilingual automatic term extractors. In addition, the survey showed that only 19 of a total of 187 used the term extraction feature in their CAT tool. Some reasons for the low usage, revealed during the observations: the users did not know how to configure the extraction parameters and the validation of the term candidates was time-consuming due to the amount of noise.

Besides TM, the institutional translators also had access to a custom MT system that they could used to retrieve possible translation suggestions for terms, phrases or entire segments when there were no matches coming from the TMs. None of the commercial translators we observed used MT via the plugins integrated in their CAT tools.

Another method to search for terminological information or translation equivalents is to look up terms and phrases in external databases directly from the CAT tool's translation editing interface. Although most translation environments offer look-up functionality in external terminology databases (e.g., IATE, UnTerm, EuroTerm) and parallel corpora (e.g., MyMemory), our research shows that the integration with CAT tools is not optimal. Both commercial and institutional translators indicated that more advanced filtering techniques are required in order to query the IATE database directly from the CAT tool's interface. In addition, online databases are not always up to date, may contain outdated references or may reflect the terminology used by a specific organisation. Nevertheless, things have changed since the study finished. The IATE team has launched a new version of IATE that is user-friendlier. Recently, in a JIAMCATT local meeting it was announced that SDL was developing a plugin for IATE to allow translators search the database directly from the interface of SDL Trados Studio. JIAMCATT is the International Annual Meeting on Computer-Assisted Translation and Terminology. JIAMCATT membership includes most international organizations, as well as various national institutions and academic bodies, active in the field of terminology and translation.

When the local resources did not return any useful results for terminology and translation problems, the translators switched to the Web to look for a solution by consulting various websites, online dictionaries and platforms. Similarly to the results of the TTC survey [52], both our survey and field research revealed that online resources are the most popular linguistic resource for researching terminology. Figure 9 shows the most used resources from each category.

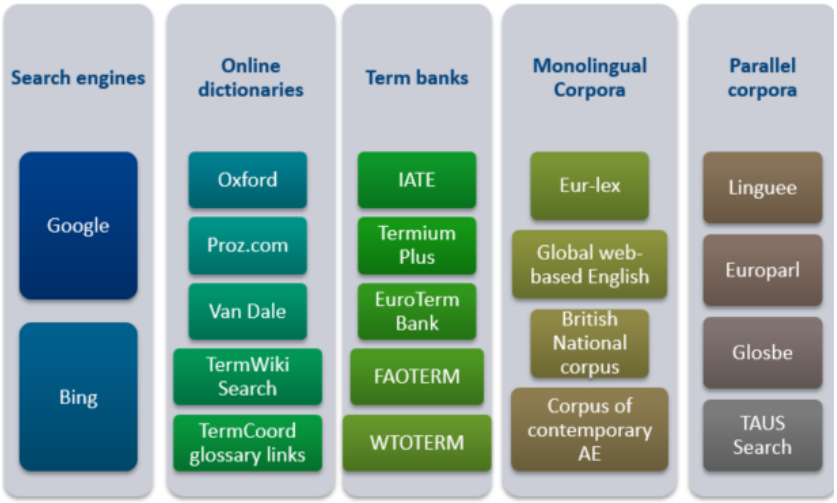


Figure 9. Most used online terminological resources.

During the observations, we noticed lots of back and forth switching between several types of online resources before taking a final decision.The web research path was often decided by the number of hits Google gave with the web searches resulting in desktop clutter as the user did not know how to manage the search results. For this purpose, one of our subjects developed a strategy not to keep more than three tabs open on his desktop. He also used the Ditto clipboard manager to record his searches, which saved time. Though the Google search engine was often used, out of the 16 translators we observed only 2 used some advanced search operators in Google. When a translation solution was found, it was copied/pasted in the translation grid and confirmed in the Translation Memory. Useful websites were added to the Favourites toolbar. At the European Parliament, for example, the web links were usually centralised and shared via the internal portals of the terminology and translation units. Out of 16 observations, we noticed only one instance when the translator actually stored the information about the researched term in their term base. These findings correlate with the results of the survey that revealed the reasons why translators do not perform proper terminology management: lack of knowledge about terminology management, someone else's responsibility, no added-value, time consuming, termbases are complex.

Another method of acquiring domain-specific terminology is manual compilation of small thematic corpora with materials collected from the Web, which can be followed by manual term extraction of a list of term candidates, validation and import of the final terms into the terminology database. The source term entries are then researched and completed with target-language equivalents. This practice was observed during the observations of the 3 institutional staff terminologists. While the manual collection of the corpora and extraction of terms are reliable methods of harvesting terminology, the participants indicated that it was time-consuming. Ideally, the users should be able to collect corpora automatically and query directly from their translation environment tool. Although there are standalone corpus compilation and query tools, such as Sketch Engine, BootCat, AntCont, the SCATE survey shows that they are hardly known and used by translators. This might be due to the fact that such tools are not supported in their CAT tools. In 2016, Sketch Engine developed a plugin for SDL Trados Studio to enable translators and terminologists to perform searches in their large collections of corpora (e.g., Eur-Lex) directly from the Translation Editing interface. The pilot showed that the plugin was hardly used by translators and, therefore, further development stopped.

Overall, the field research confirms the findings of previous studies that terminology management is mainly done on an *ad hoc* basis due to time pressure, lack of resources, limited knowledge of how to manage terminology properly and lack of immediate financial compensation. A systematic approach was observed only at the European Institutions and large commercial organizations which had dedicated terminologists in-house. Translators seem to rely heavily on their specialised TMs rather

than on termbases and/or comparable corpora. Semi-automatic term extraction, though an integrated component in the commercial CAT tools, has not yet become a standard practice in the preparation stage of a translation project. The Web represents a rich resource for knowledge and terminology acquisition but very few adopted the automatic tools for corpora compilation and query. Finally, more efficient web search strategies are needed in order to avoid desktop clutter and save and store the relevant information in an efficient way. The findings have implications for translators educators and software developers alike.

One way of optimizing the exploitation of external linguistic resources for the purpose of terminology acquisition is a seamless integration of more sophisticated terminology extraction methods from comparable corpora.

4.2. Terminology Extraction from Comparable Text

We experimented with three types of comparable corpora. The first type are corpora compiled from Wikipedia articles, which are a valuable resource for compiling comparable corpora. Wikipedia articles have the benefit that they are annotated with the categories they belong to as well as with *interwiki* links, which link an article to its counterparts in other languages. Both types of annotations allow easy compilation of a comparable corpus that is both domain-specific (using the category labels) and strongly comparable across languages (using the interwiki links). For our experiments, we constructed an English-Dutch comparable corpus in the medical domain, containing about 1000 document pairs. Datasets with aligned Wikipedia articles can be found online for many language pairs on the website of linguatools: https://linguatools.org/tools/corpora/wikipedia-comparable-corpora/.

The second type are corpora compiled from Reuters news articles. News articles are another resource to create comparable corpora. We experimented with the Reuters news dataset (http://trec.nist.gov/data/reuters/reuters.html), a multilingual collection of news articles published within the same time span. From this collection, we created a weakly-comparable corpus by comparing the topic labels (e.g., *global, economy* etc.) that are annotated on the Reuters documents, for example: when an English document and a Spanish document are both annotated with the same global label they are considered to have comparable content and are added as a document pair to the comparable corpus. We analysed the resulting dataset with multilingual probabilistic topic models: *Bilingual Latent Dirichlet Allocation (BiLDA)* [53] and *Comparable Bilingual Latent Dirichlet Allocation (C-BiLDA)* [54]. We found that, although the C-BiLDA model could uncover some interesting cross-lingual topics (clusters of related words), the dataset was not well-suited for inducing translations as the domain was too broad and the comparability across languages too low. We therefore conclude that to construct comparable corpora from news articles merely relying on high-level topic labels is insufficient. Other clues like named entities (persons, locations) and publication dates should be taken into account.

The third type are existing comparable corpora. Several automatically crawled and cleaned comparable corpora have been made freely available online in the context of the TTC project (http://www.lina.univ-nantes.fr/?Linguistic-Resources-from-the.html). These are all specialised corpora in specific domains, such as wind energy and mobile technology. They are available in different formats and in seven languages: English, French, German, Spanish, Russian, Latvian and Chinese. These characteristics make the corpora especially suited for experiments with automatic term extraction from comparable corpora. An additional advantage is that there are also (very) limited, manually validated reference term lists available for the evaluation of monolingual automatic term extraction. A final advantage is that the corpora have been used in previous experiments with automatic term extraction from comparable corpora, so any new results can easily be benchmarked against the state of the art.

We split cross-lingual terminology extraction into two subproblems: (1) term extraction, the identification of which words and phrases are (in-domain) terms; and (2) term linking, where the aim is to link terms to their correct translation. We focus mainly on term linking. We investigatie word-level methods for bilingual lexicon induction (BLI), the task of finding translations for words

and phrases from non-parallel texts; we propose a novel BLI model that integrates character-level and word-level representations; and we implement a hybrid compound splitter for Dutch that combines corpus frequency information with linguistic knowledge.

4.2.1. Comparison of Weakly-Supervised Word-Level BLI Models

During the course of the project, we saw the rise of word embeddings in natural language processing. These vector representations have shown to encode useful syntactic and semantic properties of words and have also been used to build cross-lingual spaces where translations are mapped to similar representations. Most techniques that build such cross-lingual representations require parallel corpora or bilingual dictionaries, however. We study approaches that can learn cross-lingual representations without the need for an initial seed dictionary.

In particular, we compare two bilingual topic models, BiLDA and C-BiLDA, with a bilingual extension of the continuous skip-gram model called *Bilingual Word Embedding Skip Gram (BWESG)* [55]. All three models learn bilingual word representations from subject-aligned document pairs only. Multilingual topic modeling has shown to be a robust framework for learning bilingual representations from such non-parallel data: BiLDA has been successfully applied to BLI [56] and C-BiLDA is a more recent extension to BiLDA that learns higher-quality representations when the aligned document pairs exhibit a lower degree of parallelism [54]. BWESG is a simple but effective extension to continuous skip-gram. It merges each aligned-document pair in a single bilingual document and then runs monolingual skip-gram with negative sampling [57] on the resulting document collection. To evaluate the models, we use a corpus of subject-aligned Wikipedia documents (English-Dutch) in the medical domain. From the English side of the corpus we selected 500 words, which were translated into Dutch to form the ground truth. We found that BWESG yields the best performance which indicates that also in a weakly-supervised settings, without parallel data, word embeddings are important BLI features.

4.2.2. Combining Word-Level and Character-Level Representations

From our word-level experiments, we observe that for our dataset (consisting of Wikipedia articles in the medical domain) morphology is an important clue for identifying translations. Most recent work in BLI focuses solely on word-level features, however. For this reason, we design a model that seamlessly integrates word-level features (e.g., continuous skip-gram embeddings) and character-level features. Most related work in bilingual lexicon induction manually defines a cross-lingual similarity metric between word feature vectors. For instance, many methods use cosine distance to measure the similarity between embeddings. It is not trivial to define a similarity metric that incorporates both word- and character-level information, however. Therefore, we frame bilingual lexicon induction as a classification problem. We train a binary classifier that predicts whether two given words are each other's translation. The classifier's parameters are learned from a seed lexicon of known translations.

We identify two key advantages of a classification framework for BLI. Firstly, it does not rely on an *ad hoc* combination of features but learns the patterns over different features from the bilingual seed lexicon. Secondly, the classification framework enables learning useful character-level features from the seed lexicon. This in contrast to using handcrafted features like normalised edit distance. In our model, we obtain a character-level representation by feeding the concatenation of source and target characters to an LSTM network (see Figure 10). As word-level representations, we used continuous skip-gram word embeddings. The concatenation of word and character features serves as the input to a feed-forward neural network that outputs a score between 0 and 1. The higher the score, the more confident the model is that the two given words are each other's translations.

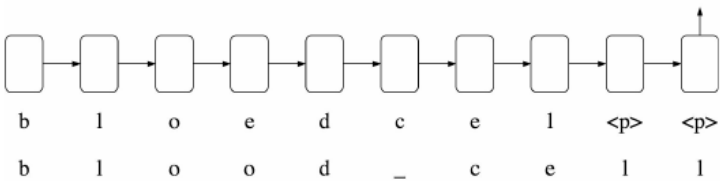


Figure 10. Character-level representation in an LSTM framework.

Our experiments show that the LSTM representation outperforms handcrafted morphology features like normalised edit distance. Furthermore, the model that combines character-level information and word-level information outperforms other baselines (including BWESG, the strongest word-level model) by a margin. For more details, see Reference [58].

In follow-up work [59], we verify that we can extend the BLI system, which could only find translations for single words, to deal with phrases. Specifically, we find that, after extracting phrases using a simple data-driven heuristic, we can treat phrases as if they were a single word: To learn character-level representations, we treat whitespace as any other character, and to learn word-level representations, phrases are tokenised as a single token.

4.2.3. Datasets and Gold Standards for Future Research

Finding comparable corpora for bilingual term extraction is not easy. Wikipedia is a useful resource, but for very specialised subjects or smaller languages, coverage is not always optimal. Moreover, while the strong comparability per document is useful, it is rare in other resources. Compiling comparable corpora *ad hoc*, such as the one from Reuters new articles, is convenient but still requires identification of the terminology. Finally, a few comparable corpora are available with manual term annotations, such as the one used from the TTC project. However, these are very rare and often contain only monolingual annotations or a very limited list of cross-lingual links. This lack of good resources means that evaluation can be challenging and it is an important obstacle for the development of supervised ML approaches for both monolingual and multilingual term extraction from comparable corpora.

To address this, we started building a dataset for term extraction, which can be used both as a gold standard and as training data for a supervised ML approach. To ensure re-usability of the data, we collect corpora in three different languages (English, French and Dutch) and four domains (corruption, dressage, heart failure and wind energy). These corpora are partly based on previous research (e.g., the wind energy corpus uses the French and English parts of the TTC corpus). In each corpus, around 50,000 tokens are manually annotated, using an annotation scheme with three different term labels and elaborate guidelines. The guidelines, including information about the term labels, are freely available online (http://hdl.handle.net/1854/LU-8503113). This results in a total of over 100,000 manual annotations in all corpora. We are currently experimenting with an ML approach to term extraction based on these data.

While this is already a useful resource, as explained in the previous sections, multilingual term extraction from comparable corpora involves two tasks: identifying terms and linking equivalent terms across languages. Since the described data only addresses the former, more annotation work was required to provide data for the latter. Therefore, the trilingual corpus about heart failure was selected and both inter- and intralingual links between terms were manually identified: equivalents across languages, synonyms, abbreviations, alternative spellings, hypernyms, hyponyms and so forth. In total, 7385 unique terms and named entities in three languages were annotated this way. This dataset is particularly suited as a gold standard for multilingual term extraction from comparable corpora for two reasons. First, the inclusion of information about related terms means that a more nuanced evaluation can be made in cases when the automatically suggested target language term is not exactly

an equivalent of the source term but is still strongly related. Second, the fact that all terms have been annotated in this corpus means that the origin of wrongly suggested equivalents can be traced: either the system could not find the equivalent or it was not present in the corpus. After all, since comparable corpora are not aligned, it is not unusual for a term to exist in one language of the corpora and not the other.

While these datasets could not yet be used to evaluate the systems presented in the previous section, they have already proven to be valuable sources of information about both terminology and comparable corpora [60]. For instance, the lack a of restriction about length or part-of-speech of the terms revealed that, as expected, nouns and noun phrases are most common but that, somewhat surprisingly, other part-of-speech patterns were often identified as well, for example, adjectives and even verbs. Single-word and two-word terms appeared most often but longer terms, up to around five tokens, were no exceptions. Ongoing research will have to confirm the further use of the data for the development of new tools. The dataset will be made available through a shared task on supervised machine learning approaches for automatic term extraction in 2020.

5. Speech Recognition

In the context of post-editing, using speech instead of typing can speed up the work of the translator. The accuracy of automatic speech recognition (ASR) can be improved by making use of the extra information present in the translation model (Section 5.1) and by adapting the language model to the current domain or topic (Section 5.2). Additionally, we explore the challenging task of speech translation in Section 5.3.

5.1. Adaptation of the Speech Recognition Language Model by Machine Translation

The aim of this research is to employ improved language models (LMs) and achieve higher recognition accuracy for spoken translations. We investigate two ways of improving the LMs: (1) using word translations to cluster similar words, which improves the reliability of word frequency statistics; (2) using the source language text and MT probabilities to steer the recognition in the right direction.

The first approach assumes that two words are similar, both semantically and syntactically, if they share the same translation in multiple languages. Similar words can then be clustered, which enables context sharing within each cluster and hence more reliable statistics for n-gram LMs containing these words. By filtering out translation errors based on part-of-speech, context and morphology, we are able to derive meaningful synonym clusters but this does not result in improved recognition, mostly due to context insensitivity, that is, words may be synonymous in certain contexts but not in others.

The second approach investigates how to improve speech recognition, based on the source language text and MT probabilities. Research in the past largely focused on rescoring either ASR n-best lists or word lattices, using the MT probabilities of the source language text. This has the disadvantage that it requires two steps, which slows down recognition and requires intermediate storage. Moreover, such multi-pass approaches are often inferior to integrated approaches because information that is lost during the first step can never be recovered in the second step. Therefore we focus on integrating the source language text and MT probabilities into the LM directly. By weighing the n-gram probabilities with the translation probabilities of the source language text, a new LM can be created for each sentence/paragraph which can directly be used by an ASR decoder. This implementation allows to reduce recognition errors by ca. 5% absolute and 20% relative on spoken Dutch translations from English, while having little to no negative effect on recognition time. Moreover, compared to an existing model [61], our model takes up only 2.8% of disk space compared to a normalized model and dramatically reduces the execution time. More information can be found in Reference [62].

Although the above implementation drastically improves the efficiency of MT-based LM adaptation, it assumes that translation consists solely of one-to-one alignments, that is, each word in the source language text can only correspond to one word in the target language text. This is a strong assumption that does not hold in reality: every language has its own way of verbalizing concepts with

some using a single word and others using multiple words for the same concept. In MT this issue is addressed by phrase-based translation models.

We integrate phrase-based models into our implementation, without compromising the recognition time. We also extend the recognizer with named entity models. These models attempt to improve recognition for proper nouns by estimating their pronunciation and language behavior. We exploit the fact that many named entities remain unchanged during English-to-Dutch translation implying that we can make reliable estimates for relevant named entities based on the source language text. Experiments show that the combination of phrase-based translation models and named entity models further reduces the recognition error to ca. 6.5% absolute and 25% relative on the same spoken Dutch translations from English. Moreover, the extensions come with the same efficiency benefits as the word-based model which allows their use in a real-time CAT environment. To our knowledge this is the first MT-based language model adaptation technique using a phrase-based translation model. More information can be found in Reference [63].

5.2. Automatic Domain Adaptation

We also investigate the effect of automatic domain adaptation for speech recognition. We study both cross-domain adaptation and within-domain adaptation: the first approach adapts a model trained on a specific domain to other domains, while the second approach adapts to the current topic of the text.

For cross-domain adaptation, we chose to create a new data set. Previous recognition experiments were always performed on spoken translations of literature for which the domain is not always very confined. For this task we instead chose to work with 14 documentaries provided by VRT, the Flemish public broadcaster (https://www.vrt.be/en/), all of which have a specific domain, that is, mostly fauna and flora. For these data we have the following parallel data streams: (1) audio in English (original), (2) script in English (original), (3) audio in Dutch (voice over), (4) script in Dutch (as input for audio in Dutch), and (5) subtitles for the deaf in Dutch.

The audio is converted to the correct format and background noise is filtered out as much as possible. Subtitles are normalized to generate a ground truth transcription which is aligned with the audio to produce the necessary timing information. Baseline experiments with models that do not employ any domain adaptation yield acceptable word error rates, ranging from 9% to 33%.

In a first attempt we investigate two methods of exploiting domain knowledge: (1) fully automatic terminology extraction; (2) user-guided terminology extraction. The first method uses BiLDA to automatically extract relevant Dutch terminology based on the English translation. In the second approach, we develop semi-automatic methods in which the user/translator enters a Dutch query/description of the translation task. This query is then used to retrieve relevant terminology, using one of the following methods:

1. Word-to-word similarity based on a Latent Semantic Analysis (LSA) model [64]
2. Word-to-word similarity based on a continuous skip-gram model [65]
3. Document-to-document similarity based on LSA, followed by extraction of the most relevant words from the best matching document.

These methods are incorporated into the SCALE toolkit, which is described in Reference [66]. Each of the investigated methods is first evaluated on text: for each documentary, the extracted terminology is compared to out-of-vocabulary (OOV) words: the most promising method is the one that is able to retrieve the most OOV words. In a next step, this terminology is added to the pronunciation lexicon and language model of the speech recognizer and the word error rate of the domain-adapted speech recognizer is measured. None of the proposed methods is able to extract enough relevant terminology consistently. Hence, we focus on other adaptation techniques. Moreover, we move from n-gram language models to the state-of-the-art RNNS LMs, more specifically long short-term memory (LSTM) [67].

A new topic of investigation for cross-domain adaptation is improving the modelling of OOV words. These are words that are not part of the speech recognizer's vocabulary and therefore cannot be recognized. OOV words are a known issue in cross-domain settings as the change of domain often introduces many domain-specific words. We work on combining word and character information in the LM, rather than only using word information. By using character information, the LM should be better able to see similarities between formally/morphologically similar words. This improves the quality of the model and reduces the number of parameters to train, because the vocabulary size when using characters is very small compared to words. Moreover, the model is better able to predict words following out-of-vocabulary words, because it can make use of the characters in the OOV word. Not only does our model improve on the existing language model, it also reduces its size. These findings are reported in Reference [68]. The code for both baseline LSTM LMs and the word-character LSTM LMs is described in Reference [69].

With respect to *within-domain adaptation*, we investigate three approaches. The first approach exploits the history, by combining the baseline LM with a continuous bag-of-words (CBOW) [65] representation of the previous words. We investigate how word embeddings are optimally combined into a history representation (e.g., mean, weighted mean or filtered mean) and how the resulting CBOW should be combined with the baseline RNN (at the input layer or the output layer of the RNN). Unfortunately, the improvements for small LMs did not extrapolate to larger LMs.

The second approach is similar to the CBOW model in that it builds a continuous representation of the history. However, rather than using word embeddings, the model uses an RNN to learn the weights of a fixed set of topics which were pretrained using Latent Dirichlet Allocation [70]. Using the weighted sum of these topics, the model should be able to predict topical words which can be combined with the baseline RNN LM. The results for this model are similar to the previous one: only improvements for smaller LMs are found. These findings are reported in Reference [71].

The third model is a neural cache LM [72]. A cache model [73] is inspired by the fact that people tend to talk about the same topic for a while, such that words that have been used before in a conversation have a higher probability of being used again. In a neural cache LM, the previous words and their hidden representations are stored in a cache. A probability for the next word is calculated based on the similarity between the hidden representation of the current word and the representations stored in the cache. That cache probability is combined with the standard LM probability. We extend the neural cache model by starting from the intuition that a cache makes more sense for content words (e.g., *bilingual, backhand*) than for function words (e.g., *the, on*). We observe perplexity improvements by using the *information weight* of a word, which is large for content words and small for function words. We use the information weights to combine the cache and LM probability and to select which words should be added to the cache. Additionally, we compare the regular cache [73] and the neural cache [72] for speech recognition and we find that, contrary to the results for perplexity, the regular cache performs better. The results of this research can be found in Reference [74].

5.3. Translation of Spoken Data

In this section we focus on punctuation and segmentation insertion, since this is an important task for speech translation. Most ASR systems generate an output stream of words, which does not contain punctuation nor segmentation, apart from some form of acoustic segmentation which splits a transcript into so called utterances [75]. As these utterances may be very long and can contain several sentences, they are very hard to translate using MT engines. We tackle this issue in two steps: firstly, punctuation prediction and secondly, segmentation prediction.

Most MT engines are trained on data that contain punctuation marks. As the output of a speech recognition system usually contains no punctuation information, a solution needs to be found for this mismatch. We investigate several approaches.

LM/LSTM based approaches—One of the commonly used methods for inserting punctuation marks into ASR output is using a language model. Using an n-gram LM for punctuation insertion,

without acoustic cues, can be considered to be the baseline of baselines. We also investigate the use of state-of-art LSTM LMs and additionally, LSTMs that are trained for sequence labeling. This means that we do not predict the next token at every time step as LMs do but we predict whether the current word should be followed by a punctuation symbol or not. The last method is specifically trained for punctuation prediction and greatly reduces the number of possible output classes—from the whole vocabulary to the set of punctuation symbols and a symbol indicating 'no punctuation'.

Monolingual translation—Peitz et al. [76] show improvements in BLEU score when using a monolingual translation system to translate from unpunctuated to punctuated text instead of an LM-based punctuation prediction method. They also do a system combination of hypotheses from different approaches and get an additional improvement in BLEU score. They assume correct sentence segmentation.

We train different configurations of monolingual MT systems from non-punctuated Dutch to punctuated Dutch (to be used before the regular Dutch to English MT system), from non-punctuated Dutch to punctuated English, from non-punctuated Dutch to non-punctuated English, from punctuated Dutch to punctuated English and from non-punctuated English to punctuated English. When we take the best configurations of each of these systems, we can measure total MT quality from unpunctuated Dutch to punctuated English in different conditions, as shown in Figure 11.

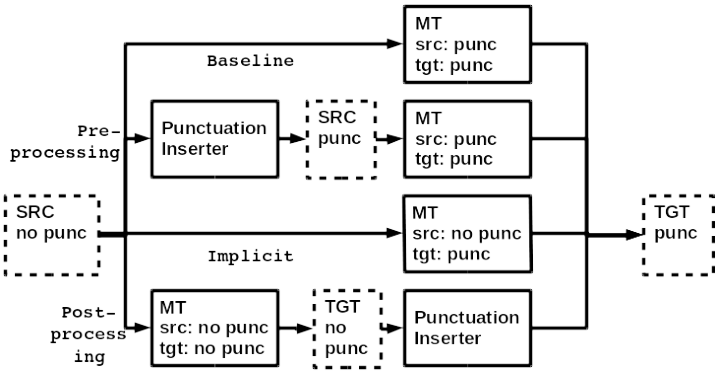


Figure 11. The different punctuation prediction strategies in a translation context.

In the **Baseline** we translate unpunctuated Dutch with the regular (punctuated) Dutch-English MT engine. In **Preprocessing**, we translate unpunctuated Dutch to punctuated Dutch and take that output and translate it to English using the regular Dutch-English MT engine. In **Implicit Punctuation**, we translate unpunctuated Dutch to punctuated English using an MT system trained on unpunctuated Dutch as source and normal, puncuated English as target. In **Postprocessing**, we translate unpunctuated Dutch to unpunctuated English using an MT system trained on unpunctuated data for both languages. We take the output (unpunctuated English) and translate it to punctuated English using an MT engine trained on unpunctuated English to normal English.

Besides these different configurations, we also use different MT models: phrase-based and hierarchical SMT and neural MT. These MT paradigms are tested both for the monolingual systems and the bilingual systems. In total, by combining the n-gram LMs, LSTM LMs, LSTM sequence labeling, phrase-based SMT, hierarchical SMT and NMT as punctuation prediction models with the different configurations to insert the punctuation (pre-MT, during-MT or post-MT) and the three MT models for the actual translation, we tested 145 different experimental conditions. Since all setups are trained and tested on the same data, this provides us a thorough comparison of punctuation prediction methods.

While there is a clear deterioration of MT quality when working with unpunctuated input, this gap can be closed for 66% in the case of our best MT system (NMT) by applying monolingual MT as punctuation insertion or by using a dedicated implicit insertion MT system. Whether we use pre- or post-processing did not result in a significant difference, in most cases indicating that the general

punctuation prediction quality for Dutch is similar to that of English. Full details are available in Reference [77].

We also made some initial steps towards segmentation insertion. As MT systems work per segment (usually a sentence), the audio transcript is best divided into segments. This can be done based on auditory (length of pauses) or linguistic cues (lexical). Experimentation with different variants of these approaches will determine which is the most promising/best functioning approach.

6. The SCATE Interface

This section describe the SCATE prototype interface in more detail. Section 6.1 describes related work, Section 6.2 describes the research into *intelligibility* of the information presented to the user and Section 6.3 describes evaluations that were performed with end users, comparing different versions of the SCATE interface and comparing the SCATE interface to another state-of-the-art CAT system, called Lilt.

6.1. Related Work

As translators rely on their computer-aided translation tools (CAT tools) to increase their productivity, end user satisfaction has become essential when developing new tools. Previous studies have shown that these aspects have been rather neglected in the past and the user interface design has been driven by the needs of the translation clients and not by the needs of the translator [78,79].

Various surveys and field studies [46,80–85] investigating human-computer interaction, show that translators value improved translation memory (TM)–machine translation (MT) integration methods (e.g., copy/paste, drag-and-drop within editor). References [86–88] show that reuse of sub-segments is possible through interactive translation prediction (ITP), a method in which users are presented, as they type, with translation suggestions from all available resources.

Suggestions are displayed either in a drop-down list or directly under the target segment. Translators seem to prefer ITP to classical post-editing because it minimises the number of keystrokes and thus increases productivity [89,90]. Commercial translation software developers have implemented this technology in different ways and use different terminology to refer to it, such as *predictive typing*, *AutoSuggest*, *Autocomplete*, or *Autowrite*.

Reference [91] shows that metadata can help translators make well-informed decisions. He concludes that metadata "helps translators adapt their translation strategies more easily according to the suggestion type". Reference [80] indicates that translators like information about the provenance of the MT suggestions and estimation of their quality. In the context of post-editing, Reference [92] argues that translators value on-the-fly highlighting of word alignment in order to keep the connection between source and target text. In other words, it appears useful to explicitly link parts of a source sentence with parts of the translation suggestion.

In SCATE we developed visual aids that explain the origin of the translation suggestions and their link with the source text.

The user interface of SCATE was designed and developed using a user-centered approach that involved professional translators in every stage of the process. A version with only cached results from the translation features is available at http://scate.edm.uhasselt.be/. We used the web-based survey on translation methods (Section 4.1) to better understand the needs of professional translators. The survey results show that ease of use is the most important motivation for choosing a translation environment, closely followed by speed of performance and features such as management of TMs and term bases. We used contextual inquiries and interviews to refine and enrich the insights on how translators work. These insights were used to define requirements for translation environments [46]. Based on these requirements, we implemented roughly four iterations of high-fidelity prototypes for the user interface. Each iteration was demonstrated to user groups form various translation companies to get feedback early. The first iteration of the prototype focused on the integration of translation suggestions from machine translation, translation memories and terminology databases, whereas later iterations focus

on the intelligibility of these suggestions [93]. Finally, we performed two rounds of evaluation on the final prototype with end-users. We explored the impact of intelligibility in a comparative study with twenty-six professional translators (Section 6.3.1). In the second round, we recruited four professional translators to compare our translation environment to Lilt [94] (Section 6.3.2).

6.2. Intelligible Translation Suggestions

The interface visualises four established translation features: a term base that contains terms and their possible translation (in this case an automatically extracted term base, details of the extraction process are discussed by Coppers et al. [93]), fuzzy matches from a TM (Section 2.1), output of an MT system (Section 2.2) and auto-completion to predict a word or even a word group. In existing translation environments, such features often act like black boxes and provide only limited justification for their suggestions [95]. In order to improve trust [96], our interface explains where translation suggestions come from, in what context(s) they have been used before and how often they have been used by other translators (Figure 12). As a result, translators can make quick and well-informed decisions on the suitability of multiple alternatives in a particular translation context.

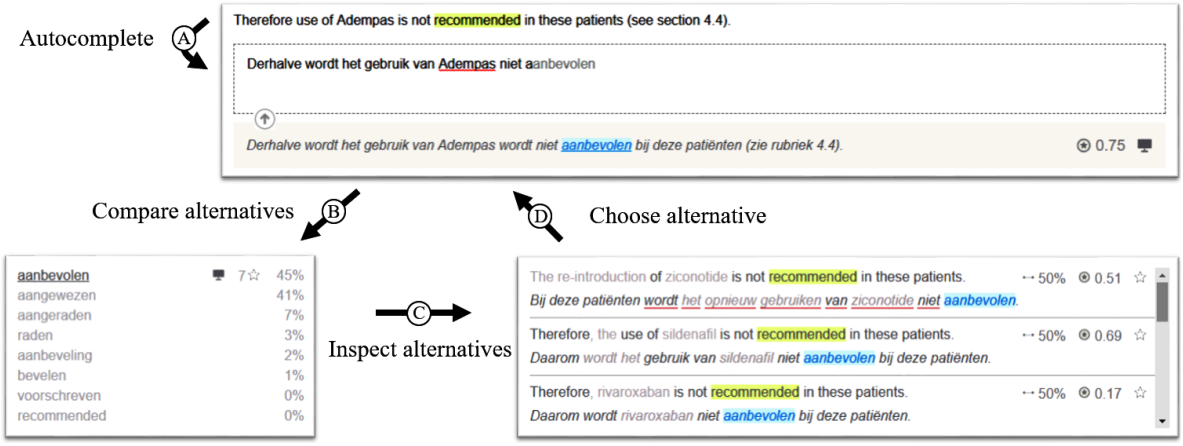


Figure 12. All translation suggestions are closely related to each other. When a translator types a character, (**A**) the auto-completion algorithm generates a suggestion. (**B**) The translator compares this prediction to other alternatives. (**C**) Interesting alternatives can be inspected in the context in which they have been used by other translators. (**D**) When a translator decides which alternative to use, it can be added to the translation by pressing ENTER.

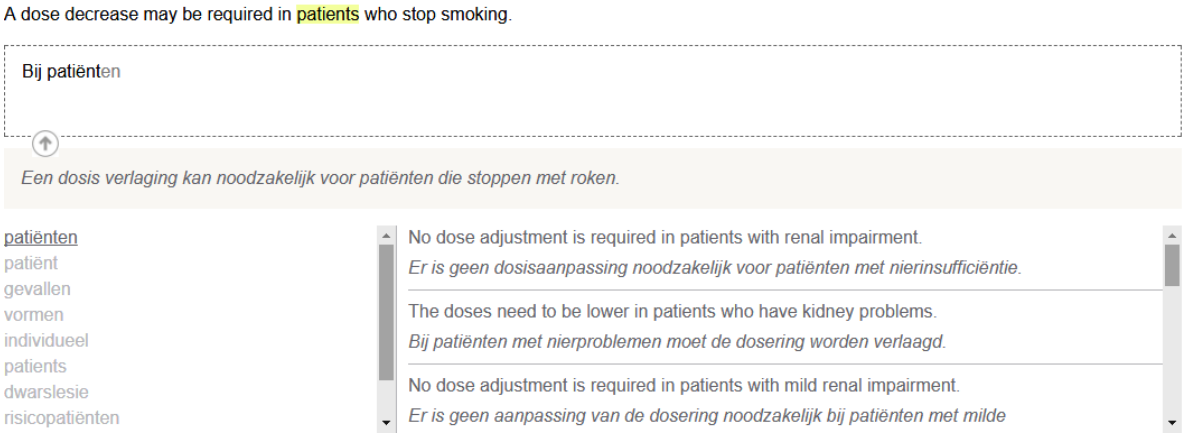
In order to efficiently combine sub-segments from various sources such as MT, TM and TB, the SCATE interface contains an auto-completion feature that uses these sources to suggest (the remainder of) a word or word group (Figure 12A). By pressing ENTER, the translator can add this suggestion to the translation. The algorithm justifies its prediction by selecting the suggestion in the sorted list of alternatives (Figure 12B), which shows several icons and metrics to explain where each alternative comes from (e.g., MT, TM and TB) and how often it has been used before by other translators. The occurrences themselves are highlighted in blue in the automatic translation and in the fuzzy matches (Figure 12C) to allow quick inspection of similar use cases. The automatic translation is shown very close to the sentence to translate and stays directly available to the translator at any time (Figure 12). As described in Section 2.2, parts in the automatic translations can be pretranslated by parts from the fuzzy matches. This behavior is made clear to the translator by printing these parts in bold in the automatic translation and in the matches they originate from.

Similar to other translation environments, the SCATE interface presents a similarity metric along the fuzzy match to make clear how similar the sentence is. In contrast with existing environments, parts that are similar according to the matching algorithm used are highlighted, rather than the differences. Furthermore, the quality of each suggestion is determined by estimating the number of post-edits that

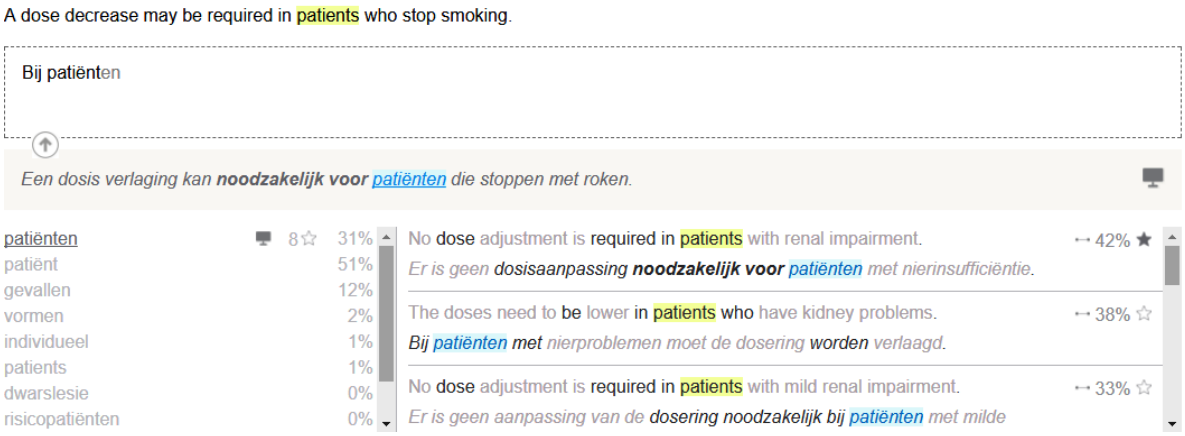
are still needed, using the technique described in Section 3.2.3. This estimate is normalized to a value between 0 and 1, with 1 representing a score for a sentence that would not need post-editing. Parts of the suggestion that probably require post-editing, are underlined in red. These visualisations help the translator to quickly understand why a match was similar and how its translation might be useful.

As a result of the tight integration of suggestions from various sources, a translator can explore up to four different relationships between suggestions at once: (1) the relationship between words and word groups in the input sentence, (2) synonym recommendations, (3) source and destination sentence in match recommendations and (4) the recommended automatic translation. As an additional advantage, all translation aids require only limited space and can be combined into a compact recommendation overview.

During a feedback round with 8 professional translators, we found that an intelligible visualisation is only perceived as useful when the information it conveys benefits the translation process and when this information is not part of the translator's readily available knowledge. For example, visualising the morphological function of the suggested alternatives (e.g., "noun") can be perceived as distracting instead. For this reason, the interface allows translators to disable the additional metrics and highlighting according to their own preferences. Figure 13a shows the interface with all explanations enabled whereas Figure 13b has explanations disabled without compromising the functionality.



(**a**) The simple version of the interface with all explanations disabled



(**b**) The intelligible version of the interface with all explanations enabled

Figure 13. Two configurations of the SCATE interface with the same functionality and suggestions.

The prototype features end-user control over the workflow (Figure 14). A user can control whether a segment requires a review, can be rejected by a reviewer or can still be edited after confirmation.

Furthermore, it can be configured whether these transitions should be automated between phases and each segment can be assigned to a translator and reviser. By default, no such constraints are enforced.

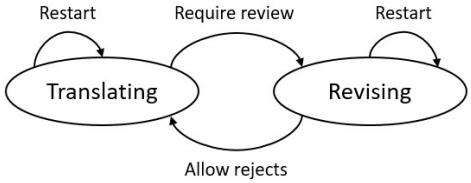


Figure 14. Each transition in the workflow is optional and can be enforced by the translation environment.

6.3. Evaluation

Section 6.3.1 describes an experiment measuring the effect of visualisations and intelligibility features. Section 6.3.2 describes a user evaluation in which we compare the Lilt (http://www.lilt.com/) and SCATE interfaces.

6.3.1. Influence of Visualisation on Experience and Preference

To investigate the impact of intelligible translation aids on the translation process, we perform a within subject user study with 26 professional translators. All participants translate two pieces of a text of comparable difficulty using the two configurations of the SCATE interface shown in Figure 13. The order in which they use each version of the interface is counterbalanced. After each condition participants fill out a survey about the interface, with an additional comparative survey at the end.

The subjects are positive to very positive about both versions of the interface in the survey questions. Analysis of the results shows that the visualisations help professional translators to assess the quality of the generated suggestions and help to understand how these suggestions can be used in translation, without distracting or negatively impacting efficiency. Intelligible visualisations do not affect the quality of translation suggestions themselves but instead inform translators about their quality and context to support better decision making. Translators only prefer intelligible translation aids when the additional information benefits the translation process and when this information is not yet part of the translator's readily available knowledge. Coppers et al. [93] provide more details about the study.

6.3.2. Comparison with Lilt

In the second round of evaluation, we carry out a user study with four professional translators and compare the SCATE prototype to Lilt, a commercial translation environment that stems from research aiming to optimally combine human translation with MT [97]. The two systems under study can be considered as examples of a new generation of translation environment tools in the sense that they differ from the mainstream and most frequently used systems among translators such as SDL Trados Studio, Wordfast and memoQ in the following respects: (1) Both systems offer a tighter integration of MT and TM suggestions than the mainstream systems, giving MT a more prominent place. However, both systems adopt a fundamentally different approach to reach this goal. (2) Both systems present the active segment more centrally on the screen and the source and target text are presented vertically instead of horizontally in the standard view. Apart from that, they offer advanced user interaction features such as autocompletion and a variety of shortcuts to copy the different types of suggestions (TM, MT, alternative translations for words or fragments).

The interfaces of both translation environments share several aspects, such as the central placement of the active segment and the order in which source, target, automatic translation and alternatives are presented. When the experiment was carried out, the underlying MT architecture in both systems was SMT. At the moment of writing, Lilt has replaced the SMT by NMT engines. The main differences can be summarised as follows: (1) SCATE always shows suggestions from

multiple sources, whereas Lilt offers these suggestions on demand. (2) Additional information about alternatives is displayed on the right-hand side of the user interface (memory search) in Lilt and is initially hidden, whereas this information is always present in the SCATE interface. (3) Lilt shows only one suggestion for the whole segment, while in SCATE the list of fuzzy matches is not limited to one. (4) Lilt uses adaptive MT while SCATE uses non-adaptive hybrid MT. (5) In SCATE, parts of suggestions, such as MT and fuzzy matches, can be used by double clicking individual words, which will add them to the translation. (Clicking once on any word will search for new alternative translations for that word.) (6) In SCATE, information is given about the source of the translation suggestions (hybrid MT, TM or term list) and additional scores are given (frequency, fuzzy match scores and a quality estimation score) whereas in the Lilt interface the source of the suggestion (TM or MT) can only be derived from the presence or absence of the fuzzy match percentage.

Four professional translators were paid for their participation (50 Euros per hour, 150 Euros in total) and signed an informed consent form. Prior to the experiment the participants were asked about their previous experience with translation and the use of translation environments. Next, they worked through a tutorial to become familiar with the user interface of either Lilt or SCATE, after which they translated a text 'for real' using the same interface. This first part was completed by a survey that asked about their experiences with the first environment. After that, they similarly worked through a tutorial, a translation session and a survey of the other interface. The experiment ended with a post-experiment survey reporting on their experience with the two interfaces.

As the SCATE prototype's MT component has exclusively been trained on English and Dutch medical texts, text selection for the experiment was also limited to medical material for this language combination. As none of the participants were experienced medical translators, text fragments were chosen from package leaflets intended for patients, on the assumption that these would be more manageable for the test subjects than highly technical texts. SCATE's corpus material is the English-and-Dutch EMEA TM as available through OPUS [98]. Although this is based on so-called EPARs (European Public Assessment Reports) rather than patient leaflets, both text types originate from the European Medicines Agency and share many features.

Care was taken to select texts on relatively new medicines that did not already feature in the EMEA TM. Two text fragments of equal size (175 words each) were prepared for the tutorials and two further fragments (225 and 232 words, 20 segments) were used for the actual translation. The test subjects' activities during translation were monitored using Inputlog [99] for keylogging and Camtasia (https://www.techsmith.com/video-editor.html) for screen recording. The order of texts and environments tested was balanced across participants. Although we could not fully control the Lilt environment, care was taken to create translation conditions that were as similar as possible. The same TM was used in both systems (198 K segments, 5.3 million words in total) and a manually created term list of 360 medical term pairs was uploaded in both systems. To keep conditions stable across participants, we also hid SCATE's button that enabled users to customise the interface (to turn features on/off).

Table 2 gives an overview of the experiments that were carried out. The order of the two environments and the two texts were balanced across participants. As the texts were of similar nature and length, it was potentially interesting to check whether working in one interface rather than another was faster or slower. A comparison of the total number of minutes spent per text per participant, however, suggests that the difference seems to be more related to individual speed rather than to the interface used, with P2 and P3 being faster in Lilt than P1 and P4 and P1, P3 and P4 having a similar speed in SCATE.

Table 2. Per participant, the order of the experiments, the environment used, the text that was translated and the total time expressed in minutes.

Participant		Environment	Text	Total Time
P1	Exp1	Lilt	Text1	23
	Exp2	SCATE	Text2	19
P2	Exp1	Lilt	Text2	17
	Exp2	SCATE	Text1	14
P3	Exp1	SCATE	Text1	19
	Exp2	Lilt	Text2	15
P4	Exp1	SCATE	Text2	19
	Exp2	Lilt	Text1	27

The study provides us with useful insights. Two translators (P2 and P3) started working on a segment immediately after opening and combined different strategies: typing, inserting suggested words as well as starting from the complete translation suggestion which they then revise or accept. The two other translators (P1 in SCATE and P4 in both interfaces) preferred copying a complete translation suggestion, (which could be either an MT or a TM suggestion) to the edit box to start from. One translator (P4) pauses for a long time before she takes action. This finding is in line with Reference [100], in which two production styles were distinguished: translators either translate a segment mentally and then type it (Prospective Thinking) or they translate as they were reading the text (Translating On-screen). In all screen recordings we noticed that, despite the training phase, the individual strategies evolved over time, a finding that was also reported by Koehn [101], in which a learning effect is described.

Figure 15 shows the percentage of time devoted to keystrokes versus mouse actions in both translation environments. Again, individual differences can be observed. P1 has a noticeably higher number of mouse actions than keystrokes, which is not surprising as she does not use any shortcuts in either environment. P2 has a remarkably higher number of keystrokes in SCATE compared to Lilt, which can be explained by his own comment in the survey that "in SCATE there was more typing work when diverting from the original suggestion."

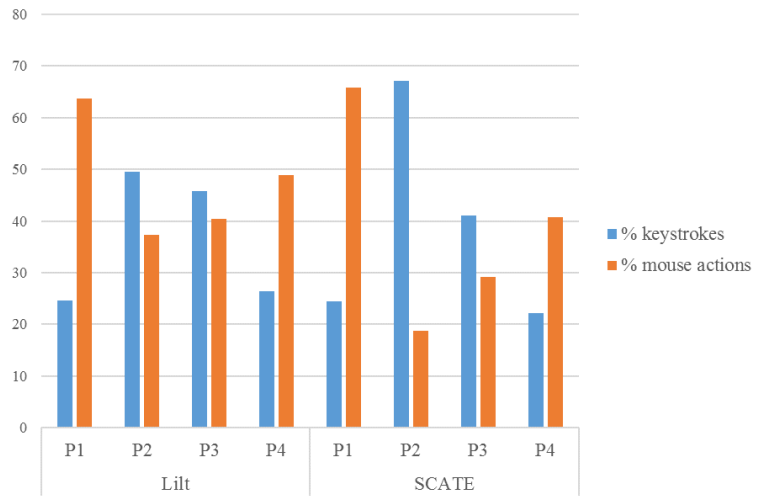


Figure 15. Percentage of time devoted to keystrokes vs. mouse actions per translation environment per participant.

Figure 16 presents the total number of characters typed versus the number of keys pressed to delete text (delete, backspace). No distinction has been made between the typing activity in- or outside

the Lilt or SCATE environment. This figure demonstrates the benefits of using interactive translation environments. Even P1, who produced most characters, only types around 700–800 characters to translate a source document of 1300–1450 characters. A more drastic decrease can even be seen in P2 and P3 in Lilt and P3 and P4 in SCATE, with fewer than 400 characters typed.

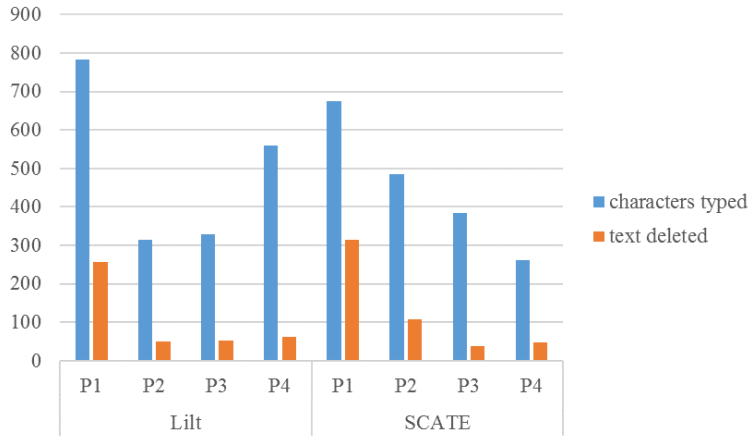


Figure 16. Total number of characters typed versus text deleted per translation environment per participant.

To exemplify the minimal typing effort, Figure 17 shows how P2 produced the translation '*Licht uw arts in als u maag-of darmproblemen hebt (gehad)*' (English: *Inform your doctor if you (have) had stomach or bowel problems.*) in the SCATE environment. The letters in dark blue are the characters that were actually typed; [RETURN] is used to insert/accept the suggested word; [BACK] to delete characters and [CTRL+RETURN] to confirm the translation.

```
{13339}[RSHIFT]L[RETURN][RETURN][RETURN][RETURN]
[RETURN][RETURN]{6680}pro[RETURN]hebt·[RSHIFT](gehad
[RSHIFT])·m[RETURN][RETURN][RETURN][RETURN][RETURN]
[BACK][BACK][BACK][BACK][BACK][BACK][BACK][BACK]
[BACK][BACK][BACK][BACK][BACK][BACK][BACK][BACK]
[BACK][BACK][BACK][BACK][BACK][BACK].[Movement][LEFT
Click]maag-·of·darm{2168}[RCTRL][RCTRL + RETURN]
```

Figure 17. Example of how a translation is produced in SCATE.

Features of the new translation environment tools that were valued most by the participants are the clean and calm design of the user interface of both systems, the interactive and adaptive MT of Lilt and the frequency information of translation alternatives of SCATE. Translators find quality estimation scores only useful when they are interpretable (the range of the scores should be clear) and when they are in line with the more traditional fuzzy match scores that translators are acquainted with. Translators would also like to know the origin of the suggestions (the difference between a TM or MT suggestion was not clear in Lilt) and they find a concordance search indispensable. An 'undo'-button would also be appreciated. The translators also raised concerns about the new interactive way of translating as translators might be more inclined to produce translations word by word. Starting from MT suggestions might have a negative impact on the overall readability of the text produced and translators might become less focused when they are presented with good translations automatically.

Perhaps the most important conclusion of this study is that translators differ from each other in the way they work. We observe individual preferences to interact with the system (shortcuts versus mouse) and different ways of using the suggestions (copying the complete suggestion followed by revision or gradually building up a translation by accepting appropriate suggestions) and it is important for CAT tools to support these different styles of working.

Customisability of the user interface (a feature that we disabled to keep experimental conditions stable) seems extremely important. This was also at the top wish list of the respondents in Reference [80] to assess the user interface needs of post-editors of MT.

7. Conclusions

We present an overview of the research that is performed in the SCATE project. We show the coherence between several different aspects of our research and how they all relate to the translator's professional workflow. Although several aspects have been published before in isolation, this paper provides the broader context and presents additional research.

We describe how several aspects of the translation technologies can be improved, such as fuzzy matching, integrating TM and MT technologies and parallel treebanks for syntax-based MT. We are convinced that acceptance of MT by the translator's community can grow through such an integration of TM and MT.

We delve into quality estimation research on the word and sentence level and as byproducts, we built a taxonomy of MT errors and a corpus of manual post-editing and annotation of the MT errors according to this taxonomy. These data allow to build informative quality estimation systems, not only indicating what goes wrong but also providing information on why this is the case.

We study translator's methods towards terminology extraction from comparable text and try out different approaches to this problem, depending on the domain. We show that it is possible to do this with only small supervision data sets.

We investigate several aspects of speech recognition in the context of translation, such as post-editing through speech and automatic domain adaptation, where we show that speech recognition can be improved by using information from within the translation engine and by working on the character level to solve the unknown word problem. We also performed an experiment to find out the best approach towards punctuation insertion in speech translation.

Last but not least we present a user interface, based on user observation in practice, which provides a proper integration of many of the above described aspects into a convenient working environment using intelligible translation suggestions coming from several different sources. We set up an experiment evaluating this user interface, comparing it to an existing commercial interface.

All these research aspects show potential in improving the translator's daily workflow, not only implying an improved productivity but also a customizable, more pleasant and calm, working environment.

Author Contributions: Conceptualization, V.V., T.V., J.P., L.V., P.W., M.-F.M., E.L., L.M., V.H., J.V.d.B. and K.L.; Funding acquisition, V.V., T.V. and F.V.E.; Investigation, V.V., T.V., L.A., B.B., J.P., L.V., G.H., I.v.d.L.-C., A.R.T., A.T., L.M., J.D., J.B., S.C. and J.V.d.B.; Methodology, V.V.; Project administration, V.V.; Supervision, V.V., F.V.E., P.W., M.-F.M., F.S., E.L., L.M., V.H. and K.L.; Visualization, S.C. and J.V.d.B.; Writing—original draft, V.V., T.V., B.B., L.V., G.H., I.v.d.L.-C., A.T., L.M., J.D., J.B. and S.C.; Writing—review & editing, V.V., P.W. and S.C.

Funding: The research in this project was funded by the Flemish Agency for Innovation and Technology IWT, project number 13007.

Acknowledgments: We would like to thank the companies and organizations taking part in the Industrial Advisory Committee of the SCATE project for their ideas, feedback and cooperation. These are Clarivate, CommArt International, CrossLang, ITP Europe, Mastervoice, Nuance, OneLiner, Televic, VRT Onderzoek en Innovatie, Xplanation, Yamagata-Europe, Yazzoom. We would also like to thank the additional translators that participated in our inquiries and evaluations. We thank the reviewers for their constructive comments.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

API	Application programming interface
ASR	Automated Speech Recognition
BiLDA	Bilingual Latent Dirichlet Allocation
BLEU	BiLingual Evaluation Understudy
BLI	Bilingiual Lexicon Induction
BWESG	Bilingual Word Embedding Skip Grams
CAT	Computer-aided Translation
C-BiLDA	Comparable Bilingual Latent Dirichlet Allocation
CBOW	Continuous Bag-of-Words
CLARIN	Comman Language Resources Research Infrastructure
DGT	Directorate General for Translation
DNT	Do-Not-Iranslate
EN	English
GrETEL	Greedy Extraction of Trees for Empirical Linguistics
GRU	Gated Recurrent Unit
HTER	Human-targeted Translation Edit Rate
IAA	Inter-Annotator Agreement
ITP	Interactive Translation Prediction
LENG	Lexical Equivalent Node Grouping
LM	Language Model
LSA	Latens Semantic Analysis
LSTM	Long Short-term Memory
METEOR	Metric for Evaluation of Translation with Explicit ORdering
ML	Machine Learning
MT	Machine Translation
NL	Dutch
NMT	Neural Machine Translation
OOV	Out-of-Vocabulary
PET	Post-Editing Time
PoS	Part-of-Speech
QE	Quality Estimation
RBMT	Rule-based Machine Translation
RNN	Recurrent Neural Network
SCATE	Smart Computer-Aided Translation Environment
SMT	Statistical Machine Translation
TAP	Think Aloud Protocol
TB	Term-Base
TBX	Term-Base eXchange
TEnT	Translation Environment
TER	Translation Edit Rate
TM	Translation Memory
UI	User Interface
VRT	Vlaamse Radio en Televisie
WMT	Workshop on Machine Translation

References

1. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
2. Snover, M.; Madnani, N.; Dorr, B.; Schwartz, R. TER-Plus: Paraphrase, semantic, and alignment enhancements to Translation Edit Rate. *Mach. Transl.* **2009**, *23*, 117–127. [CrossRef]

3. Bloodgood, M.; Strauss, B. Translation Memory Retrieval Methods. In Proceedings of the 14th Conference of the European Association for Computational Linguistics, Gothenburg, Sweden, 26–30 April 2014; pp. 202–210.
4. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 311–318. [CrossRef]
5. Lavie, A.; Agarwal, A. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In Proceedings of the Second Workshop on Statistical Machine Translation StatMT '07, Prague, Czech Republic, 23 June 2007; Association for Computational Linguistics: Stroudsburg, PA, USA, 2007; pp. 228–231.
6. Prüfer, H. Neuer Beweis eines Satzes über Permutationen. *Arch. Math. Phys.* **1918**, *27*, 742–744.
7. Vanallemeersch, T.; Vandeghinste, V. Assessing linguistically aware fuzzy matching in Translation Memories. In Proceedings of the 18th Annual Conference of the European Association for Machine Translation, Antalya, Turkey, 11–13 May 2015; pp. 153–160.
8. Koehn, P.; Senellart, J. Convergence of translation memory and statistical machine translation. In Proceedings of the AMTA Workshop on MT Research and the Translation Industry, Denver, CO, USA, 4 November 2010; pp. 21–31.
9. Steinberger, R.; Eisele, A.; Klocek, S.; Pilos, S.; Schlüter, P. DGT-TM: A freely available Translation Memory in 22 languages. *arXiv* **2013**, arxiv:1309.5226.
10. Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; et al. Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the 45th Annual Meeting of the ACL Interactive Poster and Demonstration Sessions, Prague, Czech Republic, 25–27 June 2007; Association for Computational Linguistics: Prague, Czech Republic, 2007; pp. 177–180.
11. Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; Rush, A.M. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the ACL, Vancouver, BC, Canada, 30 July–4 August 2017; doi:10.18653/v1/P17-4012.
12. Bulté, B.; Vanallemeersch, T.; Vandeghinste, V. M3TRA: integrating TM and MT for professional translators. In Proceedings of the 21st Annual Conference of the European Association for Machine Translation, Alacant/Alicante, Spain, 28–30 May 2018.
13. Kotzé, G.; Vandeghinste, V.; Martens, S.; Tiedemann, J. Large Aligned Treebanks for Syntax-based Machine Translation. *Lang. Resour. Eval.* **2016**, *51*, 249–282. [CrossRef]
14. Koehn, P. Neural Machine Translation. *arXiv* **2017**, arxiv:1709.07809.
15. Vandeghinste, V.; Martens, S.; Kotzé, G.; Tiedemann, J.; Van den Bogaert, J.; De Smet, K.; Van Eynde, F.; van Noord, G. Parse and Corpus-based Machine Translation. In *Essential Speech and Language Technology for Dutch: Results by the STEVIN Programme*; Spyns, P.; Odijk, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 305–319.
16. Williams, P.; Sennrich, R.; Post, M.; Koehn, P. *Syntax-Based Statistical Machine Translation*; Synthesis Lectures on Human Language Technologies; Morgan & Claypool Publishers: San Rafael, CA, USA, 2016.
17. Li, J.; Xiong, D.; Tu, Z.; Zhu, M.; Zhang, M.; Zhou, G. Modeling Source Syntax for Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 688–697.
18. Eriguchi, A.; Tsuruoka, Y.; Cho, K. Learning to Parse and Translate Improves Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 2, pp. 72–78.
19. Aharoni, R.; Goldberg, Y. Towards String-To-Tree Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 132–140. [CrossRef]
20. Nadejde, M.; Reddy, S.; Sennrich, R.; Dwojak, T.; Junczys-Dowmunt, M.; Koehn, P.; Birch, A. Predicting Target Language CCG Supertags Improves Neural Machine Translation. In Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, 7–8 September 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 68–79. [CrossRef]

21. Chen, X.; Liu, C.; Song, D. Tree-to-tree Neural Networks for Program Translation, 2018. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, QC, Canada, 2–8 December 2018.
22. Och, F.; Ney, H. A Systematic Comparison of Various Statistical Alignment Models. *Comput. Linguist.* **2003**, *29*, 19–51. [CrossRef]
23. Macken, L. Analysis of Translational Correspondence in view of Sub-sentential Alignment. In Proceedings of the METIS-II Workshop on New Approaches to Machine Translation, Leuven, Belgium, 11 January 2007; pp. 97–105.
24. Kotzé, G. Complementary Approaches to Tree Alignment: Combining Statistical and Rule-Based Methods. Ph.D. Thesis, University of Groningen, Groningen, The Netherlands, 2013.
25. Tiedemann, J. Lingua-Align: An Experimental Toolbox for Automatic Tree-to-Tree Alignment. In Proceedings of the Seventh International Conference on Language Resources and Evaluation LREC, Miyazaki, Japan, 7–12 May 2010.
26. Zhechev, V.; van Genabith, J. Maximising TM Performance through Sub-Tree Alignment and SMT. In Proceedings of the Ninth conference of the Association for Machine Translation in the Americas, Denver, CO, USA, 31 October–5 November 2010.
27. Vanallemeersch, T. Data-driven Machine Translation using Semantic Tree Alignment. Ph.D. Thesis, KU Leuven, Leuven, Belgium, 2017.
28. Augustinus, L.; Vandeghinste, V.; Vanallemeersch, T. Poly-GrETEL: Cross-Lingual Example-based Querying of Syntactic Constructions. In Proceedings of the 10th Language Resources and Evaluation Conference (LREC), Portorož, Slovenia, 23–28 May 2016.
29. Augustinus, L.; Vandeghinste, V.; Schuurman, I.; Van Eynde, F. GrETEL. A Tool for Example-Based Treebank Mining. In *CLARIN in the Low Countries*; Odijk, J., van Hessen, A., Eds.; Ubiquity Press: London, UK, 2017; Chapter 22, pp. 269–280.
30. Vandeghinste, V.; Augustinus, L. Making Large Treebanks Searchable. The SoNaR case. In *Challenges in the Management of Large Corpora (CMLC-2) Workshop Programme*; LREC: Reykjavík, Iceland, 2014.
31. Vanroy, B.; Vandeghinste, V.; Augustinus, L. Querying Large Treebanks : Benchmarking GrETEL Indexing. *Comput. Linguist. Neth. J.* **2017**, *7*, 145–166.
32. Bojar, O.; Chatterjee, R.; Federmann, C.; Graham, Y.; Haddow, B.; Huck, M.; Jimeno Yepes, A.; Koehn, P.; Logacheva, V.; Monz, C.; et al. Findings of the 2016 Conference on Machine Translation. In Proceedings of the First Conference on Machine Translation (Volume 2, Shared Task Papers), Berlin, Germany, 11–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 131–198. [CrossRef]
33. Bojar, O.; Chatterjee, R.; Federmann, C.; Graham, Y.; Haddow, B.; Huang, S.; Huck, M.; Koehn, P.; Liu, Q.; Logacheva, V.; et al. Findings of the 2017 Conference on Machine Translation (WMT17). In Proceedings of the Second Conference on Machine Translation (Volume 2: Shared Task Papers), Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 169–214.
34. Specia, L.; Blain, F.; Logacheva, V.; Astudillo, R.; Martins, A. Findings of the WMT 2018 Shared Task on Quality Estimation. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, Brussels, Belgium, 31 October–1 November 2018; pp. 689–709.
35. Tezcan, A.; Hoste, V.; Macken, L. SCATE Taxonomy and Corpus of Machine Translation Errors. In *Trends in E-tools and Resources for Translators and Interpreters*; Approaches to Translation Studies; Pastor, G.C., Durán-Muñoz, I., Eds.; Brill | Rodopi: Leiden, The Netherlands, 2017; Volome 45, pp. 219–244.
36. Macken, L.; De Clercq, O.; Paulussen, H. Dutch Parallel Corpus: A Balanced Copyright-Cleared Parallel Corpus. *Meta J. Trad./Meta Transl. J.* **2011**, *56*, 274–390. [CrossRef]
37. Tezcan, A.; Hoste, V.; Macken, L. Estimating Post-Editing Time Using a Gold-Standard Set of Machine Translation Errors. *Comput. Speech Lang.* **2018**, *55*, 120–144. [CrossRef]
38. Van Noord, G. *At Last Parsing is Now Operational, TALN06. Verbum Ex Machina. Actes de la 13e Conference sur le Traitement Automatique des Langues Naturelles*; Leuven University Press: Leuven, Belgium, 2006; pp. 20–42.
39. Martins, A.F.; Astudillo, R.F.; Hokamp, C.; Kepler, F. Unbabel's Participation in the WMT16 Word-Level Translation Quality Estimation Shared Task. In Proceedings of the First Conference on Machine Translation, Berlin, Germany, 11–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 806–811.

40. Tezcan, A.; Hoste, V.; Macken, L. Detecting Grammatical Errors in Machine Translation Output Using Dependency Parsing and Treebank Querying. *Balt. J. Mod. Comput.* **2016**, *4*, 203–217.
41. Tezcan, A. Informative Quality Estimation of Machine Translation Output. Ph.D. Thesis, Ghent University, Ghent, Belgium, 2018.
42. Chollampatt, S.; Ng, H.T. Neural Quality Estimation of Grammatical Error Correction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018.
43. Specia, L.; Paetzold, G.; Scarton, C. Multi-level translation quality prediction with quest++. In Proceedings of the ACL-IJCNLP 2015 System Demonstrations, Beijing, China, 26–31 July 2015; pp. 115–120.
44. Beyer, H.; Holtzblatt, K. *Contextual Design: Defining Customer-Centered Systems*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1997.
45. Lewis, C. *Using the "Thinking Aloud" Method in Cognitive Interface Design*; Research Report; IBM TJ Watson Research Center: Cambridge, MA, USA, 1982.
46. Van den Bergh, J.; Geurts, E.; Degraen, D.; Haesen, M.; van der Lek-Ciudin, I.; Coninx, K. *Recommendations for Translation Environments to Improve Translators' Workflows*; Translating and the Computer 37; AsLing: London, UK, 2015.
47. Steurs, F.; van der Lek-Ciudin, I. *Report on Human Terminology Extraction. Deliverable D3.1*; Technical Report; KU Leuven: Leuven, Belgium, 2016.
48. Bowker, L. Productivity vs. Quality? A Pilot Study on the Impact of Translation Memory Systems. *Localis. Focus* **2005**, *4*, 13–20.
49. LeBlanc, M. Translators on translation memory (TM). Results of an ethnographic study in three translation services and agencies. *Transl. Interpret.* **2013**, *5*, 1–13. [CrossRef]
50. Delpech, E.M. Leveraging Comparable Corpora for Computer-assisted Translation. In *Comparable Corpora and Computer-Assisted Translation*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2014.
51. Bernardini, S.; Castagnoli, S. Corpora for translator education and translation practice. In *Topics in Language Resources for Translation and Localisation*; John Benjamins: Amsterdam, The Netherlands, 2008; pp. 39–55.
52. Blancafort, H.; Ulrich, A.X.; Heid, U.; Tatiana, S.C.; Gornostay, T.; Claude, K.A.L.; Méchoulam, C.; Daille, B.; Sharoff, S. User-centred Views on Terminology Extraction Tools: Usage Scenarios and Integration into MT and CAT Tools. In *Translation Careers and Technologies: Convergence Points for the Future (TRALOGY)*; INIST: Vandoeuvre-les-Nancy, France, 2011.
53. De Smet, W.; Moens, M.F. Cross-Language Linking of News Stories on the Web using Interlingual Topic Modeling. In Proceedings of the CIKM 2009 Workshop on Social Web Search and Mining (SWSM@CIKM), Hong Kong, China, 2 November 2009; pp. 57–64.
54. Heyman, G.; Vulić, I.; Moens, M.F. C-BiLDA Extracting Cross-lingual Topics from Non-Parallel Texts by Distinguishing Shared from Unshared Content. *Data Min. Knowl. Discov.* **2016**, *30*, 1299–1323. [CrossRef]
55. Vulić, I.; Moens, M.F. Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Beijing, China, 26–31 July 2015; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 719–725. [CrossRef]
56. Vulić, I.; De Smet, W.; Moens, M.F. Identifying Word Translations from Comparable Corpora Using Latent Topic Models. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT), Portland, OR, USA, 19–24 June 2011; pp. 479–484.
57. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the NIPS, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
58. Heyman, G.; Vulić, I.; Moens, M.F. Bilingual Lexicon Induction by Learning to Combine Word-Level and Character-Level Representations. In Proceedings of the EACL, Valencia, Spain, 3–7 April 2017; pp. 1085–1095.
59. Heyman, G.; Vulić, I.; Moens, M.F. A Deep Learning Approach to Bilingual Lexicon Induction in the Biomedical Domain. *BMC Bioinform.* **2018**, *19*, 259. [CrossRef]
60. Rigouts Terryn, A.; Hoste, V.; Lefever, E. A Gold Standard for Multilingual Automatic Term Extraction from Comparable Corpora: Term Structure and Translation Equivalents. In Proceedings of the 11th Language Resources and Evaluation Conference (LREC), Miyazaki, Japan, 7–12 May 2018.

61. Rodríguez, L.; Reddy, A.M.; Ros, R.C. Efficient Integration of Translation and Speech Models in Dictation Based Machine Aided Human Translation. In Proceedings of the ICASSP, Kyoto, Japan, 25–30 March 2012; pp. 4949–4952.
62. Pelemans, J.; Vanallemeersch, T.; Demuynck, K.; Van hamme, H.; Wambacq, P. Efficient Language Model Adaptation for Automatic Speech Recognition of Spoken Translations. In Proceedings of the INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015; pp. 2262–2266.
63. Pelemans, J.; Vanallemeersch, T.; Demuynck, K.; Verwimp, L.; Van hamme, H.; Wambacq, P. Language Model Adaptation for ASR of Spoken Translations Using Phrase-based Translation Models and Named Entity Models. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, 20–25 March 2016; pp. 5985–5989. [CrossRef]
64. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by Latent Semantic Analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407. [CrossRef]
65. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arxiv:1301.3781.
66. Pelemans, J.; Verwimp, L.; Demuynck, K.; Van hamme, H.; Wambacq, P. SCALE: A Scalable Language Engineering Toolkit. In Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, 23–28 May 2016.
67. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
68. Verwimp, L.; Pelemans, J.; Van hamme, H.; Wambacq, P. Character-Word LSTM Language Models. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Volume 1: Long Papers, EACL 2017, Valencia, Spain, 3–7 April 2017; pp. 417–427.
69. Verwimp, L.; Van hamme, H.; Wambacq, P. TF-LM: TensorFlow-based Language Modeling Toolkit. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, 7–12 May 2018.
70. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
71. Boes, W.; Van Rompaey, R.; Verwimp, L.; Van hamme, H.; Wambacq, P. *Domain Adaptation for LSTM Language Models*; Computational Linguistics in the Netherlands: Leuven, Belgium, 2017.
72. Grave, E.; Joulin, A.; Usunier, N. Improving Neural Language Models with a Continuous Cache. In Proceedings of International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
73. Kuhn, R.; Mori, R.D. A Cache-Based Natural Language Model for Speech Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 570–583. [CrossRef]
74. Verwimp, L.; Pelemans, J.; Van hamme, H.; Wambacq, P. Information-Weighted Neural Cache Language Models for ASR. In Proceedings of the IEEE Workshop on Spoken Language Technology (SLT), Athens, Greece, 18–21 December 2018.
75. Matusov, E.; Mauser, A.; Ney, H. Automatic Sentence Segmentation and Punctuation Prediction for Spoken Language Translation. In Proceedings of the 2006 International Workshop on Spoken Language Translation, IWSLT 2006, Keihanna Science City, Kyoto, Japan, 27–28 November 2006; pp. 158–165.
76. Peitz, S.; Freitag, M.; Mauser, A.; Ney, H. Modeling Punctuation Prediction as Machine Translation. In Proceedings of the IWSLT, San Francisco, CA, USA, 8–9 December 2011; pp. 238–245.
77. Vandeghinste, V.; Verwimp, L.; Pelemans, J.; Wambacq, P. A Comparison of Different Punctuation Prediction Approaches in a Translation Context. In Proceedings of the Annual Conference of the European Association for Machine Translation EAMT, Alicante, Spain, 28–30 May 2018.
78. Lagoudaki, E. Translation Memories Survey 2006. User's perceptions around TM use. In *Translation and the Computer*; ASLIB: London, UK, 2006; Volume 28, pp. 1–29.
79. O'Brien, S.; O'Hagan, M.; Flanagan, M. Keeping an eye on the UI design of Translation Memory: How do translators use the "Concordance" feature ? In Proceedings of the 28th Annual Conference of the European Association of Cognitive Ergonomics, Delft, The Netherlands, 25–27 August 2010; pp. 187–190.
80. Moorkens, J.; O'Brien, S. Assessing User Interface Needs of Post-Editors of Machine Translation. *Hum. Issues Transl. Technol.* **2017**, 109–130.
81. Teixeira, C.S.C.; Moorkens, J.; Turner, D.; Vreeke, J.; Way, A. Creating a Multimodal Translation Tool and Testing Machine Translation Integration Using Touch and Voice. *Informatics* **2019**, *6*, 13. [CrossRef]

82. Pal, S.; Naskar, S.K.; Zampieri, M.; Nayak, T.; van Genabith, J. CATaLog Online: A Web-based CAT Tool for Distributed Translation with Data Capture for APE and Translation Process Research. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, Osaka, Japan, 11–16 December 2016; The COLING 2016 Organizing Committee: Osaka, Japan, 2016; pp. 98–102.
83. Herbig, N.; Pal, S.; van Genabith, J.; Krüger, A. Multi-Modal Approaches for Post-Editing Machine Translation. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems CHI'19, Glasgow, UK, 4–9 May 2019; ACM: New York, NY, USA, 2019; pp. 231:1–231:11. [CrossRef]
84. Nayak, T.; Pal, S.; Naskar, S.K.; Bandyopadhyay, S.; van Genabith, J. Beyond Translation Memories: Generating Translation Suggestions based on Parsing and POS Tagging. In Proceedings of the 2nd Workshop on Natural Language Processing for Translation Memories (NLP4TM-2016), Portoroz, Slovenia, 28 May 2016.
85. Orasan, C.; Parra, C.; Barbu, E.; Federico, M. (Eds.) *2nd Workshop on Natural Language Processing for Translation Memories (NLP4TM 2016)*; ELRA: Portoroz, Slovenia, 2016.
86. Koehn, P.; Haddow, B. Interactive Assistance to Human Translators using Statistical Machine Translation methods. In Proceedings of the MT Summit XII, Ottawa, ON, Canada, 26–30 August 2009; pp. 1–8.
87. Sanchis-Trilles, G.; Alabau, V.; Buck, C.; Carl, M.; Casacuberta, F.; García-Martínez, M.; Germann, U.; González-Rubio, J.; Hill, R.L.; Koehn, P.; et al. Interactive translation prediction versus conventional post-editing in practice: A study with the CasMaCat workbench. *Mach. Transl.* **2014**, pp. 217–235. [CrossRef]
88. Torregrosa Rivero, D.; Pérez-Ortiz, J.A.; Forcada, M.L. Comparative Human and Automatic Evaluation of Glass-Box and Black-Box Approaches to Interactive Translation Prediction. *Prague Bull. Math. Linguist.* **2017**, *108*, 97–108. [CrossRef]
89. Green, S.; Wang, S.I.; Chuang, J.; Heer, J.; Schuster, S.; Manning, C.D. Human Effort and Machine Learnability in Computer Aided Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1225–1236. [CrossRef]
90. Zaretskaya, A. The Use of Machine Translation among Professional Translators. In Proceedings of the EXPERT Scientific and Technological Workshop, Malaga, Spain, 26–17 June 2015; pp. 1–12.
91. Teixeira, C. The Impact of Metadata on Translator Performance: How Translators Work With Translation Memories and Machine Translation. Ph.D. Thesis, Universitat Rovira i Virgili and Katholieke Universiteit Leuven: Leuven, Belgium, 2014.
92. Vieira, L.N.; Specia, L. A Review of Translation Tools from a Post-Editing Perspective. In Proceedings of the 3rd Joint EM+/CNGL Workshop Bringing MT to the User: Research Meets Translators (JEC), Luxembourg, 14 October 2011; pp. 33–42.
93. Coppers, S.; Van den Bergh, J.; Luyten, K.; Coninx, K.; van der Lek-Ciudin, I.; Vanallemeersch, T.; Vandeghinste, V. Intellingo: An Intelligible Translation Environment. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada, 21–26 April 2018; ACM: New York, NY, USA, 2018; CHI '18, pp. 524:1–524:13. [CrossRef]
94. Green, S.; Heer, J.; Manning, C.D. The Efficacy of Human Post-Editing for Language Translation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, 27 April–2 May 2013; ACM: New York, NY, USA, 2013; pp. 439–448.
95. Sinha, R.; Swearingen, K. The Role of Transparency in Recommender Systems. In Proceedings of the Extended Abstracts on Human Factors in Computing Systems CHI EA '02, Minneapolis, MN, USA, 20–25 April 2002; ACM: New York, NY, USA, 2002; pp. 830–831. [CrossRef]
96. Bellotti, V.; Edwards, K. Intelligibility and Accountability: Human Considerations in Context-Aware Systems. *Hum.-Comput. Interact.* **2001**, *16*, 193–212._05. [CrossRef]
97. Green, S.; Chuang, J.; Heer, J.; Manning, C.D. Predictive Translation Memory: A Mixed-Initiative System for Human Language Translation. In Proceedings of the 27th annual ACM symposium on User Interface Software and Technology, Honolulu, HI, USA, 5–8 October 2014; ACM: New York, NY, USA, 2014; pp. 177–187.
98. Tiedemann, J. News from OPUS—A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In *Recent Advances in Natural Language Processing*; Nicolov, N., Bontcheva, K., Angelova, G., Mitkov, R., Eds.; John Benjamins: Amsterdam, The Netherlands, 2009; Volume V, pp. 237–248.

99. Leijten, M.; Van Waes, L. Keystroke Logging in Writing Research: Using Inputlog to Analyze and Visualize Writing Processes. *Writ. Commun.* **2013**, *30*, 358–392. [CrossRef]
100. Asadi, P.; Séguinot, C. Shortcuts, Strategies and General Patterns in a Process Study of Nine Professionals. *Meta* **2005**, *50*, 522–547. [CrossRef]
101. Koehn, P. A process study of computer-aided translation. *Mach. Transl.* **2009**, *23*, 241–264. [CrossRef]