Preliminary communication
https://doi.org/10.31784/zvr.7.1.7
Received: 15 January 2019
Accepted: 4 February 2019

TRANSLATION ERROR ANALYSIS IN TREAT: A WINDOWS APP USING THE MQM FRAMEWORK

Suzana Majcunić
MA in Informatics, University of Rijeka, Department of Informatics, Radmile Matejčić 2, 51 000 Rijeka, Croatia; e-mail: suzana.majcunic@gmail.com

Maja Matetić
PhD, Full Professor, University of Rijeka, Department of Informatics, Radmile Matejčić 2, 51 000 Rijeka, Croatia; e-mail: majam@uniri.hr

Marija Brkić Bakarić
PhD, Assistant Professor, University of Rijeka, Department of Informatics, Radmile Matejčić 2, 51 000 Rijeka, Croatia; e-mail: mbrkic@uniri.hr

ABSTRACT

The aim of this research paper is to conduct a thorough analysis of inter-annotator agreement in the process of error analysis, which is well-known for its subjectivity and low level of agreement. Since the process is tiresome in its nature and the available user interfaces are pretty distinct from what the average annotator is accustomed to, a user-friendly Windows 10 application offering a more attractive user interface is developed with the aim to simplify the process of error analysis. Translations are performed with Google Translate engine and English-Croatian is selected as the language pair. Since there has been a lot of dispute on inter-annotator agreement and the need for guidelines has been often been emphasized as crucial, the annotators are given a very detailed introduction into the process of error analysis itself. They are given a presentation with a list of the MQM guidelines enriched with tricky cases. All annotators are native speakers of Croatian as the target language and have a linguistic background. The results demonstrate that a stronger agreement indicates more similar backgrounds and that the task of selecting annotators should be conducted more carefully. Furthermore, a training phase on a similar test set is deemed necessary in order to gain a stronger agreement.

Key words: *Windows 10 application, error analysis, inter-annotator agreement, machine translation evaluation*

1. INTRODUCTION

Ever since the machine translation (MT) society has made a huge shift from a phrase-based statistical MT to a neural MT (NMT), a lot of research initiatives have focused on translation error types in an attempt to better describe differences between these two approaches. Since the quality evaluation of machine translation is inherently subjective and the quality is context-dependent, there is no single standard in assessing translation quality (Secara, 2001). A single sentence can be translated in many ways. All this makes the quality assessment one of the most debated topics in translation.

Automatic and human evaluations of MT mostly provide quantitative evaluation (Stymne, 2011). Error analysis refers to the identification and classification of individual errors in a translated text (Stymne and Ahrenberg, 2012). It is one of the well-accepted ways to assess translation in qualitative terms as it can point to strengths as well as weaknesses of a certain MT system (Stymne and Ahrenberg, 2012). When analysing or comparing different translation systems, one wants to get answers such as what kind of errors systems make more often, whether one system is superior in any or all aspects, whether a modification brings some improvements though invisible on the score, etc. (Popovic and Burchardt, 2011).

Multidimensional quality framework (MQM) is a result of a thorough investigation of major human and machine translation assessment metrics (Lommel, Uszkoreit and Burchardt, 2014). It can be described as a comprehensive list of quality issue types which serves as a mechanism for declaring specific metrics for quality assessment and error annotation tasks.

Human quality annotations are well-known for their low inter-annotator agreement (IAA) (Lommel, Popovic and Burchardt, 2014). The low IAA might arise due to the disagreement on the severity of an error or on its precise span, but also due to errors which reflect multiple issues. Some authors (Lommel, Popovic and Burchardt, 2014) even create a formal decision tree and improved guidelines to assist with annotation in order to improve IAA. Furthermore, the authors Stymne and Ahrenberg (Stymne and Ahrenberg, 2012) argue that it is possible to get a reasonable agreement either by using a simple error taxonomy or by using a more detailed taxonomy and a set of guidelines.

There is a whole line of research comparing NMT to SMT with the aim of detecting its strengths or weaknesses. As expected, there is not much research in this area involving Croatian. A detailed human analysis of the outputs produced by NMT and PBMT systems when translating news texts in the English-to-Croatian language direction is performed in Kubička (Klubička, 2017). Errors are annotated according to a detailed error taxonomy relevant to the problematic linguistic phenomena of the language pair and compliant with the hierarchical listing of issue types defined as part of the MQM.

Compared to Kubička (Klubička, 2017), the number of annotators in this paper is doubled, i.e. there are four annotators. However, their background is less similar. Three out of four annotators are final-year master students. The IAA is calculated on three levels: the sentence level irrespective of the error type, number, and position; the sentence level irrespective of the error type and position;

and error type. Since the task of evaluation or annotation is generally often given to students, this paper puts focus on checking their suitability by comparing IAA scores with respect to their formal education background. Please note that the analysis is less detailed compared to Kubička (Klubička, 2017), which should result in higher IAA scores.

The paper is organized as follows. In the next section the methodology is described. It is followed by results. The discussion is provided in section 4. Section 5 concludes the paper.

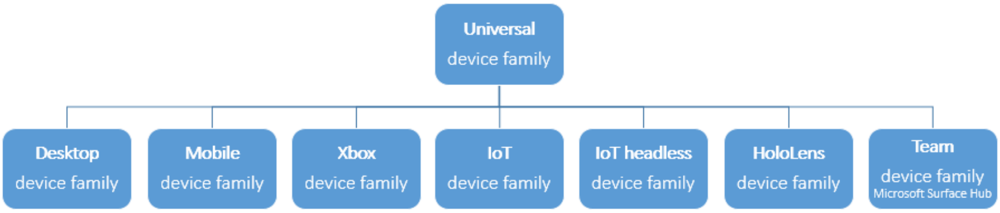
2. METHODOLOGY

The following section is divided into two subsections. In the first subsection the development of the tool used for annotation is outlined, and its user interface and features are presented, and in the second subsection the experimental study is described.

2. 1 TREAT application

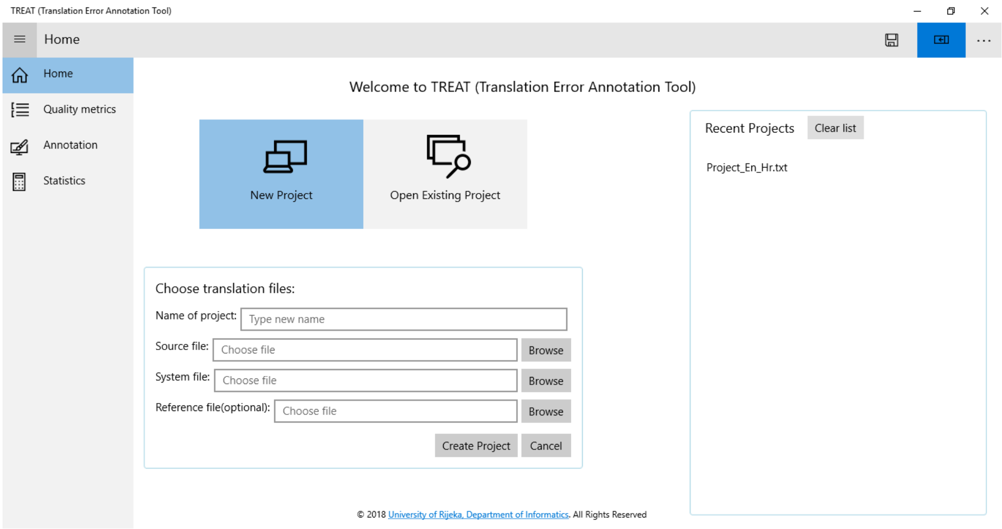
Since annotators do not usually have advanced computer skills, the first goal set for this study is to create a friendly working environment to which they are accustomed to. The app should support displays of any size and orientation so that the annotators can do their job anytime and anywhere. Universal Windows Platform (UWP) is chosen as the new open-source and cross-platform framework for building applications for all operating systems, including Windows, Mac, and Linux. UWP apps work on all Windows 10 devices (Figure 1). .NET Core is chosen as the application programming interface (API). The programming language used is C#. Graphical part of the application is implemented in the declarative eXtensible Application Markup Language (XAML), which is also developed by Microsoft. The name of the app is TREAT (TRanslation Error Annotation Tool). The user interface is shown in Figure 2.

Figure 1. Windows device family overview



Source: *Microsoft* (2018)

Figure 2. TREAT user interface



TREAT (Translation Error Annotation Tool)

Welcome to TREAT (Translation Error Annotation Tool)

Choose translation files:

Name of project: Type new name

Source file: Choose file

System file: Choose file

Reference file(optional): Choose file

Recent Projects

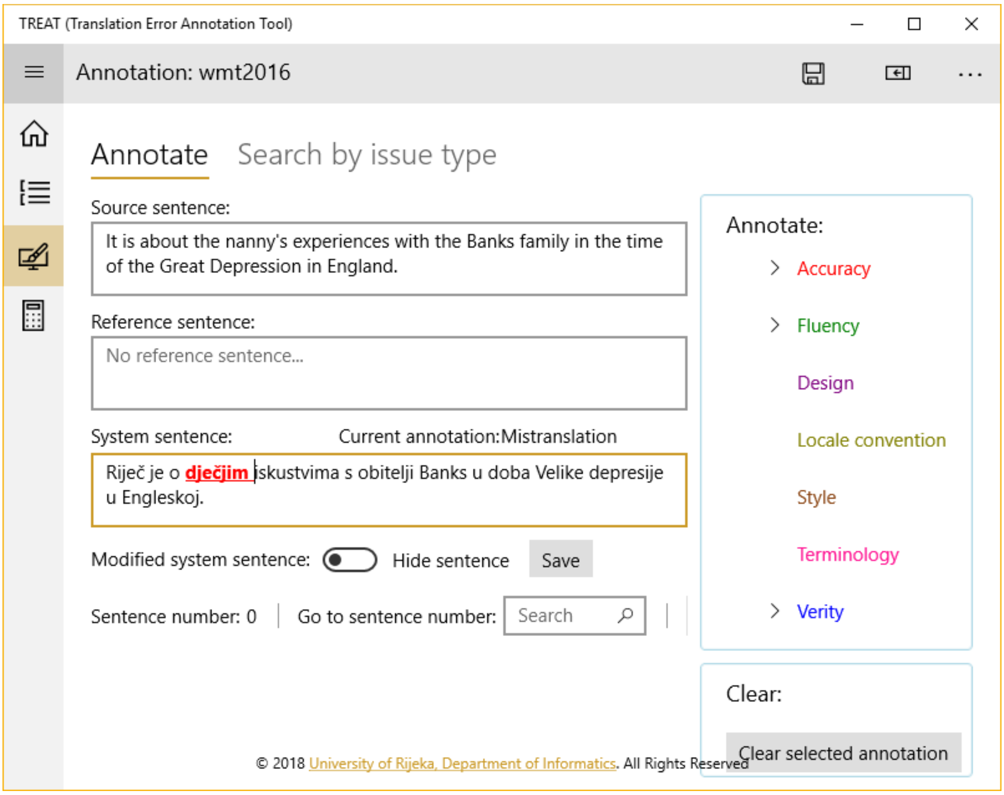
Project_En_Hr.txt

© 2018 University of Rijeka, Department of Informatics. All Rights Reserved

Since using older versions of Microsoft Visual Studio on certain platforms that precede Windows 10 has some limitations, the development is done on Windows 10 platform. The integrated development environment the app is built in, is the Microsoft Visual Studio Enterprise 2017 (version 15.7.4).

The app has a built-in support for MQM hierarchy. Since the tree-view control is used for the purpose, the minimum Windows version the app can run on is 1083. The annotation process in TREAT is shown in Figure 3. The input to the annotation process are two textual files – the source file and the target file. Additionally, the file with reference translations can be provided. If the annotation project has already been created, then the project file can be used as input.

Figure 3. Annotation in TREAT



Source sentence:
It is about the nanny's experiences with the Banks family in the time of the Great Depression in England.

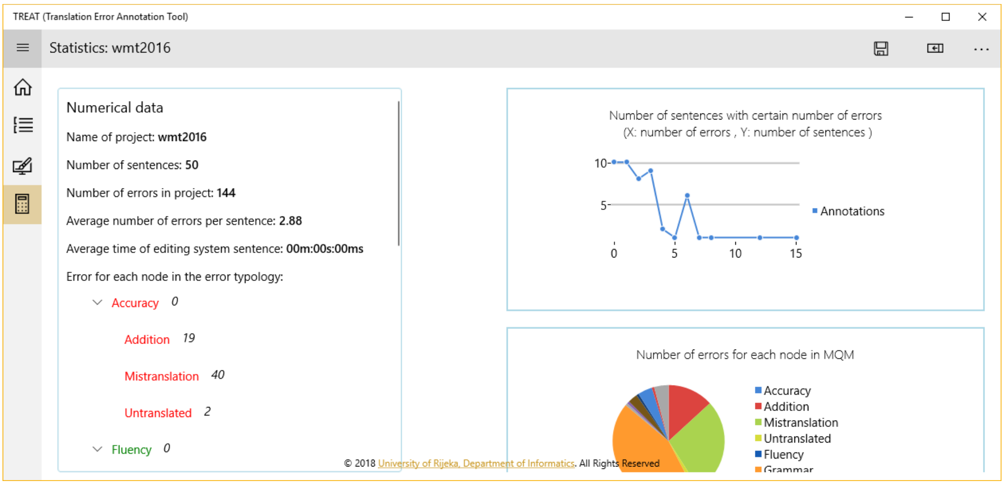
Reference sentence:
No reference sentence...

System sentence: Current annotation:Mistranslation
Riječ je o **dječjim** iskustvima s obitelji Banks u doba Velike depresije u Engleskoj.

Sentence number: 0

Basic features of the developed app are listed below: MQM hierarchy can be easily modified (new nodes can be added, nodes can be excluded from evaluation, etc.) a detailed statistics of annotated issue types can be generated and basic visualizations of statistical data can be generated (Figure 4) statistics can be saved in a textual file (Figure 5).

Figure 4. Statistics and visualizations in TREAT



TREAT (Translation Error Annotation Tool)

Statistics: wmt2016

Numerical data

Name of project: **wmt2016**

Number of sentences: **50**

Number of errors in project: **144**

Average number of errors per sentence: **2.88**

Average time of editing system sentence: **00m:00s:00ms**

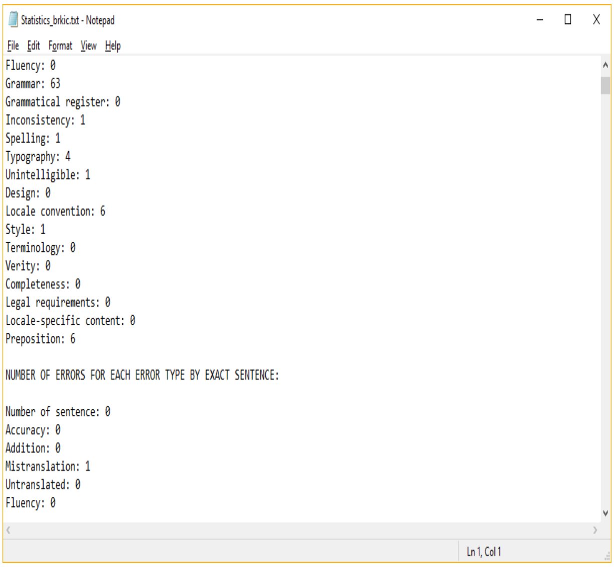
Error for each node in the error typology:

- Accuracy 0
  - Addition 19
  - Mistranslation 40
  - Untranslated 2
- Fluency 0

© 2018 University of Rijeka, Department of Informatics. All Rights Reserved

Number of sentences with certain number of errors
(X: number of errors , Y: number of sentences)
10
5
0
5
10
15
Annotations
Number of errors for each node in MQM
Accuracy
Addition
Mistranslation
Untranslated
Fluency
Grammar

Figure 5. An excerpt from the file with statistics



```
Fluency: 0
Grammar: 63
Grammatical register: 0
Inconsistency: 1
Spelling: 1
Typography: 4
Unintelligible: 1
Design: 0
Locale convention: 6
Style: 1
Terminology: 0
Verity: 0
Completeness: 0
Legal requirements: 0
Locale-specific content: 0
Preposition: 6

NUMBER OF ERRORS FOR EACH ERROR TYPE BY EXACT SENTENCE:

Number of sentence: 0
Accuracy: 0
Addition: 0
Mistranslation: 1
Untranslated: 0
Fluency: 0
```

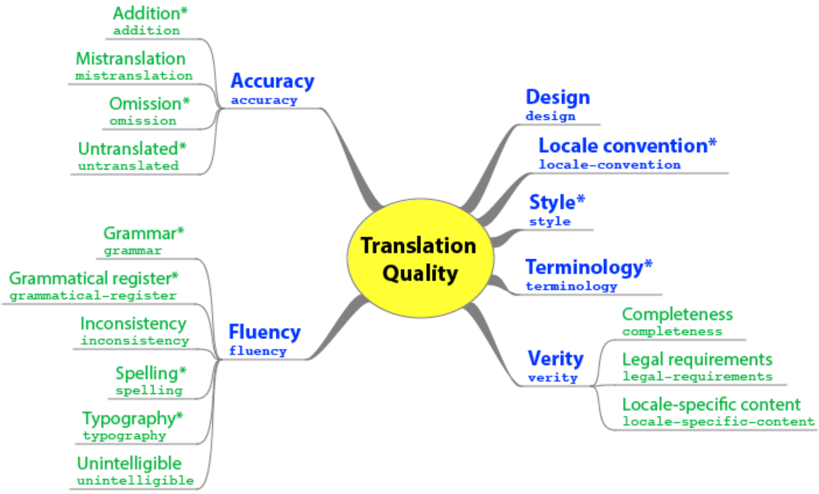
2.2 Experimental study

The following subsection is divided into five parts, which give the details on the MQM framework, the annotators, the test set, the official MQM annotation guidelines, and the inter-annotator agreement scores. Google Translate NMT engine is used for obtaining Croatian translations of English sentences which are then annotated by human annotators and the obtained annotations are used for calculating inter-annotator agreement scores.

2. 2. 1 MQM framework

MQM defines over 100 issue types. The term *issue* is used to refer to any potential error detected in a text. At the top level there are 10 categories: *accuracy, design, fluency, internationalization, locale convention, style, terminology, verity, compatibility,* and *other*. Since it would not be viable to perform the annotation process using the full MQM tag set, it is necessary to choose a smaller subset in accordance with research questions. Naturally, starting with the core tag set is considered to be a good practice (Figure 6).

Figure 6. MQM core



Source: *MQM definition* (2015)

Since the focus of this research is on testing the software and investigating IAA in relation to linguistic background, the annotators are instructed to use only the MQM core.

While issue types such as addition, mistranslation, omission, untranslated or spelling should be pretty clear, short clarifications are provided for the remaining issues present in the MQM core. Grammar should be applied to all the grammar or syntax related issues of the text, other than spelling and orthography. Maybe somewhat counter-intuitive, when the content uses wrong pronouns or verb forms when their formal counterparts are required or vice-versa, grammatical register should be annotated and not style. Consistency issue is detected by reading the text with understanding. For example, the text states one fact or certain instructions in one place and something contradictory in the other. Issues related to the mechanical presentation of a text (punctuation is used incorrectly or a text has an extraneous hard return in the middle of a paragraph) should be annotated as typography. This category should be used for any typographical errors other than spelling. If the exact nature of the error cannot be determined and a major break down in fluency occurs, unintelligible mark-up should be applied. An example of issue annotated as locale convention might be using a comma (,) instead of a period (.) as a thousands separator.

An example of a style issue is when the source text is light and humorous, while the target one is serious and formal in style. Terminology should not be used if a text is simply mistranslated but it should be used in cases where the translation could be a valid translation of the source if certain terminology is not mandated. Since Design refers to incorrect formatting, and the annotators were provided with source and target sentences in textual format, this issue category might have been easily excluded from the annotation process. Verity issue is triggered when the text makes statements that contradict the world of the text, i.e. the text says a feature is present on a certain model of automobile (because it is true in the source locale) when in fact it is not available in the target locale. Completeness can be contrasted with omission as it refers to instances in which needed content is missing in the source language, while omission refers to instances in which content present in the source language is not present in the translation. Cases in which the source text does not meet legal requirements are generally critical errors that will require rewriting the source text. An example might be the translation of some notice which should have been replaced according to the instructions and not merely translated. An example of a locale-specific content might be a text in a manual which describes a feature not available in the product on the target market. As expected, due to the nature of the source text, no issues under the category Verity have been detected.

Since the annotations for Omission were not uniformly laid (i.e. some annotators annotated source words as omissions and some annotated spaces in the target as omissions), this issue subtype has been excluded from our analysis. Reference sentences have not been provided.

2. 2. 2 Annotators

Annotators are all native speakers of Croatian. One of the annotators has a BA degree in English language (E1), two in German (E2, E3), and one has a Master degree in English (E4). They all have a BA degree in Informatics and the first three are on their final year of Master studies. All the annotators are very confident of their knowledge of English language. Additionally, student annotators are awarded additional credits for the completion of the task within the elective course on translation tools. The possibility of gaining scores ought to make annotators more perceptive, cautious, and adherent to the guidelines.

2. 2. 3 Test set

Our test set consists of the randomly selected sentences of the English test set of 2016 news translation shared task at WMT[1]. The annotators annotated 50 sentences. The size of the test set is limited to 50 sentences due to several reasons. The first reason is that the annotation process is extremely time-consuming so it would be too much to ask annotators to annotate more sentences under the volunteering framework. The second reason is that annotators might become too tired and thus less attentive under annotation overload. They were only presented with the English source text, and its respective GT output[2]. The reader should be aware of the fact that GT output

[1] http://www.statmt.org/wmt16/translation-task.html

[2] obtained December 6, 2018

evolves with time and that the translations obtained only a month later differ greatly and are of seemingly much better quality.

2. 2. 4 Guidelines

Prior to annotation, the annotators were familiarized with the TREAT software and the official MQM annotation guidelines, which offer detailed instructions for annotation within the MQM framework[3]. These guidelines include the decision tree. The decision tree is organized in such a way that it eliminates specific issues before moving to general ones. Some of the guidelines are outlined here to familiarize the reader with the annotation process.

- Always select the level about which you are most certain – do not guess!
- If multiple types apply, select the first one that the decision tree guides you to.
- Annotate only the words that constitute an error, even if there are other words in between.
- If correcting one error will eliminate other errors, tag only that error.

2. 2. 5 IAA

After the process of annotation, IAA is calculated using Cohen's kappa (κ) (Cohen, 1960). The Cohen's Kappa is given in (1), where p_o refers to the relative observed agreement among annotators or the accuracy, and p_e refers to the expected agreement. Since Cohen's Kappa is appropriate only for pair-wise comparisons, the similarity between each pair of annotators is evaluated separately. The authors Lommel, Popovic and Burchardt (Lommel, Popovic and Burchardt, 2014) additionally take the average score as the final measure.

$$\kappa = \frac{p_o - p_e}{1 - p_e} \qquad (1)$$

IAA scores are calculated at the sentence level, meaning that positions or the exact tokens annotated as errors are disregarded. As Lommel (Lommel, Popovic and Burchardt, 2014) point out, though annotators can agree that a sentence contains the same issue, they might disagree on the span that the issue covers. In the first calculations only error as the most general category is taken into account or whether there is an agreement that error exists at all. This agreement is labelled 'Any errors', similarly to the authors in Klubička (Klubička, 2017). A refined score is calculated by taking into account the number of errors detected in the sentence. Afterwards, calculations are performed for every error type separately. All three level calculations exclude error positions. This means that the scores can be either equal to the scores which take positions into account or overestimates. Since Cohen's Kappa is appropriate only for pair-wise comparisons, Fleiss' kappa is also calculated (Fleiss, 1971). The value p_e is the sum of squares of the proportions of all assignments made to each category. In order to calculate p_e, first the extent to which annotators agree is calculated for each

[3] A decision tree provided to aid the annotation process can be found at http://www.qt21.eu/downloads/annotatorsGuidelines-2014-06-11.pdf.

individual sentence according to (2), where *n* stands for the number of annotators and *k* for the number of categories, and then the average is taken.

$$P_i = \frac{1}{n(n-1)}\left[\left(\sum_{j=1}^{k} n_{ij}^2\right) - n\right] \tag{2}$$

3. RESULTS

Table 1 shows the distribution of issue types per annotator.

First Cohen's kappa is calculated pairwise by just taking into account whether a sentence has been annotated as erroneous or not (labelled 'Any error' as mentioned previously). The agreement on the number of errors per sentence is also calculated. The IAA scores are shown in Table 2. The darker shade of grey colour indicates stronger IAA. The asterisk (*) next to the annotators' labels stands for Fleiss' kappa.

Table 3 gives IAA scores per issue type. *Terminology*, *Style* and *Grammatical Register* are omitted, as these annotations seem to be tied to a specific annotator.

4. DISCUSSION

Taking a closer look at the IAA results, it can be concluded that despite very precise guidelines and getting additional credits for the task, solely linguistic background is not enough to get a substantial or almost perfect IAA. According to Landis (Landis and Koch, 1977), 0–0.2 represents slight, 0.2–0.4 fair, 0.4–0.6 moderate, 0.6–0.8 substantial, and 0.8–1.0 almost perfect agreement. The 'Any errors' value represents agreement on whether there are errors in a given sentence, while 'Number of errors' represents agreement on the number of errors recorded per sentence.

The numbers of errors recorded per annotator differ greatly (Table 1). If the annotator with the highest number of errors recorded is taken as the base, it is evident that the remaining three annotators detect only between 72 to 88% errors. It is also interesting that one annotator has many doubts as far as the issues under the accuracy category are concerned and therefore, reluctant to make guesses, used the superordinate category level, while others have no such doubts.

Table 1. The distribution of issue types found per annotator

	E1	E2	E3	E4
Accuracy	12	0	0	0
Addition	2	11	3	19
Grammar	22	42	64	69
Grammatical register	13	0	0	0
Inconsistency	27	0	0	1

Locale convention	1	0	0	6
Mistranslation	28	43	47	40
Spelling	1	0	0	1
Style	0	0	0	1
Terminology	0	0	5	0
Typography	1	0	0	4
Unintelligible	0	0	1	1
Untranslated	7	7	6	2
TOTAL	114	103	126	144

Table 2. ‘Any error’ and the number of errors IAA κ scores

	Any error	Number of errors
E1-E2	0.42	0.27
E1-E3	0.41	0.33
E1-E4	**0.63**	**0.52**
E2-E3	0.45	0.39
E2-E4	0.42	0.1
E3-E4	0.41	0.2
E1-E2-E3-E4	0.46	0.21

Table 3. IAA per issue type

	Accuracy	Addition	Grammar	Inconsist.	Mistransl.	Locale conven.	Spelling	Typography	Unintellig.	Untransl.
E1-E2	0	0.41	0.21	0	0.51	0	0	0	-	0.43
E1-E3	0	-0.05	0.24	0	0.46	0	0	0	0	0.49
E1-E4	0	0.21	0.32	0.12	0.57	-0.03	-0.02	-0.03	0	0.20
E2-E3	-	0.34	0.45	-	0.23	-	-	-	0	0.29
E2-E4	-	0.51	0.44	0	0.56	0	0	0	0	0.20
E3-E4	-	0.17	0.60	0	0.36	0	0	0	1	-0.06
E1-E2-E3-E4*	-0.29	0.24	0.22	-0.02	0.24	-0.03	-0.01	-0.02	0.33	0.29

In the work of Klubička (Klubička, 2017), the κ scores are relatively consistent across all error types for each analysed system, mostly ranging between 0.35 and 0.55. The obtained IAA is mostly fair for the 'number of errors', and moderate for the 'any error' scenario. Given the lowered complexity of the annotation schema, higher scores have been expected. However, it seems that scores encode annotators' backgrounds. A substantial agreement exists between the two annotators who graduated or will graduate English in the 'Any error' category. If the number of errors annotated per sentence is taken into account, the agreement is lower, but still the highest for E1 and E4. For most IAA tasks, agreement of at least 0.85 is required for a measure to be considered reliable.

The discussion is restricted to the obtained agreements on the four issue subtypes for which each individual annotator has spent at least 1% of his or her annotations – *addition*, *grammar*, *mistranslation*, and *untranslated*. Although *fluency* can be judged by looking only at the target sentence, the agreement in the constituting issue sub-type *grammar* is surprisingly low, i.e. from fair to moderate, with the exception of agreement between E3 and E4 which is substantial. The agreement in the category *mistranslation* is the highest for annotators E1 and E4, although very close to the agreement between E2 and E4. Although these pairwise agreements reach moderate and sporadically substantial agreements, Fleiss' kappa scores are fair for all but 'any error' category.

Since the agreement on the *untranslated* content is not strikingly high as one would expect, a manual analysis of the sentences is conducted. It turns out that as far as *untranslated* content is concerned, annotators mostly disagree on proper nouns. Also, untranslated word to which a suffix is added in accordance with target language grammar is treated by one of the annotators as *unintelligible* and by others as *untranslated*.

A particular attention is paid to the sparse annotation types. It can be concluded that it is worth investigating whether even intra-annotator agreement would reach substantial level. For example, a period missing after the year has once been annotated as locale convention and some other time as typography by the same annotator. By taking a closer look at the explanations given previously, the annotator's confusion is justified. First the annotators are instructed to annotate issues related to the mechanical presentation of the text (punctuation is used) as *typography*, and later on using a comma (,) instead of a period (.) as a thousands separator is provided as an example of *locale convention* issue. Although some errors stem from incomplete or confusing instructions, some can be attributed to the lack of attention. Some annotators fail to detect some issues (e.g. ft as the unit of measure), morphologically wrong term has been identified as misspelling by one of the annotators, style annotation has been applied to one pronoun, although instructions explicitly say that such cases should be annotated as grammatical register, etc.

5. CONCLUSION

Annotators did not report any bugs or problems with the developed application. However, during the result analysis phase it became obvious that some features had to be added. The main drawback was that one had to use additional tools and filters on the project files for more in-depth analysis which included calculating Cohen's and Fleiss' kappa scores. It would be extremely useful if the scores could be obtained from the very app. The first feature should enable reading in two project files and calculating Cohen's kappa scores per error as well as per error type on a sentence basis. The second should enable reading in any number of project files and calculating Fleiss' kappa score on the same basis.

More importantly, the results presented in this paper raise concerns about employing students as annotators since it is difficult to measure their background knowledge and select those with similar knowledge levels. Although one might conclude that not even awarding additional credits motivates annotators enough to read guidelines and study examples, it seems plausible that guidelines and examples are too simplistic and that annotators lack training on a set of sentences compiled from the same source as the test set. An evaluation filter might also be employed which would correct annotations of less competent evaluators. In that way the annotation process might be conducted in two passes – first pass by all the annotators and then second pass by the 'super-annotator', i.e. someone who is more competent or has the best score on the test taken by annotators. To conclude, the goal of the analysis presented in this and similar papers should not be merely pursuing high IAA but aim at obtaining high-quality annotations as these affect the future of MT. In future researches a similar test set with referent annotations will be prepared and a training on these real-world examples will be conducted prior to the annotation task. A suggested two phase annotation procedure will also be further investigated.

ACKNOWLEDGEMENT

This research has been supported by the University of Rijeka under Grant No. 17.14.2.2.01.

LITERATURA

Cohen, J. (1960) 'A Coefficient of Agreement For Nominal Scales', Educational and Psychological Measurement, 20(1), pp. 37-46. https://doi.org/10.1177/001316446002000104

Fleiss, J. L. (1971) 'Measuring nominal scale agreement among many raters', Psychological Bulletin, 76(5), pp. 378-382. https://doi.org/10.1037/h0031619

Klubička, F. (2017) 'Fine-grained Human Evaluation of an English to Croatian Hybrid Machine Translation System', (July). https://doi.org/10.1515/pralin-2017-0014

Landis, J. R. and Koch, G. G. (1977) 'The Measurement of Observer Agreement for Categorical Data Data for Categorical of Observer Agreement The Measurement', Biometrics, 33(1), pp. 159-174. https://doi.org/10.2307/2529310

Lommel, A., Popovic, M. and Burchardt, A. (2014) 'Assessing Inter-Annotator Agreement for Translation Error Annotation Assessing Inter - A nnotator Agreement for Translation Error Annotation', in MTE: Workshop on Automatic and Manual Metrics for Operational Translation Evaluation.

Lommel, A., Uszkoreit, H. and Burchardt, A. (2014) 'Multidimensional Quality Metrics : A Flexible System for Assessing Translation Quality', 12(Tradumàtica 12), p. 455-463. https://doi.org/10.5565/rev/tradumatica.77

Microsoft (2018). Available at: https://docs.microsoft.com/.

MQM definition (2015). Available at: http://www.qt21.eu/.

Popovic, M. and Burchardt, A. (2011) 'From Human to Automatic Error Classification for Machine Translation Output', (May). https://doi.org/10.2478/v10108-011-0011-4

Secara, A. (2001) 'Translation Evaluation - a State of the Art Survey', pp. 39-44.

Stymne, S. (2011) 'B LAST : A Tool for Error Analysis of Machine Translation Output', (June), pp. 56-61.

Stymne, S. and Ahrenberg, L. (2012) 'On the practice of error analysis for machine translation evaluation', pp. 1785-1790.

Prethodno priopćenje
https://doi.org/10.31784/zvr.7.1.7
Datum primitka: 15. 1. 2019
Datum prihvaćanja: 4. 2. 2019

ANALIZA POGREŠAKA U PREVOĐENJU U APLIKACIJI TREAT: WINDOWS APLIKACIJI KOJA SE TEMELJI NA MQM OKVIRU

Suzana Majcunić
Mag. inf., Sveučilište u Rijeci, Odjel za informatiku, Radmile Matejčić 2, 51 000 Rijeka, Hrvatska;
e-mail: suzana.majcunic@gmail.com

Maja Matetić
Dr. sc., redoviti profesor, Sveučilište u Rijeci, Odjel za informatiku, Radmile Matejčić 2, 51 000 Rijeka, Hrvatska; *e-mail*: majam@uniri.hr

Marija Brkić Bakarić
Dr. sc., docent, Sveučilište u Rijeci, Odjel za informatiku, Radmile Matejčić 2, 51 000 Rijeka, Hrvatska;
e-mail: mbrkic@uniri.hr

SAŽETAK

Cilj rada je izvršiti temeljitu analizu slaganja među označivačima u postupku analize pogrešaka koji je poznat po svojoj subjektivnosti i niskoj razini slaganja. Budući da je sam postupak po prirodi zamoran, a sučelja dostupnih alata i usluga poprilično se razlikuju od onog na što je prosječni označivač naviknut, u svrhu pojednostavljenja samog postupka analize pogrešaka razvijena je Windows 10 aplikacija s poznatim i atraktivnim korisničkim sučeljem. Englesko-hrvatski prijevodi preuzeti su s usluge Google Translate. Budući da je slaganje među označivačima čest predmet rasprave i da je od neospornog značaja istaknuta potreba za smjernicama, označivačima je dan vrlo detaljan uvid u postupak analize pogrešaka. Također, popis MQM smjernica uz primjere potencijalnih pogrešaka uobličen je u prezentaciju i dan označivačima na raspolaganje. Označivačima je ciljni, tj. hrvatski jezik materinski, a svi imaju određenu razinu lingvističke pozadine. Rezultati otkrivaju da veća razina slaganja ukazuje na sličnije formalno obrazovanje i da proces odabira označivača treba biti pažljivo osmišljen. Štoviše, testiranje na sličnom skupu podataka trebalo bi prethoditi odabiru označivača kako bi se postigla veća razina slaganja.

Ključne riječi: *Windows 10 aplikacija, analiza pogrešaka, slaganje među označivačima, evaluacija strojnog prevođenja*