Part of Speech Annotation of Intermediate Versions in the Keystroke Logged Translation Corpus

Tatiana Serbina
RWTH Aachen University
Kármánstraße 17-19
52062 Aachen, Germany
serbina@anglistik.rwth-aachen.de

Paula Niemietz
RWTH Aachen University
Kármánstraße 17-19
52062 Aachen, Germany
niemietz@anglistik.rwth-aachen.de

Matthias Fricke
RWTH Aachen University
Dennewartstraße 27
52068 Aachen, Germany
matthias.fricke@ima-zlw-ifu.rwth-aachen.de

Philipp Meisen
RWTH Aachen Universtiy
Dennewartstraße 27
52068 Aachen, Germany
philipp.meisen@ima-zlw-ifu.rwth-aachen.de

Stella Neumann
RWTH Aachen University
Kármánstraße 17-19
52062 Aachen, Germany
neumann@anglistik.rwth-aachen.de

Abstract

Translation process data contains non-canonical features such as incomplete word tokens, non-sequential string modifications and syntactically deficient structures. While these features are often removed for the final translation product, they are present in the unfolding text (i.e. intermediate translation versions). This paper describes tools developed to semi-automatically process intermediate versions of translation data to facilitate quantitative analysis of linguistic means employed in translation strategies. We examine the data from a translation experiment with the help of these tools.

1 Introduction

Within the area of translation studies, there is a growing interest in the investigation of the process-related aspects of translation (see e.g. Göpferich, 2008 for an overview). Insights into the ongoing translation process can be gained by conducting psycholinguistic experiments, often characterized through a combination of eye-tracking and keystroke logging methods (e.g. Alves et al., 2010; Jakobsen, 2011). The resulting process data is typically analyzed in terms of behavioral measures, such as pauses during text production and gaze patterns within the texts, linked to the more abstract level of cognitive processing during a translation task. We adopt a corpus perspective on the keystroke logs (Alves and Magalhães, 2004; Alves and Vale, 2009, 2011), which contain rich information on key presses and mouse clicks during a translation session. This perspective entails that the data present in the logs can be queried, enabling us to perform quantitative, linguistically informed analyses of the translations. We take into account not only originals and the corresponding final versions of translated texts – which are also present in the traditional parallel corpora used in translation studies and contrastive linguistics, e.g. the CroCo corpus (Hansen-Schirra et al., 2012) – but also the intermediate versions of translations. We define the intermediate versions as variants of the unfolding texts produced at certain points in time during the translation process. The explicit linguistic annotation of text versions proposed here is not found in existing data collections containing keystroke logs: for instance, the TPR database (Carl, 2012a) involves part of speech (POS) annotation of source and target language tokens but does not analyze the intermediate versions. Investigation of these text versions allows us to identify potential translation problems and strategies, contributing to our understanding of cognitive processing, and also to provide best practice solutions for problems encountered in machine translation.

Proceedings of LAW IX - The 9th Linguistic Annotation Workshop, pages 102–111,
Denver, Colorado, June 5, 2015. ©2015 Association for Computational Linguistics

However, in order to study specific research questions from the field of translation studies with the help of such a corpus, we first need a transformation of sequences of production, deletion and separation keystrokes (see section 3.1) into word tokens, their annotation with linguistic information and also alignment between originals and the corresponding translations. The present paper concentrates on completed work involving the tokenization and (semi-)automatic POS annotation of the intermediate versions identified in the unfolding translations.

The corpus presents a type of non-canonical language, which is to some extent comparable to spoken data, as it also contains online repairs of the ongoing text production (cf. Heeman and Allen, 1999). Online repair can take place when a word or a grammatical structure present in one of the intermediate versions is replaced by another variant, either immediately before the participant moves to the translation of the subsequent parts or at a later stage of the translation process. This can be shown using Example 1 taken from the keystroke logged translation corpus (KLTC). It contains the source text (ST), two intermediate versions of the unfolding translation (IT_1 and IT_2) and the target text (TT).

ST Crumpling a sheet of paper seems
IT_1 Ein Blatt Papier zu knüllen *scheint*
‘a leaf paper to crumple seems’
IT_2 Ein Blatt Papier zu knüllen
‘a leaf paper to crumple’
TT Ein Blatt Papier zu knüllen *erscheint*
‘a leaf paper to crumple appears’

Example 1. KLTC, translator A11.

From the intermediate versions of the text we know that the translator typed *scheint* ‘seems’, deleted it, and at a later point typed *erscheint* ‘appears’. In other words, this experiment participant replaced one verb with another nearly synonymous one, filling the same slot in the produced sentence. Apart from such cases, the corpus also contains several versions of the same word tokens along with incomplete tokens and structures. Taking into account these non-canonical features, traditional NLP tools have to be modified to some extent, in order to make the automatic processing of the process data feasible.

The type of data included in the current version of the keystroke logged translation corpus is described in section 2. Section 3 presents how our Tokenizer processes the intermediate translation versions and discusses alternative methods of POS annotation. In section 4 we show how these pre- and post-processing steps can help us in the analysis of translation studies phenomena. Finally, section 5 provides an outlook on the next steps.

2 Keystroke logged translation corpus

The data used for this study was collected using the keystroke logging software Translog II (Carl, 2012b) and the remote eyetracker Tobii TX 300. It comprises two source texts (two variants[1] of a popular-scientific text originally published in the journal *Scientific American*[2]), nine translations and the matching set of nine key logs. All translation participants are German L1 students of English linguistics with little or no experience in translation. During the translation task from English into German, they were allowed to consult the bilingual online dictionary *leo*.

The source and target texts considered in this paper contain a total of 2,188 words. This calculation does not include word tokens identified in the intermediate versions. At the present stage of the project, we have concentrated on this small data set to test the automatic annotation procedures that have been developed. Once the gold standard is established, we intend to apply these methods to annotate further data available within the corpus.

3 Processing intermediate versions

3.1 Tokenizer

The Tokenizer automatically searches for words and word tokens in a selected set of keystroke events identifying the intermediate versions of the target text. The initial data, created by Translog II,

[1] We used two variants of the source text in order to counterbalance grammatically simple and complex stimuli. This will allow us to investigate the link between grammatical complexity and cognition in future work.

[2] Scientific American Online, February 5, 2002, Sarah Graham: A New Report Explains the Physics of Crumpled Paper. http://www.scientificamerican.com/article.cfm?id=a-new-report-explains-the

consists of the source text (ST) and the final target text (TT) along with a list of all keystrokes, i.e. the keys pressed, and the timestamp of each keystroke during the translation process. In order to transform series of connected keystroke events into word tokens, each file is processed in a number of steps, as illustrated in (1).

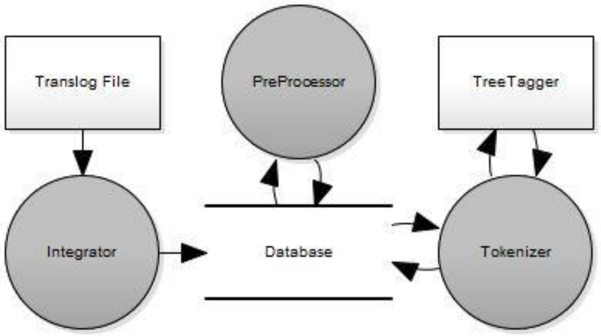
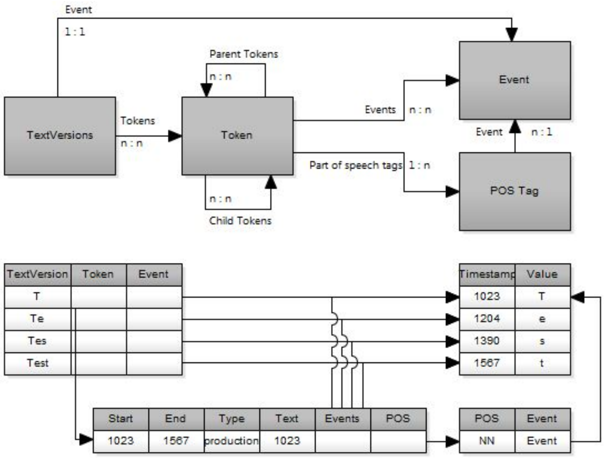


Figure 1. From a file to the annotated data.

In the first step (Integrator), the events as well as the ST and the TT, are loaded from the original XML files generated by Translog II into our corpus in which data is saved in the form of a database. This ensures easy and fast data access for future modification and annotation. In addition, the data quality is monitored through integrity checks. In the second step (PreProcessor), a type is assigned to each event based on the action performed. The types we used to categorize the events are *production* (letter keys or numbers), *deletion* (delete or backspace), *separation* (space, return or punctuation), *navigation* (use of the arrow keys or mouse to change the cursor position), *system* (for application specific messages like 'start' or 'stop logging') and *clipboard* (copy, paste and cut). This ensures the usage of normalized labels for all events across different Translog versions and applications. The third and last step (Tokenizer) replays the logged recording and creates different tokens and intermediate text versions. Each result is written into the database. Thus, the results are easily searchable, can be exported into a .tsv or other file for further analysis, or visualized by a GUI.

A token consists of the token string, a list of keystroke logging events that belong to the token, a list of parent tokens, a list of child tokens, and a list of POS tags (cf. section 3.2). If an existing token is modified in some way, it receives the label ‘parent’ and the modified version is referred to as its ‘child’ token. The Tokenizer also generates a version of the currently replayed text at each time an event caused a modification in the text. Figure 2 illustrates the data structure and an example for the token *Test*. As shown, the token *Test* was created by four events (*T*, *e*, *s* and *t*) and is classified as ‘production’ type. The target text (e.g. the character sequence *TextVersions*) is available after each event and refers to the token it belongs to. In addition, the created token is linked with its POS information.



TextVersion	Token	Event
T		
Te		
Tes		
Test		

Timestamp	Value
1023	T
1204	e
1390	s
1567	t

Start	End	Type	Text	Events	POS
1023	1567	production	1023		

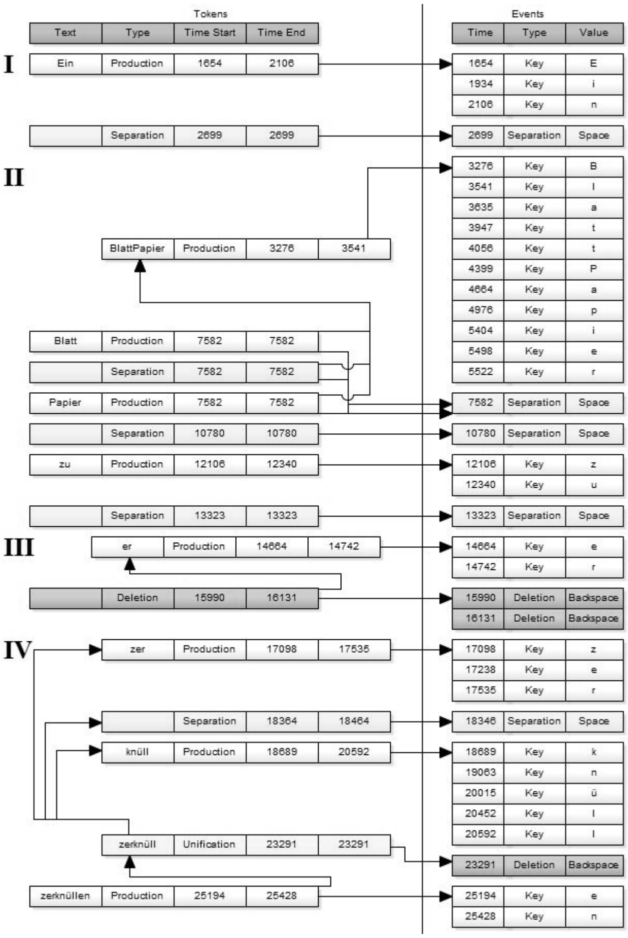
POS	Event
NN	Event

Figure 2. An example of a token and its data structure.

At each text modifying event (i.e. production, deletion, separation, or clipboard[3]) there are two possible actions, namely to extend an already existing token or create a new one. A token is extended if and only if the modifying event is not identical to that of the current token. For example, a word that is written in one production burst without an intervening deletion or navigation is always saved as one token (cf. Figure 3.I for production of the word token *ein* ‘a’). In contrast, a new token is created each time the type of the event differs. For example, if a word is separated into two words by typing a space, three new tokens are created (two word tokens of the type ‘production’ and one separation token), all having the same parent token. Figure 3.II shows this process for production of the two word tokens *Blatt* ‘sheet’ and *Papier* ‘paper’ from the sequence *BlattPapier*. If an existing token is shortened by an event of type ‘deletion’, a new token is generated which has the former production or separation token as its parent. Tokens that are

[3] The copy-clipboard event does not modify the text and is, therefore, ignored here. Nevertheless, the cut- or paste-events are handled as text modification.

deleted stay in the list of tokens and can be found in the keystroke logs exactly at the place where they have been deleted. A token present in the intermediate version can be deleted completely, so that it is not present in the final target text (cf. 3.III for deletion of the token *er*). Moreover, a deleted separation token can lead to a unification token that joins two separate tokens together into a new one (cf. 3.IV for the production of the word token *zerknüllen* 'scrunch' with an intermediate stage of the token *zerknüll* that is created by deleting the space within formerly separated tokens *zer* and *knüll*). The Tokenizer returns a list of tokens that were found in the recording as well as a list of text versions which represent every intermediate version of the target text at any given point in time.



Tokens

	Text	Type	Time Start	Time End
I	Ein	Production	1654	2106
		Separation	2699	2699
II	BlattPapier	Production	3276	3541
	Blatt	Production	7582	7582
		Separation	7582	7582
	Papier	Production	7582	7582
		Separation	10780	10780
	zu	Production	12106	12340
		Separation	13323	13323
III	er	Production	14664	14742
		Deletion	15990	16131
IV	zer	Production	17098	17535
		Separation	18364	18464
	knüll	Production	18689	20592
	zerknüll	Unification	23291	23291
	zerknüllen	Production	25194	25428

Events

Time	Type	Value
1654	Key	E
1934	Key	i
2106	Key	n
2699	Separation	Space
3276	Key	B
3541	Key	l
3635	Key	a
3947	Key	t
4056	Key	t
4399	Key	P
4664	Key	a
4976	Key	p
5404	Key	i
5498	Key	e
5522	Key	r
7582	Separation	Space
10780	Separation	Space
12106	Key	z
12340	Key	u
13323	Separation	Space
14664	Key	e
14742	Key	r
15990	Deletion	Backspace
16131	Deletion	Backspace
17098	Key	z
17238	Key	e
17535	Key	r
18346	Separation	Space
18689	Key	k
19063	Key	n
20015	Key	ü
20452	Key	l
20592	Key	l
23291	Deletion	Backspace
25194	Key	e
25428	Key	n

Figure 3. A result of the application of the Tokenizer.

3.2 Part of Speech annotation

As mentioned above, intermediate versions encountered in the keystroke logged translation corpus exhibit features typically associated with non-canonical data. As such, they can be compared to other types of non-standard language, including computer-mediated communication or learner texts. Previous studies in this area have noted the challenges of applying the existing NLP tools and tagsets, which are often trained on the basis of newspaper language, to the data that deviates from this standard (Neunerdt et al., 2013; Zinsmeister et al., 2014). This issue is addressed by development of modified taggers as well as adaptations of the tagsets, for instance to include tags that are unique to a certain type of data (cf. e.g. Neunerdt et al., 2013 for annotation of social media texts or Westpfahl and Schmidt, 2013 for enrichment of spoken German).

The POS annotation of our data is created by the Tokenizer, using the latest version of the TreeTagger (Schmid, 1994) working with the Stuttgart-Tübingen TagSet (STTS: Schiller et al., 1999). At the moment the annotation can be called in two different modes: either post mode or direct mode.

In post mode, all tokens occurring in all final versions of the TTs are first annotated, creating an experiment-specific list of possible tokens along with their corresponding POS tags. Then, after the Tokenizer has emulated the entire Translog recording, the tokens found in the intermediate versions are matched against this experiment-specific list. If an intermediate version token can be found in this list, then a reference to the corresponding POS tag is saved with the token, as shown in Figure 4.I for the tokens *Ein* 'a' and *Blatt* 'sheet'. If no match is found, the Tokenizer searches for a POS tag that poses the closest match to the token string by using the Levenshtein distance (Levenshtein, 1966) with a set maximum distance.

In the direct mode the TreeTagger is called each time the text is modified (i.e. if a modifying event is detected). The Tokenizer creates an array containing all words in the current text adjusted to match the requirements of the TreeTagger, which does not allow spaces or any of several other special characters like “, / or line feeds. The data returned by the TreeTagger is modified in a way that allows it to match the provided tags to the tokens that formed the current text version, cf. Figures 2 and 5. Thus, each token has a list of POS tags and each POS tag has a reference to the event that led to its existence. A new POS tag is only added to the list if it differs from the previous element in the list. Figure 4 illustrates this process as the word classes (e.g. Noun [NN] → Separated verb particle [PTKVZ] → Article [ART]) of the tokens change over the course of their creation.

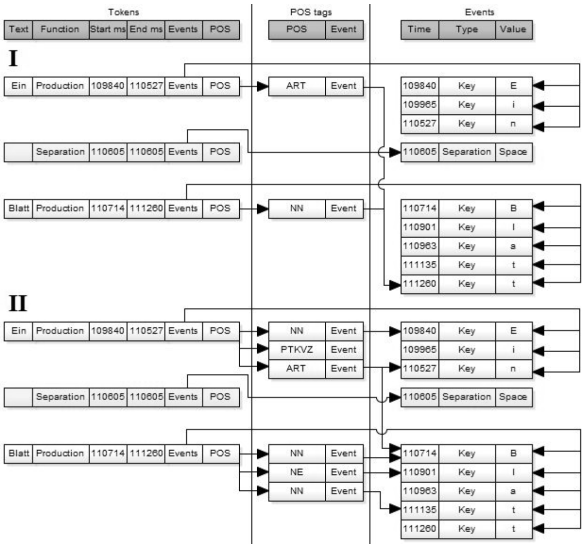


Figure 4. The created tokens, POS tags and events.

Figure 5 shows the events used to create the data presented in 4.II after time-stamp 111260. The tokens *Ein* ‘a’ and *Blatt* ‘sheet’ both have a list of POS tags that expanded during the creation of the TT. The additional reference to the creating event inside the POS tag makes it possible for the user to search in both directions: from the POS tag to the event which led to its creation, as well as from an event to all the POS tags created by this event. For example, the production event ‘B’ with the timestamp 110714 is referenced in multiple POS tags and marks the change of *Ein* from a separated verb particle [PTKVZ] into an article [ART] and the creation of *Blatt* as a noun [NN].

Timestamp	**Text**
109840	E[NN]
109965	Ei[NN]
110527	Ein[PTKVZ]
110605	Ein [PTKVZ]
110714	Ein[ART] B[NN]
110901	Ein[ART] Bl[NE]
110963	Ein[ART] Bla[NE]
111135	Ein[ART] Blat[NN]
111260	Ein[ART] Blatt[NN]

Figure 5. Change of POS tags in the creation of the TT.

Each of these POS modes has advantages over the other. The post mode successfully eliminates false positive matches that occur in the direct mode like *Ei* ‘egg’ [NN] at 109965 as seen in Figure 5, which does not make any sense in the given text. This disadvantage of the direct mode is connected to lower reliability in assignment of certain POS tags. For instance, in Figure 5 the intermediate token *Ein* is tagged as a separated verb particle, even though the [ART] tag is more plausible taking into account the general frequency of the relevant elements. At the same time, the direct mode has the advantage of preserving the references to the point in time at which a POS tag was matching the token. The major disadvantage of the post mode is that it is limited to words that appeared in a TT – but not every word does. For example, if a word like *Papierblatt* ‘paper sheet’ is created in an intermediate version but is always changed to *Blatt Papier* ‘sheet of paper’ there will be no matching tag for *Papierblatt* in the precompiled TT corpus. The direct mode, on the other hand, can assign a [NN] tag to the token *Papierblatt.*

As the first step in evaluating the accuracy of the POS enrichment we looked at the post mode annotation of the data from six participants. A participant produced on average 297 tokens (whereby all token modifying events except for spaces were counted). An average of 73% of these tokens were exact POS matches; an additional 18% were assigned using Levenshtein distance (a considerable amount of tokens in this group consisted of punctuation marks). The remaining 9% of the tokens did not receive any POS tag (about half of these tokens consisted of a single letter). In terms of using the Levenshtein distance for annotation, we found that, on average, 70% of string matches and their POS tags could be considered contextually correct. Next steps will include evaluation of the direct mode and, where necessary, manual correction of tag assignments.

The open design of the Tokenizer and the data structure ensure that files from other keystroking logging systems can be easily added and compared with each other, independent of the origin. Furthermore, additional POS or grammatical tagging tools can be integrated easily within the process.

4 Initial analysis of annotated translation revisions

The keystroke logged translation corpus enriched with information on intermediate word tokens and parts of speech can be used to investigate translation strategies employed during the translation process. These strategies are reflected through revisions of the unfolding target text which exhibit, for instance, alternative lexical choices for the same slot in a sentence, or the choice of different syntactic structures. Such types of revisions can be performed to correct or further refine the target texts (Malkiel, 2009a). Previous research has suggested that revisions of the target text can be considered as one of the indicators of difficulties encountered during the translation process (Dragsted, 2012: 86). In other words, the place and type of corrections, among other measures, can be used to operationalize the difficulty, i.e. the amount of cognitive effort, involved in a translation of certain linguistic features.

In this study we adopt a bottom-up perspective and look at cases where multiple attempts at translating the same source text word have been identified. Previous investigations of such self-corrections have typically relied on time-consuming manual analyses of the keystroke logs[4] using either the replay function (Malkiel, 2009a) or visualization of the data (Dragsted, 2012) illustrated in Figure 6, where the symbol '•' represents the space key and '◄' stands for backspace. Our pre-processing of the data (cf. section 3) helps us to identify multiple attempts belonging to the same translation event automatically, which will facilitate subsequent quantitative analysis.

erhöte•i◄◄◄◄hte•e◄Energiespeicherung

Figure 6. Linear representation in Translog II.

An explorative examination of our data shows various types of revision. One of these is lexical substitution, illustrated through Example 1 above. Malkiel (2009a: 158) observes that more than half of the revisions (excluding changes in spelling) in her data can be attributed to the category of replacing a word or expression with a synonym[5].

Our data sample contains only a few revisions that are very straightforward examples of lexical substitutions where a complete word is typed and then replaced with a different one. In another group of cases, intermediate versions contain incomplete tokens which are deleted and replaced by an alternative, or simply a change in grammatical gender of an article, as is the case in Example 2:

ST Yet the fact that the ball is able…
IT Doch die Tatsache, dass *der*
'yet the fact that the:masc'
TT Doch die Tatsache, dass *die* *Kugel*
'yet the fact that the:fem ball'

Example 2. KLTC, Translator A6.

It is difficult to disentangle alternative text production versions of a string from a simple correction of typing or grammatical errors. Whereas in some cases we could safely assume that the spelling changes were made to correct a typing error (e.g. the string *Pape* changed to *Pap* and then completed to form the word token *Papier* 'paper'), other intermediate versions (as in Example 2) are more ambiguous. Rather than excluding these instances from further analysis, we adopt the notion of target hypotheses. In the context of translation data, target hypotheses refer to several potential plans of the translator for the unfolding target text (Serbina et al., forthc.). This method was originally developed to account for non-standard structures in learner language (Lüdeling, 2008; Reznicek et al., 2013): instead of establishing one of the canonical structures potentially intended by the learner, researchers can formulate several hypotheses that can function in the respective context. During the development of the corpus, the formulation of alternative target hypotheses motivated through the linguistic context of intermediate versions and final target texts allows us to consider possible intentions of the translator, leaving further interpretation of the data to the analysis stage.

Coming back to Example 2, the change from the word token *der* 'the:masc' to the token *die* 'the:fem' can be considered a typing error. This would mean that the translator's plan was to type *die Kugel* 'ball', which appears in the final version, and s/he accidentally typed first the wrong article. However, we can also suggest an alternative target hypothesis, according to which the change from masculine to feminine article form is deliberate. As

[4] See, however, Carl et al. (2010) for an example of an automatic analysis.

[5] It should be, however, noted that in the study by Malkiel (2009a) this group of revisions is rather broad, comprising clarifications (*impossible deadlines* changed into *impossible deadlines to meet*) and modifications in the order of elements (*We once used to* change into *Once, we used to*).

the source text contains *ball*, we might hypothesize that the translator originally planned to use the cognate *Ball* ‘ball’ (requiring the masculine article *der*) but at some point changed to the synonym *Kugel*. The formulation of this hypothesis is additionally motivated by the final target versions of all participants: this instance of the noun *ball* was translated by *Ball* by five out of nine participants. Assuming this target hypothesis, the change of plan could be potentially explained through the wish to avoid cognates, which are more readily accessible than other synonyms but can result in non-idiomatic target language expressions (Malkiel, 2009b).

The POS annotation of the word tokens in the intermediate translation versions can be used to systematically extract all such cases in which one article, or alternatively, an attributive pronoun or adjective is replaced with another. In German, all of these word classes reflect grammatical gender. Therefore, a change in the morphological ending of such an element can hint at a change in translator’s plan (similar to Example 2 discussed above). To identify such cases, we analyzed text parts, where one of the elements mentioned above was altered creating another form of the same word. In these cases, two or more subsequent word tokens tagged as article [ART], a type of an attributive pronoun [PIAT], [PDAT], [PPOSAT], [PRELAT], [PWAT] or an attributive adjective [ADJA] appear in the data, only one of which is preserved in the final version of the translation.

In this step, 49 sequences of tokens meeting the formulated requirements have been extracted. The quantification of examples involving revisions that lead to a production of longer sequences, such as *der weiteren Kompression des Blattes* ‘the further compression of the sheet’ considers the number of nominal slots with which the preceding elements have to agree. In other words, in this particular example, revisions of the initial definite article, the following adjective, both of which agree with the noun *Kompression* ‘compression’, and the second definite article, which agrees with the noun *Blatt* ‘sheet’, are counted as two distinct cases. On the basis of changes in suffixes that were most likely performed to change grammatical gender rather than case or number, 39% (19/49) of the examples distributed across eight keystroke logs were classified as involving several target hypotheses on the level of lexical choices (even though it was not always possible to determine what a potential alternative version was). In one additional case, the experiment participant deleted a part of the produced noun phrase only to retype it. Here it is even less clear whether there was a change of plan or perhaps general uncertainty. The noun *Kugel* ‘ball’ was involved in the revisions most frequently, namely in 32% (6/19) of cases in the data from four different participants. At least in some cases there is good reason to believe that the original plan was to produce its synonym *Ball* (cf. Example 2 above).

While previous studies dismissed all instances of revisions aimed at correcting the spelling of a word (Malkiel, 2009a) and the so-called short-distance revisions, i.e. immediate modifications of the words (Carl et al., 2010), as typing errors, the discussion above shows that there might be more to these types of revisions. We consider the cases described above as examples that can give us additional insights into (possible) translation strategies, which are within reach because of the linguistic annotation of the keystroke logging data.

Until now we have discussed revisions characterized by a mere lexical replacement. In addition, the small data sample examined here contains a few changes of syntactic structures. For instance, one revision has been interpreted as an example of explicitation, named among the properties of translated texts (Baker, 1996). As seen in Example 3, the intermediate translation version is characterized by ellipsis of the head noun within the subject function of the second clause. However, the reference to *Kanten* ‘edges’ is made more explicit later, when the translator inserts the second instance of the noun.

ST these ridges collapse and smaller ones form
IT kollabieren die Kanten und kleinere werden
‘collapse the edges and smaller are’
gebildet
‘formed’
TT kollabieren die Kanten und kleinere *Kanten*
‘collapse the edges and smaller edges’
werden gebildet.
‘are formed’

Example 3. KLTC, Translator A2.

A small number of revisions involving the level of syntactic structures could be explained taking into account the participant group in question. Previous studies indicated that, in contrast to professional translators, (translation) students tend to concentrate on the level of individual words (Lörscher, 1996: 30; Malkiel, 2009a: 161), trying primarily to solve problems connected to lexical choices (Lörscher, 1996: 30-31). Therefore, once the analysis of intermediate versions is extended to include experiments with professional translators, we expect to find more complex revisions related to larger stretches of text.

This initial investigation of our sample data has indicated the benefits of the available enrichment of intermediate translation versions. Using this annotation, we are now able to systematically extract a specific group of cases which potentially reflect a change in translation plan. Formulation of several alternative target hypotheses enables us to stay objective by indicating a range of possibilities that exist during the translation process. If we adopt the hypotheses according to which the changes in suffixes observed in the data reflect modifications in translators' plans, the translation of the nouns following the revised premodifiers likely pose additional cognitive effort for the participants of the experiment. It is certainly necessary to keep in mind that not all of changes in plan are visible as "traces in the typing data in the form of corrections" (Dragsted, 2012: 95). However, automatic identification of changes during the translation process that result in different parts of speech may give us additional clues as to the intentions of the translators.

5 Outlook

Further development of the Tokenizer will address special cases in which the tool identifies a large number of children tokens in the intermediate versions that do not represent additional value to the researcher. These production tokens are generated when the translator types a larger chunk of text without using a separation character (e.g. space) to separate the new word from an existing word token. In the current version of the Tokenizer, the token immediately preceding the inserted material functions as a parent token for all of the inserted characters that are immediately attached to it. A solution can be an automatic identification of these cases that would facilitate their resolution, i.e. chunking into more meaningful word tokens.

Until now the target hypotheses have been generated based on a manual inspection of the data. But to effectively manage larger volumes of data, it is possible to partly automatize the annotation procedure by taking into account the range of translations available for any given source text item in the final translations of all experiment participants (cf. Koehn, 2009 for a similar approach in machine translation). This step requires alignment on different linguistic levels created between originals and the corresponding translations, both final and intermediate versions. Once the alignment links are available, automatic generation of a list of likely target units is planned.

Moreover, as mentioned above, we intend to apply the pipeline of pre- and post-processing steps described in this paper to larger collections of data, in particular to study the revision strategies of professional translators. Based on larger samples of revisions involving changes in syntactic structures, it will be possible to develop queries similar to the one discussed above for further types of modifications using the POS annotation available for intermediate translation versions. This, in turn, is a prerequisite for a quantitative study across several participants. The results on revisions could then be linked to the available eye-tracking data to get further insights into the cognitive processing during the process of translation.

The annotation procedures discussed in the present paper are not limited to the analysis of translation data. Since translation logs involve noncanonical features, the described methods can be generalized to other types of non-standard language found, for instance, in computer-mediated communication or spoken data. Moreover, a quantitative analysis of features present in the intermediate translation versions contributes to identification of effective translation strategies that can be applied in machine translation.

Acknowledgments

The work on the Tokenizer was performed within the HumTec Boostfund project *e-cosmos*, funded by the Excellence Initiative of the German State and Federal Governments.

The data used for analysis was generated within the German Research Foundation (DFG) project *TRICKLET* (Translation Research in Corpora, Keystroke Logging and Eye Tracking), research grant no. NE1822/2-1.

References

Alves, Fabio and Célia Magalhaes. 2004. Using small corpora to tap and map the process-product interface in translation. *TradTerm,* 10: 179–211.

Alves, Fabio and Daniel Couto Vale. 2009. Probing the unit of translation in time: Aspects of the design and development of a web application for storing, annotating, and querying translation process data. *Across Languages and Cultures*, 10(2): 251–273.

Alves, Fabio, Adriana Pagano, and Igor da Silva. 2010. A new window on translators' cognitive activity: Methodological issues in the combined use of eye tracking, key logging and retrospective protocols. In Methodology, technology and innovation in translation process research: A tribute to Arnt Lykke Jakobsen, Inger M. Mees, Fabio Alves, and Susanne Göpferich, editors. Frederiksberg, Copenhagen, pages 267–91.

Alves, Fabio and Daniel Couto Vale. 2011. On drafting and revision in translation: A corpus linguistics oriented analysis of translation process data. *Translation: Computation, Corpora, Cognition,* 1: 105–122.

Baker, Mona. 1996. Corpus-based translation studies: The challenges that lie ahead. In *Terminology, LSP and translation: Studies in language engineering in honour of Juan C. Sager*, Harold Somers, editor. Benjamins, Amsterdam, pages 175–186.

Carl, Michael. 2012a. The CRITT TPR-DB 1.0: A database for empirical human translation process research." In *Proceedings of the AMTA 2012 Workshop on Post-Editing Technology and Practice*: 9–18.

Carl, Michael. 2012b. Translog - II: A program for recording user activity data for empirical reading and writing research. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*: 4108–4112.

Carl, Michael, Martin Kay, and Kristian T.H. Jensen. 2010. Long-distance revisions in drafting and post-editing. *Proceedings of CiCling*: 193-204.

Dragsted, Barbara. 2012. Indicators of difficulty in translation: Correlating product and process. *Across Languages and Cultures,* 13(1): 81-98.

Göpferich, Susanne. 2008. *Translationsprozessforschung: Stand - Methoden - Perspektiven*. Narr, Tübingen.

Hansen-Schirra, Silvia, Stella Neumann, and Erich Steiner. 2012. *Cross linguistic corpora for the study of translations: Insights from the language pair English-German*. de Gruyter, Berlin.

Heeman, Peter A. and James F. Allen. 1999. Speech repairs, intonational phrases, and discourse markers: Modeling speakers' utterances in spoken dialogue. *Computational Linguistics*, 25(4): 527–571.

Jakobsen, Arnt Lykke. 2011. Tracking translators' keystrokes and eye movements with Translog. In *Methods and strategies of process research: Integrative approaches in translation studies*, Cecilia Alvstad, Adelina Hild, and Elisabet Tiselius, editors. Benjamins, Amsterdam, pages 37–55.

Koehn, Philipp. 2009. A process study of computer-aided translation. *Machine Translation Journal*, 23(4): 241-263.

Levenshtein, Vladimir. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics-Doklady,* 10(8): 707–710.

Lörscher, Wolfgang. 1996. A psycholinguistic analysis of translation processes. *Meta,* 41(1): 26-32.

Lüdeling, Anke. 2008. Mehrdeutigkeiten und Kategorisierung: Probleme bei der Annotation von Lernerkorpora. In: Patrick Grommes and Maik Walter, *Fortgeschrittene Lernervarietäten.* Niemeyer, Tübingen, pages 119–140.

Malkiel, Brenda. 2009a. From Ántonia to My Ántonia. Tracking self-corrections with Translog. In *Behind the mind. Methods, models and results in translation process research.* Susanne Göpferich, Arnt Lykke Jakobsen und Inger M. Mees, editors. Frederiksberg, Samfundslitteratur, pages 149–166.

Malkiel, Brenda. 2009b. When idioti (idiotic) becomes "fluffy". Translation students and the avoidance of target language cognates. *Meta,* 54(2): 309–325.

Neunerdt, Melanie, Michael Reyer, and Rudolf Mathar. 2013. A POS tagger for social media texts trained on web comments. *Polibits*, 48: 61-68.

Reznicek, Marc, Anke Lüdeling, and Hagen Hirschmann. 2013. Competing target hypotheses in the Falko corpus: A flexible multi-layer corpus architecture. In *Automatic treatment and analysis of learner corpus data,* Ana Díaz-Negrillo, editor. Benjamins, Amsterdam, pages 101–123.

Schiller, Anne, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, University of Stuttgart.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. *Proceedings. International Conference on New Methods in Language Processing,* Manchester, UK.

Serbina, Tatiana, Paula Niemietz, and Stella Neumann. Forthcoming. Development of a keystroke logged translation corpus. In *New directions in corpus-based translation studies*, Claudio Fantinuoli and Federico Zanettin, editors. Berlin: Language Science Press.

Vinay, Jean-Paul, and Jean Darbelnet. 1995 (1958). *Comparative stylistics of French and English: A methodology for translation,* Juan C. Sager and M.-J.

Hamel, editors and translators. Benjamins, Amsterdam.

Westpfahl, Swantje, and Thomas Schmidt. 2013. POS für(s) FOLK: Part of Speech Tagging des Forschungs- und Lehrkorpus Gesprochenes Deutsch. *Journal for Language Technology and Computational Linguistics*, 1: 139-153.

Zinsmeister, Heike, Ulrich Heid, and Kathrin Beck. 2014. Adapting a part-of-speech tagset to non-standard text: The case of STTS. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*: 4097-4104.