L-Università ta' Malta

Universidad del País Vasco
Euskal Herriko Unibertsitatea

How does machine translation affect language? Analyzing the effect of machine translation on translated texts.

Author: Jelena Sarajlić

Advisors: Nora Aranberri Monasterio, Claudia Borg

With the support of the Erasmus+ Programme of the European Union

Erasmus Mundus European Masters Program
in Language and Communication Technologies (EM LCT)

Master Thesis

September 2022

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Laburpena

Master-lan honen helburua da aztertzea itzultzaile automatiko neuronalek duten eragina proposatzen dituzten itzulpenetako hizkuntzan, aniztasun eta aberastasun lexikoari, morfologikoari eta sintaktikoari dagokionez. Horretarako, lau itzultzaile neuronal entrenatu dira. Entrenamendua, ebaluazioa eta itzulpen automatikoak egin dira domeinu eta luzera antzeko bi corpus erabilita (bata bereziki lan honetarako sortua). Bi hizkuntza-pare landu dira, noranzko bietan: ingelesa eta gaztelania batetik, ingelesa eta kroaziera bestetik. Aberastasun lexikoari erreparatuta, emaitza gehienek adierazten dute galera maila bat edo beste. Hala ere, metriketako batek aniztasun lexikoaren gehitzea gertatu izana erakusten du. Aberastasun morfologikoari buruz emaitzak ez dira argiak, izan ere, metrika gehienek galera txikia edo galerarik eza adierazten dute, eta, bi kasutan, aberastasunaren igoera. Kategoria gramatikalen eta sintaxiaren distribuzio-analisiari begiratuta, gure emaitzak bat datoz ikerlariek aurretiaz egindako baieztapenekin, hau da, itzultzaile neuronalek maiztasun handiko elementuen agerpenak areagotzen dituzte eta maiztasun gutxikoenenak mugatu.

Abstract

This Master Thesis analyses the effect of neural machine translation on the language of the translation in terms of lexical, morphological, and syntactical diversity or richness. Four neural machine translation models are trained. Two different corpora of similar length and domain, one of which was created in this work, are used to train and evaluate the models, as well as translate text. Two language pairs were used in both directions: English and Spanish; and English and Croatian. Regarding lexical richness, the majority of our results indicate a degree of lexical loss in the translations. One metric shows a gain of lexical diversity in one of the translations. In morphological richness, the results are not as clear, with most of the metrics showing slight to no loss, or even a gain of richness in two of the translations. Part of speech distribution analysis, as well as parse distribution analyses, both seem to confirm claims made by some that neural machine translation systems increase the frequency of most and decrease the frequency of least frequent items.

Contents

List of Figures

List of Tables

Glossary

EN ISO 639-1 code for English. 13

ES ISO 639-1 code for Spanish; Castilian. 13

HR ISO 639-1 code for Croatian. 13

Acronyms

ADJ adjective. 41

ADP adposition. 42

ADV adverb. 42

AUX auxiliary. 42

BPE byte-pair encoding. 13, 14, 17, 19

CCONJ coordinating conjunction. 42

CDU cosine distance of uniform synonym distribution and the actual synonym distribution (metric used for evaluating lexical diversity). 23

D Simpson's Diversity Index (metric used for evaluating grammatical diversity). 25, 27

DET determiner. 42

DGT the corpus created for the purpose of this Master Thesis using DGT-Translation Memory releases 2013-2016, 2018-2020. 12, 14, 32

EP the Europarl corpus. 14

H Shannon Diversity metric used for evaluating grammatical diversity. 25

INTJ interjection. 42

LFP Lexical frequency profile. 22

MTLD measure of textual lexical diversity. 7, 19, 20, 35

NOUN noun. 42

NUM numeral. 41

PART particle. 42

POS part of speech. 28

PRON pronoun. 42

PROPN proper noun. 42

PTF primary translation frequency (metric used for evaluating lexical diversity). 23

REF the reference file that was used to test and evaluate the trained model. 19, 32

SCONJ subordinating conjunction. 41

SYM symbol. 42

TR the translation done by one of the trained models. 19, 32

TTR type-token ratio. 7, 19, 20, 35

VERB verb. 42

X other. 42

Acknowledgements

First and foremost, I wish to thank my supervisors, Nora Aranberri Monasterio and Claudia Borg, for all their help, advice, and guidance throughout the process of writing my thesis, and calming me when I had doubts and insecurities at any point of my Master's degree.

I'm also thankful to many UPV/EHU scholars who helped me with computational and theoretical issues when I encountered them: Gorka Labaka, Xabier Soto, and Kike Fernandez, to name a few. I believe it is also appropriate to thank Ruben Urizar for doing a lot of bureaucracy needed for Master Theses and making the process seem a little less intimidating.

A big thank you to Eva Vanmassenhove for providing me the resources needed to recreate a paper of hers and her collaborators.

My family, friends, and closest ones need to be thanked for their patience and relentless support, and for always listening to me talk for extended periods of time about my thesis (despite the fact that I suspect they hardly understood anything I was telling them).

Last, but not least, if any of my former professors and teachers are reading this, please know that this Master Thesis is here partially because of you as well. I believe each and every one of you helped me form as a student and a person. Thank you!

1 Introduction

Ever since the 1950s and the first machine translation systems, the branch of machine translation has been steadily and tirelessly improved. Starting from interlingua systems and continuing with rule-based and statistical, data driven approaches, today neural machine translation systems are able to create translations of high quality. The issue of whether we can use machines to translate was long ago answered, and now, being closer than ever to actively using machine translations that do not have to be post-edited in everyday life, it is time to start considering what effect this might have on the language that we are translating to.

Translation studies have long ago noticed several peculiarities in translations, such as the effect that source language sometimes had on the target language , but also features that seem to be persistent in all translations (Laviosa-Braithwaite, 1998). Interestingly enough, it seems that this effect, often referred to as *translationese*, can be identified by machines better than humans (Baroni and Bernardini, 2006); and using translated text in the source data can be beneficial for training machine translation systems due to their adaptation to the nature of translationese (Lembersky et al., 2012). Furthermore, human translations are known to have more explicit, normalized, and less rich or diverse language in terms of lexicon, syntax, etc. than the original text (Laviosa-Braithwaite, 1998). This effect is also present in machine translated text, as Vanmassenhove et al. (2019, 2021) show.

In this work, we analyze the effect of neural machine translation on two translation pairs across three corpora in total: two corpora for English↔Spanish, and one for English↔Croatian.

In Section 2 we present some related work on the topic, distinguishing between theoretical and practical work. In Section 3, we identify the goals of the work. Section 4 describes the methodology of creating and preparing corpora used for training neural machine translation systems and the procedure of training, translating, and evaluating the translation in terms of BLEU (Post, 2018) score. It then continues to describe how we go about assessing differences in lexical, morphological, and syntactical diversity of reference and translation texts. Section 5 displays the results of all the work, starting from translation evaluation, and continuing on to all the metrics and tools used to analyze the effect of machine translation.

2 Related Work

In this section, we present some related work on the effect of (machine) translation to the translated language. This section is divided into theoretical and practical work in order to make a distinction between the two.

2.1 Theoretical

The effects of translation on texts and language in general have been discussed for quite some time now in translation studies.

The term *translationese*, a set of common features that translations exhibit caused by translating from the source language, was first mentioned by Gellerstam (1986) (as cited in Santos, 1995). This effect or set of features is also known as *interference* (Toury, 2012 as cited in Kranich, 2014) and *shining-through* (Teich, 2003 as cited in Kranich, 2014).[1] While some of the first discussions of translationese were focused on inspecting the specific effect of the source language to the target language in translations (see , for examplesantos1995grammatical), it has also been established that translationese exhibits some features regardless of source and target language (Baker et al., 1993, Blum-Kulka et al., 1996, Laviosa-Braithwaite, 1998; as cited in Kranich, 2014).

Santos (1995) writes about grammatical translationese and tests several specific features (tense and aspect) of translations of English to Portugese and vice versa. She identifies four cases of grammatical translationese and then exemplifies and quantifies their appearance in translations. She proves the existence of several translationese features in the translations and notices that a relatively fine analysis was necessary in order to do so. She also mentions that when comparing the translations to the source, a smaller corpus suffices for detecting translationese, but that a larger corpus is necessary if only translated text is available for analysis. The relation of language closeness and translationese is also mentioned; stating that translationese tends to appear more easily when translating languages closer to each other.

Laviosa-Braithwaite (1998) lists out some features that appear in (human) translations

[1] However, it seems that these are not real synonyms: while all terms describe effects of translation on the target language, based on Koppel and Ordan (2011) *shining through* and *interference* seem to be refer to the effects in specific translation pairs, while *translationese* refers to general effects of translation independent of language.

irrelevant of the source and target texts. Some of the features discussed are simplification, explicitation, and distinctive distribution of lexical items. Simplification can be lexical, syntactic, or stylistical, and is the process of making the translation simpler in any of these three domains, with the actual way of simplifying depends on the domain itself. Blum and Levenston (1978) (as cited in Laviosa-Braithwaite, 1998) use evidence from several translation studies and exemplify the cases of simplification in the lexical domain: usage of superordinate terms if no hyponyms are available in the target language; usage of synonyms that are more commonly known; concepts approximation etc. In the syntactical and stylistical domain, Vanderauwera (1985) (as cited in Laviosa-Braithwaite, 1998) finds evidence for changes of non-finite clauses into finite clauses for syntax; and in the stylistical domain occurrences such as breaking longer sequences into shorter ones, omitting repetition, etc. Explicitation is the process of translators sacrificing the implicitness found in the source text in order to improve the clarity of the translation. This is mostly reflected as insertion or addition of words to the translation in order to provide clearer explanations. Normalization is the process of adapting the translation to the target language conventions, be it changes in punctuation style or "translating" foreign names. Vanderauwera (1985) (as cited in Laviosa-Braithwaite, 1998) finds that this process also adds to the clarity of the text because not only are cultural differences breached, but also incomplete and "clumsy" sentences are completed and rephrased, and chapters are ordered in a more logical manner, to name a few.

Tirkkonen-Condit (2002) conducts a study on whether or not humans are able to tell translations (done by humans) apart from original text. The conclusion of the study is that humans cannot distinguish between translations and original text. However, the factor that led subjects to their conclusions were unique or target language-specific items appearing in the text. Based on this finding, the author states that the role of unique items in translations needs to be further researched.

Kranich (2014) takes a detailed look into language change caused by (human) translation. Based on some previous works done in lexical contact through translation (LCTT), he identifies nine hypotheses about the nature of LCTT, discusses every one and their plausibility. Some of the hypotheses that have been identified as true by the author are: 1) lexical borrowing is more prominent than structural borrowing; 2) all linguistic domains are affected; 3) the impact of source language to the target language will be strongest at situations where the target language community has no (commonly accepted) written standard. Several statements are made in this paper that are interesting in the scope of our work: 1) human translators have awareness of norms and standards given a translation pair due to their required proficiency in both, and machine translation systems do not; 2) on the other hand, human translators (usually) have no insight into the frequencies of words, patterns, and structures, while machine translation systems do. The latter might be what plays into humans' inability and computer system's ability to tell translations and original text apart (see Tirkkonen-Condit, 2002).

2.2 Practical

Other than theoretical work done on translationese, its features, and its potential effects on language, many authors have attempted to identify translationese using machine learning approaches (see Baroni and Bernardini, 2006), or to compare qualities of source texts and translations in terms of, for example, lexical richness (see Vanmassenhove et al., 2021). Some authors have also done practical work and performed tests on translationese without using machine learning or computational resources (see Santos, 1995, for example).

Baroni and Bernardini (2006) attempt to train support vector machines for the task of identifying translations (of high quality, done by humans). Their results score up to 86.7% of accuracy and show that an ensemble of support vector machines outperforms the average result of a group of humans in this task, even when professional translators are part of the group. Distribution of function words, morphosyntactical categories and some certain parts of speech seem to be the clues that help support vector machines identify translations from text originally written in the language in question.

Lembersky et al. (2012) experiment with using original, translated, and a mixture of original and translated text to train language models for (phrase-based) machine translation. Their results show that language models trained on (human) translations, either from the source language or from other languages, have better performance than language models trained on texts originally written in the language examined. With regard to translation types, language models trained with translations from source to target language outperform language models trained with translations from other language to target language.

Zhang and Toral (2019) look into the effect of using translated texts in test sets on the performance of machine translation systems. The authors' results show that using translations in test sets leads to higher direct assessment scores and even changes the rank of the systems (in a competitive scenario).

Vanmassenhove et al. (2019) look into the loss and decay of lexical richness in machine translated data. They argue that machine translation systems have a tendency to over-accentuate frequent patterns in reference text and ignore less frequent ones; and also mention the inability of neural models to produce diverse output. Specifically, they state that neural models are more prone to generalization than statistical models. They translate and back-translate two language pairs (English↔Spanish and English↔French) and test their hypotheses, as well as lexical diversity using several metrics. They conclude that lexical diversity and richness do suffer after machine translation (with neural models retaining the most of it), and that overall machine translation does add to the frequency of most frequent items and take away the from the frequency of least frequent items.

Vanmassenhove et al. (2021) also test several machine translation architectures for the effect of loss of lexical and grammatical richness. The results they get are in line with Vanmassenhove et al. (2019) and confirm the existence of what they call "algorithmic bias". For all metrics and language pairs they find a loss of lexical and grammatical richness between the reference and the translation texts. They also mention that neural architectures (transformers) to cause the least loss.

In this work, we will continue in the direction of Vanmassenhove et al. (2019, 2021) and test neural machine translation systems for the effect they cause in the translations regarding lexical, grammatical, and syntactical richness. Vanmassenhove et al. (2021) chose Spanish and French as languages to use for their experiments because they are more morphologically complex than English. In order to further test the effect this might have, we use Croatian instead of French, as it is even more morphologically complex than Spanish.

3 Goals

The goal of this work is to analyze the effect machine translation has on two translation pairs (bidirectional translation): English ↔ Spanish; and English ↔ Croatian. The scenario we want to test this effect in is a "real-life" scenario of training a neural machine translation model (i.e. the current state-of-the-art architecture) from scratch and looking into the effect on the test set, i.e. a text for which the translation model is being trained. Furthermore, the source language for some of our data is unknown, as is sometimes the case in real life applications.

This work will hopefully further reassure previous findings in this domain and perhaps raise some further questions and open new directions for future work. Its relevancy stems from the fact that understanding the effect machine translation has on the translated language itself can help us (better) understand a) what machines learn when they learn to translate; b) what we need to keep in mind when training machine translation systems, and perhaps most importantly, c) what to keep in mind when putting machine translated texts into use, whether it is administrative, educational, or recreational use.

The first step will be to create and gather data needed for machine translation. After this, machine translation systems will be trained following Vanmassenhove et al. (2021) and OpenNMT's suggested training parameters[2]. Next, the translated texts will be analyzed to assess the effect machine translation has had on the texts. More specifically, the texts will be analyzed on three levels: lexical, morphological, and syntactical diversity or richness. For lexical diversity, we will report the following metrics and analyses: type-/token ratio, measure of textual lexical diversity, Yule's I, lexical frequency profile, and synonym frequency analyses. Morphological diversity will be analyzed using Shannon's entropy, Simpson's diversity index, and part of speech distribution. For syntactical diversity, we will look into the distribution of parses in the reference and translation texts.

[2] `https://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model`

4 Methodology

In this section the methodology of the work necessary to analyze the effect of machine translation on the quality of the (translated) text is explained. Vanmassenhove et al. (2021) analyzed the lexical and grammatical richness or diversity of machine translated texts in order to assess the effect machine translation has had on the text itself. Since their work performs a pretty wide and detailed analyses, it was decided to replicate their approach and the process of doing so is described below. After recreating part of their work using the same data, this is used as the “baseline” for some of the lexical and grammatical richness tests carried out on other datasets with an additional language pair. The preprocessing, training, translation, and postprocessing of the datasets, as well as the measuring of the lexical and (most of) the morphological diversity, was replicated and/or adapted from the work of Vanmassenhove et al. (2021).

4.1 Overview of Vanmassenhove et al. (2021)

Vanmassenhove et al. (2021) train statistical phrase-based and neural models for translation of English to French, French to English, English to Spanish, and Spanish to English. They translate texts using the trained models, evaluate the quality of the translation itself using BLEU and TER scores, and finally, test the translations for lexical and morphological richness. To test the lexical richness, some of the metrics they use are the following:

- Type-token ratio (TTR)
- Measure of textual lexical diversity (MTLD)
- Yule’s I (the inverse of Yule’s K)
- an adapted version of Lexical frequency profile
- Synonym frequency analysis

For testing grammatical richness, the following tests/metrics are applied:

- Shannon’s entropy
- Simpson’s diversity index

Eva Vanmassenhove was kind enough to provide the authors’ GitHub repository[3] for

[3] https://github.com/dimitarsh1/BiasMT

the purpose of recreating their work. A different repository[4] owned by the same account contains scripts used to preprocess data, train models, test them, and postprocess the data, as well as perform the lexical richness analyses. As mentioned above, the authors performed tests on the translations of a statistical phrase-based system and neural systems (RNN and Transformer architectures). The recreation described here focuses only on the transformer model, as it is the current state-of-art, and the primary focus of this work is not to compare different architectures and their results, but rather to look into the effect of machine translation.

4.2 Creating a Corpus

Vanmassenhove et al. (2021) chose Spanish and French to compare with English because both languages are morphologically more complex than English and they wanted to see how this complexity affects translation quality. In order to widen the work the authors presented, it was decided to add an additional language pair - English ↔ Croatian. One of the reasons Croatian was chosen is the fact that it is morphologically even more complex than Spanish or English. Croatian is a South Slavic language with fusional morphology and seven cases, and is a low-resource language.

According to one META-NET study (Rehm et al., 2014), Croatian was classified as having weak to no support in three out of the four categories of language technologies they looked into (the fourth category having fragmentary support).[5] A more recent report[6] from 2022, done by Marko Tadić within the European Language Equality (ELE) project, states that out of 12 categories, Croatian has fragmentary support in seven, and weak support in six.[7]

The Europarl corpus (Koehn, 2005), which Vanmassenhove et al. (2021) used as training data, was first created in 2005 with the latest version released in 2011. Given that Croatia has become a member of the European Union in 2013, and that the source of data for Europarl is the proceedings of the European Parliament, no version of Europarl contains Croatian. To the knowledge of this author, a ready-to-use corpus comparable

[4] https://github.com/dimitarsh1/NMTScripts

[5] To make an interesting comparison, Basque was classified as having more support than Croatian in three out of the four categories; while in the fourth category they have the same level of support.

[6] Available at: `https://european-language-equality.eu/wp-content/uploads/2022/03/ELE___Deliverable_D1_7__Language_Report_Croatian_.pdf`

[7] For the sake of comparison, according to the same report Basque has weak support in seven and fragmentary support in five categories.

in size and domain to Europarl which features Croatian as one of its languages does not exist. In order to add Croatian to this work, it was decided to create a new corpus using the European Commission's Directorate-General for Translation's data[8] (Steinberger et al., 2012). The data is released yearly as a `.zip` folder containing an abundance of `.tmx` files.

As stated above, Croatia became a member of the European Union in 2013, so files issued before 2013 were not translated to Croatian. Releases 2014 to 2016 were downloaded from the links available on the Directorate-General for Translation's Translation Memory website[9], while those from 2017 onwards were not available for download on this website (there appear to be some technical issues). The first part of the dataset was created using the files from releases 2014 to 2016.

Directorate-General for Translation Translation Memory's website[10] lists an existing tool for extracting bilingual pairs - `TMXtract.jar`, but it did not seem to be functional. Instead, a custom Python script was written to transform the `.tmx` files to `.xml`, and then extract sentences of the desired language pairs. The files were first transformed to `.xml` format because an XML parser[11] was used to process them. To ensure alignment between translation units of all languages is preserved, the script only extracted those translation units that were aligned in English, Spanish, French, and Croatian. The translation units were written to separate `.txt` files, and 1 179 025 translation units (per language) were extracted from these releases. When one takes into account the cleaning of the dataset that will come and the split to train, validation, and test sets, it is clear that this dataset is much smaller than Europarl.

Later, an additional source[12] of the Directorate-General for Translation Translation Memory data was found, and it had releases 2018 to 2020 available for download. More translation units were added to the corpus after some minor changes to the original Python script, thus creating a larger corpus which had 2 223 797 translation units (per language) aligned between English, Spanish, French, and Croatian. Release 2017 was not available

[8] Available at: `https://joint-research-centre.ec.europa.eu/language-technology-resources/dgt-translation-memory_en`, releases 2013-2016. European Commission retains ownership of the data.

[9] `https://joint-research-centre.ec.europa.eu/language-technology-resources/dgt-translation-memory_en#download`

[10] `https://joint-research-centre.ec.europa.eu/language-technology-resources/dgt-translation-memory\_en#download`

[11] `https://pypi.org/project/beautifulsoup4/`

[12] Available at: `https://data.europa.eu/data/datasets/dgt-translation-memory?locale=en`, releases 2018-2020. European Commission retains ownership of the data.

Decision of the EEA Joint Committee

of 13 December 2013

amending Annex I ([annex name]) to the EEA Agreement

THE EEA JOINT COMMITTEE,

Having regard to the Agreement on the European Economic Area ('the EEA Agreement'), and in particular Article 98 thereof,

Whereas:

Figure 1: Examples of frequent translation units at the beginning of documents.

for download at any of the sources that were found and is therefore not featured in the corpus.

The nature of this data is written legislation, regulations, and other such documents of the EU (more precisely, the data source is documents of Acquis Communautaire, which is "the entire body of European legislation, comprising all the treaties, regulations and directives adopted by the European Union"[13]), and therefore comes with a certain amount of noise and repetition. As explained in the Directorate-General for Translation Translation Memory's website[14], the exact source language of files in unknown, but many texts are originally written in English and then translated to other languages.

Figures 1, 2, 3 and 4 provide examples of translation units which could be considered noise or too repetitive. For example, almost every document in the corpus starts with a preamble which consists of one or more of the translation units listed in Figure 1, and ends with one of the translation units listed in Figure 2. Some translation units provide no useful data for training a machine translation model, like those listed in Figures 3 and 4.

A Python script was written in order to clean out some of the noise found in the data

[13] Source: `https://joint-research-centre.ec.europa.eu/language-technology-resources/dgt-translation-memory_en#dgt-memory` ; Introduction - view details.

[14] `https://joint-research-centre.ec.europa.eu/language-technology-resources/dgt-translation-memory_en#dgt-memory`

Done at Brussels, 13 December 2013.

For the EEA Joint Committee

The President

College member

Figure 2: Examples of frequent translation units at the end of documents.

Article [number]

Chapter [number]

ANNEX [number]

Figure 3: Examples of frequent, very short and repetitive translation units.

OJ L 202, 27.7.2013, p. 33.

32013 R 0691: Commission Regulation (EU) No 691/2013 of 19 July 2013 (OJ L 197, 20.7.2013, p. 1).';

Figure 4: Examples of translation units or parts of translation units referencing documents, articles, etc.

Article [number] | CHAPTER [number] | ANNEX [number]

For the EEA Joint Committee | Decision of the EEA Joint Committee | THE EEA JOINT COMMITTEE

Whereas

The President | College Member

Done at [location] | of [date] | HAS ADOPTED THIS DECISION:

Figure 5: Examples of translation units which were removed from the corpus.

	Total number of extracted translation units per language
Releases 2014-2017	1 179 025
+ **Releases 2018-2020**	2 223 797
After cleaning	2 113 792

Table 1: Total number of the translation units in the created dataset.

using regular expressions. Specifically, translation units that can be seen in Table 5 were deleted.

Other translation units which contained references to documents (like those listed in Figure 4) were also removed (but not all - these were often mentioned in the middle of sentences and it was very difficult to completely clean them out without jeopardizing the sentence itself). Some translation units were (deliberately or not) copied to all languages and those were also deleted from the corpus. More specifically, those translation units which were identical in all languages were dropped (these were, for example, some non-linguistic notations such as '$< \alpha, \beta < +5°$', or names of establishments, such as 'AIR MADAGASCAR'). A filtering step was added to try and preserve names, but delete other noise. Finally, all of these data cleaning steps resulted in the removal of 110 005 noisy translation units, and the cleaned version of the corpus has 2 113 792 translation units (per language). From this point on, this corpus will be referred to as 'DGT'.

4.3 Data Preprocessing

Vanmassenhove et al. (2021) use the OpenNMT library and various other tools to preprocess their data. The preprocessing steps listed in the Github repository Eva Vanmassenhove provided[15] are to:

- 1. split the data
- 2. apply the tokenizer
- 3. apply the truecaser
- 4. train and apply byte-pair encoding (BPE), create dictionary.

In order to ensure maximal clarity and reproducibility, we will now briefly discuss all the preprocessing steps mentioned above. The recreation is focused only on the English-Spanish language pair (from now on: 'EN-ES', or 'ES-EN' for the opposite translation direction), and the added language pair of English-Croatian (from now on: 'EN-HR', or 'HR-EN' for the opposite translation direction). In other words, the French portion of Europarl was not used in any part of recreating Vanmassenhove et al. (2021). The same steps and scripts listed in the aforementioned Github repository were employed as the preprocessing of all the data. It should be mentioned that some of the scripts were amended in order to fix some existing dependency issues and rename the output files, but none of the scripts were changed in a way that should affect their output and its quality.

4.3.1 Preprocessing Europarl

Europarl's data had empty lines which caused errors when calculating the BLEU score. Therefore, step 'zero' was to clean out empty lines from both the source and the target texts. Splitting the data into train, validation, and test set was done using the author's `1_split_ttv.sh` script, keeping Vanmassenhove et al.'s (2021) splitting ratios in mind (75.98% for train, 23.72% for validation, and 0.3% of data for test set, as seen in Table 2).

The next step was to tokenize using the script `2_tokenize.sh`. The tokenizer used in this script is Moses tokenizer (Koehn et al., 2007) (`.perl`). This script also cleaned out sentences that were too long, and this threshold was left as-is in the original script.

After tokenizing came truecasing the data using (`3_truecase_data.sh`). The truecaser script was changed to take the cleaned version of the training data (the final output of the

[15] https://github.com/dimitarsh1/NMTScripts

previous script), instead of the only-tokenized version. Finally, training and applying BPE (`3_dictionary_bpe.sh`) was carried out with the default setting of 50 000 operations.

From now on, the Europarl corpus will be referred to as 'EP'.

4.3.2 Preprocessing DGT

The preprocessing of the corpus created for the purpose of this Master Thesis using DGT-Translation Memory releases 2013-2016, 2018-2020 (DGT) corpora had some additional steps. Due to the fact that this data source is legislative documents of European Union, and despite the fact that de-noising was performed, there was still significant overlap between the training, validation, and test sets. After applying the `1_split_ttv.sh` script on the DGT corpora, the overlap was as follows:

- (EN) percentage of training sentences in the test set: 15.67%
- (ES) percentage of training sentences in the test set: 14.03%
- (EN) percentage of training sentences in the validation set: 14.43%
- (ES) percentage of training sentences in the validation set: 12.62%.

In comparison, the overlap between datasets of Europarl sets were approximately between 0.5 and 1%. It is clear that this overlap is too big for fair and representative training of a machine translation model. This is why an additional preprocessing step of removing overlapping sentences from all combinations of the datasets was undertaken twice. The first removal of overlapping translation units happened during manually splitting the corpus into train, validation, and test splits, and then again after all other preprocessing steps had been carried out. Table 2 shows the number of sentences per data split per corpus, and the percentage of sentences in every split.

After manually spliting the corpus into training, validation, and test splits, the same tokenizing (`2_tokenize.sh`) and BPE (`3_truecase_data.sh`) scripts were applied. For Croatian the tokenizer reported an issue: `WARNING: No known abbreviations for language 'hr', attempting fall-back to English version....` This warning was ignored and the fall-back to the English version was used, and a manual inspection of the tokenization seemed fine.

It is worth mentioning that the DGT datasets for English-Spanish and English-Croatian were processed separately; and that the English corpus used to train English-Spanish and English-Croatian models is not the same collection of translation units. This is because it was nearly impossible to ensure that the data for all three languages is processed and split in the same way due to the removal of overlapping sentences. The removal of overlapping translation units is also cause for the final number of translation units to be much lower that the raw number of translation units, and it made it very difficult to manage the ratio of translation units in the data splits versus the total number of translation units. Table 2 shows, however, that the numbers of translation units and the ratios of splits between all corpora are somewhat comparable.

Corpus	Translation pair (↔)	**Train**	**Validation**	**Test**
Vanmassenhove et al. (2021)	EN-ES	1 472 203 (75.9%)	5 734 (0.3%)	459 633 (23.7%)
EP		1 486 952 (75.1%)	5 882 (0.3%)	462 712 (23.6%)
DGT		1 289 235 (75.29%)	8 486 (0.49%)	414 591 (24.21%)
DGT	EN-HR	1 289 18 (75.21%)	8 493 (0.49%)	416 403 (24.29%)

Table 2: Number of translation units and the ratio of given set as opposed to the final number of translation units per dataset for EP and DGT data.

4.4 Training Models

Vanmassenhove et al. (2021) report that they used OpenNMT[16] (Klein et al., 2017) for training the transformer machine translation models. The OpenNMT version they use, however, is not specified. The most recent available version (2.2.0., OpenNMT-py) at the time of training the models was used. Vanmassenhove et al. (2021) do specify the parameters they used for training, and they are the same settings described in OpenNMT FAQ[17]. Additionally, they mention that the "learning decay [is] enabled" (Vanmassenhove et al., 2021, p. 2205). All and only the parameters from the FAQ were used, with the learning decay set to its default value. These parameters include, but are not limited to those laid out in Figure 6[18].

Before running the training script, Vanmassenhove et al. (2021) use a `preprocess.py` script, which seems to be deprecated in the latest OpenNMT version. Because of this,

[16] `https://opennmt.net/`

[17] `https://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model`

[18] Data taken from `https://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model`

```
batch_type: ''tokens''
batch_size: 4096
accum_count: [4]
accum_steps: [0]

optim: ''adam''
learning_rate: 2
warmup_steps: 8000
decay_method: ''noam''
label_smoothing: 0.1
param_init: 0
param_init_glorot: true
normalization: ''tokens''

position_encoding: true
enc_layers: 6
dec_layers: 6
heads: 8
rnn_size: 512
word_vec_size: 512
transformer_ff: 2048
```

Figure 6: Some of the parameters used for training models.

the `onmt_build_vocab.py` script was ran in order to build the vocabularies necessary for training. Multiple approaches to building and using the vocabulary were tried out prior to settling on this one: the first idea was to use the already trained BPE models to apply on-the-fly tokenization which is introduced in the latest version of OpenNMT. However, the BPE model created by `3_dictionary_bpe.sh` consistently caused `RuntimeError: unsupported BPE version`. A different BPE architecture was applied and tried out, as well as creating a BPE model on-the-fly, but these approaches were either not performing well, or were deemed too different from the author's approach. This is why, in the end, the `onmt_build_vocab` was used with the preprocessed versions of the data, no transforms specified, and with the number of samples set to -1 (the whole corpus is used to create a vocabulary). The vocabularies created were used in both training directions, but were not shared vocabularies (i.e. there are two separate vocabularies for each language of the pair, but they are used for both directions). Table 3 shows the sizes of vocabularies after running `onmt_build_vocab` on the preprocessed files. It is interesting to note that even though the same vocabulary files were used for both translation directions, the size of the loaded dictionaries at the start of training differ by a couple of words.

Corpus	**Vocab size**
Vanmassenhove et al. (2021) EN	47 639
Vanmassenhove et al. (2021) ES	49 283
EP EN (EN-ES)	46 287
EP ES (EN-ES)	48 989
EP EN (ES-EN)	46 289
EP ES (ES-EN)	48 987
DGT EN (EN-ES)	48 131
DGT ES (EN-ES)	48 557
DGT EN (ES-EN)	48 133
DGT ES (ES-EN)	48 555
DGT EN (EN-HR)	48 089
DGT HR (EN-HR)	49 700
DGT EN (HR-EN)	48 091
DGT HR (HR-EN)	49 698

Table 3: Vocabulary sizes for training after building the vocabulary based on the preprocessed files.

After building the vocabularies, the models were trained. Following Vanmassenhove et al. (2021), all models were trained with one GPU for a maximum of 150 000 steps, and early stopping was enabled with the criteria of no improvement in perplexity on the

Model	GPUs	Total steps	Best model at step:
EP EN-ES	1	90 000	40 000
EP ES-EN	1	90 000	40 000
EP EN-ES	1	90 000	40 000
EP ES-EN	1	80 000	30 000
DGT EN-HR	1	90 000	40 000
DGT EN-HR	1	80 000	30 000

Table 4: Model, number of GPUs it was trained on, total steps trained for and the step at which the best model was identified.

validation set for 5 checkpoints. Due to this early stopping criteria, the models were in reality trained for 90 000 steps at most. Table 4 shows the exact information of steps for each model trained.

4.5 Translation

After the model has trained, the best model identified by the training log of OpenNMT was used to translate the test set. Models were evaluated every 10 000 steps, as is the default setting of OpenNMT. Even though validating (and saving) a model more frequently might be better for truly identifying the best model, it was decided to use the default settings in order to mimic a "real-life" scenario. Perplexity on the validation dataset was used as the early stopping criteria following Vanmassenhove et al. (2021), with patience set to 5 (stop if there is no improvement in perplexity after evaluating on the validation dataset 5 times).

One big difference is the fact that Vanmassenhove et al. (2021) translate training data which was completely observed during training because they argue that this is the best scenario to evaluate and asses the effect of machine translation. In this work we translate the test data, which is unseen during training, and perform all tests and analyses on test sets (reference) and translation of test sets (translation). The motivation for this is the fact that we are interested in seeing and looking into the effect of machine translation in a realistic scenario.

4.5.1 Postprocessing Translations

The translations were postprocessed using the same postprocessing steps and commands as Vanmassenhove et al. (2021), but not the postprocessing script itself that is available in their Github repository.[19] First, BPE segmentation was removed using regular expressions. Next, detokenizer and detruecaser were applied. Finally, `sacrebleu` (Post, 2018) was used in the command line to obtain BLEU scores, which are reported in Table 5 in Section 5.1. The original train, validation, and test splits of the DGT corpora are also postprocessed in the same manner in order to calculate metrics and use them for comparison with translations. This is necessary because the final removal of overlapping translation units was done after applying all preprocessing steps, and therefore the splits used during training were segmented into subwords, tokenized and truecased.

4.6 Measuring Lexical Diversity

After training neural machine translation models and successfully translating the test set, and evaluating the translations in terms of BLEU scores, we can start the analyses of the translation quality in several linguistic domains. The first domain we will analyze is lexicon, i.e the lexical diversity or richness.

As in Vanmassenhove et al. (2021), lexical diversity was scored using some common lexical diversity tools such as type-token ratio (TTR) and measure of textual lexical diversity (MTLD). The authors also used Yule's I and a number of metrics that they created or adapted. All metrics will be described below. The authors do have a script which calculates some of the metrics they report in the paper (`score_lexical_diversity.py`[20]), however it was unclear whether the input should be tokenized or untokenized; and it also gave very unusual results when passing the whole translated text as input (no matter if it is passed as a Python list, Python string, or a `.txt` file). Instead of using this script, the metrics were calculated using a custom written class and its methods, which attempted to follow the author's implementation with some improvements. For every metric, the exact way of its calculation will be described.

Lexical diversity was calculated on the translation, as well as the reference text. For example, text `test.en` was translated using the `model_en-es`, giving us text `translation_en-es`. Lexical diversity metrics were calculated for `translation_en-es`, which is denoted with TR and also for `test.es`, which is denoted with REF.

[19] `https://github.com/dimitarsh1/NMTScripts/blob/main/7_postprocess.sh`

[20] Available at: `https://github.com/dimitarsh1/NMTScripts/blob/main/score_lexical_diversity.py`

4.6.1 TTR (Type-token Ratio)

Type-token ratio TTR is a measure for scoring vocabulary richness that looks at the ratio of unique words (types) versus the total words (tokens) in a text (Oakes and Ji, 2012). It provides an idea of a texts' repetitiveness or diversity. The higher the TTR, the richer or more diverse the vocabulary of the text. The main flaw of this metric is the fact that it is sensitive to text length, meaning that the longer the text, the smaller the TTR score tends to be (Oakes and Ji, 2012). This is not that surprising given that after reaching a certain length, a text must become more and more repetitive in terms of lexicon (McCarthy and Jarvis, 2010).

TTR is calculated by dividing the number of types with the number of tokens. While Vanmassenhove et al. (2021) calculate TTR manually, in this work it was decided to calculate TTR using the `lexical-diversity`[21] Python library. The translation units were first tokenized using `lexical-diversity`'s tokenizer. This tokenizer tokenizes words on whitespace and lowercases them; while also removing sentence boundaries. This enables us to more easily focus on the lexicon itself by ignoring punctuation and casing. The list of tokens is then passed to the `TTR()` method of `lexical-diversity`.

4.6.2 MTLD (Measure of Textual Lexical Diversity)

Measure of textual lexical diversity MTLD is a metric that uses TTR in its calculations - it reports the mean length of a portion of text in which a given TTR value is maintained (McCarthy, 2005 as cited in Vanmassenhove et al., 2021). It takes into account the so-called point of stabilization, i.e. the point where the sharp and drastic changes in TTR values are stabilized (McCarthy and Jarvis, 2010). According to McCarthy and Jarvis (2010), it is considered a more sophisticated method of measuring lexical diversity.

MTLD is calculated by dividing the number of words in a text by a factor count; where the factor count increases by one every time the TTR of a sequence falls under a certain threshold (the default being .720) (McCarthy and Jarvis, 2010). Vanmassenhove et al. (2021) calculate MTLD using `lexical-diversity`'s `MTLD()` method, and this work uses the same approach. The input to the `MTLD()` method of `lexical-diversity` is a list of tokens obtained with `lexical-diversity`'s tokenizer. This ensures that the input to the calculations of TTR and MTLD is the same.

[21] `https://pypi.org/project/lexical-diversity/`

4.6.3 Yule's I

Yule's I is the inverse of Yule's K (Yule, 1944 as cited in Vanmassenhove et al., 2021). Yule's K "measures constancy of text and the repetitiveness of vocabulary" (Vanmassenhove et al., 2021, p. 2208). It was designed to be more resilient to the aspect of text length, but is a measure better suited for text uniformity rather than diversity or richness because it returns lower values for texts which have a richer vocabulary (Oakes and Ji, 2012). Note that Oakes and Ji (2012) report that Yule's K also tends to decrease with text length, but not as smoothly as TTR does.

Yule's I is calculated as follows:

$$(M1 \times M1)/(M2 - M1)^{22}$$

where M1 is the number of tokens, and M2 is the sum of taking a number of words with a certain count and multiplying it by a square of that count (so, all words that appear three times will be multiplied by 3^2). Usually, the MTLD result is multiplied by 10 000 (Oakes and Ji, 2012). The calculation of Yule's I was taken from Vanmassenhove et al.'s (2021) `score_lexical_richness.py` script. The number of tokens and their count were taken from the lists of tokens obtained by `lexical-diversity`'s tokenizer. Once again, this ensures that the input to Yule's I is the same as the input for TTR and MTLD.

4.6.4 Statistical Significance

Statistical significance of the results of TTR, MTLD, and Yule's I is established for every system and translation direction and it was based on statistical significance tests implemented in Vanmassenhove et al.'s (2021) code[23].

The authors take a random sample of 1000 sentences and calculate how many times one text outscored the other (how many times the reference text scored better than the translation, and vice versa) and calculate the p-value of TTR, MTLD, and Yule's I scores on the samples using `scipy.stats.ttest_ind`.

In our implementation of lexical diversity calculations, it was difficult to take a sample of sentences since the inputs to TTR, MTLD, and Yule's I are lists of tokens obtained with

[22] Formula taken from `https://github.com/dimitarsh1/NMTScripts/blob/f08d669ccf51bfe466ba70d176fdbdf485eef632/score_lexical_diversity.py`, line 96.

[23] Available at: `https://github.com/dimitarsh1/NMTScripts/blob/f08d669ccf51bfe466ba70d176fdbdf485eef632/score_lexical_diversity.py`, lines 15-36; 39-54; 130-147

lexical-diversity's tokenizer. Instead, we took 1000 random token sequences of 3000 tokens and performed statistical significance calculations on these.

4.6.5 LFP (Lexical Frequency Profile)

As mentioned above, Vanmassenhove et al. (2021) create and/or adapt several metrics for lexical diversity/richness. The first metric they create or adapt is the Lexical frequency profile (LFP). LFP (Laufer, 1994; Laufer and Nation, 1995, as cited in Vanmassenhove et al., 2021) is usually used to score a language learner's grasp of a language and relies on the assumption that the better the students' vocabulary, the more less-frequent words he or she will use. This in turn makes the text more sophisticated (Kyle, 2019 as cited in as cited in Vanmassenhove et al., 2021). LFP divides the words of a text into four bands: percentage of words that are in the 1000 most common words; percentage of words that are in the 1000-2000 most common words; percentage of academic words not occurring in the first two bands; and all other words. The word frequencies are taken from outside sources such as word frequency lists. Following this logic, the less words in first two bands, the more nuanced the text should be.

Vanmassenhove et al. (2021) adapted this metric to be used for quantifying lexical richness of a translation by calculating the word frequencies using the training data and removing band 3 (academic words not featured in bands 1 and 2), calculating only bands 1, 2, and 4 (0-1000, 1000-2000, and 2000-rest). The comparison of frequency bands of the source and target data provides interesting information about the decrease in text sophistication after machine translation. The authors do not lowercase, tokenize, remove numerals, etc; but do use already tokenized text as the input[24]. In this work, LFP was implemented on text that was tokenized with lexical-diversity's tokenizer – which lowercases tokens and removes punctuation. The reason for this is the fact that our primary interest is true lexical diversity, and given that this tokenizer removes punctuation and lowercases all words, we can focus on the lexicon only.

Vanmassenhove et al. (2021) state that they use the original training data to calculate word frequencies, but also that they use the same data to train, evaluate, and test their systems, as well as the fact that they translate the training data. This means that the LFP

[24] This is not mentioned in Vanmassenhove et al. (2021), but in the implementation code at https://github.com/dimitarsh1/BiasMT/blob/4b4012c3cb117b7229120afe976ec10fc03228fb/scripts/diversity/biasmt_metrics.py, lines 339-341.

on text A is calculated using frequencies of text A, and the LFP of text B is calculated using word frequencies of text B. Strictly speaking, this means that the results of text A and text B are not fully comparable; but one could discuss their relative differences if the fact that different data was used as the starting point is kept in mind. In this work, we determine word frequencies using the original training data and calculate LFP on both the reference and translation texts using the same word frequencies as a starting point. This gives us a nice compromise between the original LFP implementation, where independent word lists are used to calculate word frequencies, and the implementation of Vanmassenhove et al. (2021) who use the same data they are measuring LFP on as the word frequency source.

4.6.6 Synonym Frequency (PTF and CDU)

Vanmassenhove et al. (2021) develop two metrics for analyzing synonym frequency analysis: primary translation frequency (PTF) and cosine distance of uniform synonym distribution and the actual synonym distribution (CDU). These metrics are based on the authors' hypothesis that machine translation systems and their algorithmic bias will cause a certain word to be overused in the translation at the expense of other, also valid translations. A difference between synonym frequency of the source and target data could be (further) evidence of machine translation systems lowering lexical richness and/or diversity of a text; and possibly even the fact that the semantic richness of the translation is affected by machine translation. To be more specific: regarding PTF, Vanmassenhove et al. (2021) argue that a large prevalence of the most frequent option is a sign of algorithmic bias. Regarding CDU, they argue that even though a uniform distribution is not realistic in actual translation, the distance value between the uniform and actual distributions still provides information on the possible effect of machine translation on synonym distribution.

Vanmassenhove et al. (2021) describe the calculation of PTF and CDU as follows. The source and target texts are lemmatized and only nouns, verbs, and adjectives are used in this analysis. Next, a list of source words and target words which are their valid translation is obtained (those translation words make the 'synonyms'). All the synonyms' occurrences in the translated text are then counted. The counts are used to create a vector. Since there is no available code of these two metrics implementations, they were manually implemented following the description from the paper. It must be kept in mind that this means there exists risk of faulty or different implementation from the author's implementation.

PTF was calculated by taking the count of the most frequent synonym and dividing it synonym count: `{apropos: [('pertinente', 6229), ('oportuno', 829), ('a propósito', 0)]}`

synonym distribution vector: `[6229, 829, 0]`

uniform vector: `[6629+829+0` ÷ `3]` × `3 = [2352.6, 2352.6, 2352.6]`

PTF calculation: `6629` ÷ `(829+0) = 0.8825446302068575`

CDU calculation:
`scipy.spatial.cosine_distance([2352.6, 2352.6, 2352.6], [6229, 829, 0]) = 0.3515295264672129`

Figure 7: Calculating PTF and CDU for one example.

by the sum of counts of all other synonyms.

CDU was calculated by taking the sum of all synonym counts and dividing the sum with the number of synonyms, resulting in a uniform distribution vector. `scipy`'[25] `spatial.cosine_distance` was used to calculate the cosine distance between the uniform vector and the actual distribution vector.

For every source word, PTF and CDU were calculated together, and if one of the calculations caused a `ZeroDivisionError` (error encountered when attempting to divide something by zero), both PTF and CDU calculation for this word were skipped. The average PTF and CDU results for every source word and synonyms pair was taken as the text's PTF and CU result. Figure 4.6.6 shows calculation of PTF and CDU for one example, which was randomly chosen from the DGT EN-ES data.

It must be noted that neither of these metrics take context or domain into account, which might affect the meaningfulness of the results. For example, Vanmassenhove et al. (2021) give an example of the English word *look* and its possible Spanish translations: *mirar*, *esperar*, *buscar*, *parecer*, *dar*, *vistazo*, *aspecto*, *ojeada*, *mirada*. According to an online dictionary, WordReference[26], *ojeada* can be translated into English as "glance, quick look, throw an eye over" or "keep an eye on".[27] It is hard to imagine a legislative text which

[25] `https://scipy.org/`

[26] Available at: `https://www.wordreference.com`

[27] This example can be seen here: `https://www.wordreference.com/es/en/translation.asp?spen=`

would be using such an expression, and even harder to imagine a legislative text using this expression often. Furthermore, the same translation can be used to calculate PTF and CDU of multiple source words.

Following Vanmassenhove et al. (2021), 1) synonym frequency metrics are calculated only in one direction, with English as the source language; 2) SpaCy[28] was used to lemmatize Spanish texts; and 3) the same bilingual dictionary, "en-es-en Dic"[29] was used to obtain the synonyms. For Croatian, English-Croatian dictionary files[30] were used, and the texts were lemmatized using a fork of Stanford Stanza pipeline for (some) South Slavic languages named `CLASSLA`[31] (Ljubešić and Dobrovoljc, 2019).

4.7 Measuring Morphological Diversity

Vanmassenhove et al. (2021) use Shannon's entropy (H) and Simpson's diversity index (D) to measure what they call the grammatical diversity of a text. They use the term "grammatical diversity" to indicate those elements of language which appear in the morphological and syntactical domains. For Shannon's entropy, Simpson's diversity index and its inverse, the code from Github repository[32] of Vanmassenhove et al. (2021) was used.

4.7.1 Shannon's Entropy

We can use the following definition to define Shannon's entropy: "Shannon entropy (H) measures the level of uncertainty associated with a random variable" (Vanmassenhove et al. 2021, p. 2209). The authors use it to measure the entropy of wordforms of a lemma, i.e. its inflectional paradigm.

Shannon's entropy is a concept taken from information theory and was constructed by Shannon (1948) (as cited in Vanmassenhove et al., 2021). It can also be explained as the level of information a certain entity provides: if something unexpected appears, the informativeness or entropy is higher; and lowers in the case of something usual and expected appearing[33]. If we apply this to lemmas and their inflectional paradigms, the

ojeada.

[28] https://spacy.io/. We used es_core_news_lg pipeline.

[29] https://github.com/mananoreboton/en-es-en-Dic

[30] https://github.com/gigaly/rjecnik-hrvatskih-jezika

[31] https://pypi.org/project/classla/

[32] https://github.com/dimitarsh1/BiasMT/blob/4b4012c3cb117b7229120afe976ec10fc03228fb/scripts/diversity/biasmt_metrics.py, lines 114-203.

[33] https://en.wikipedia.org/wiki/Entropy_(information_theory)

Wordforms for lemma *asociado*: `’asociado’: 1672, ’asociados’: 3, ’asociadas’: 1, ’asociada’: 6`

total number of wordforms: `1672+3+1+6 = 1682`

calculation of $p(wf|l)\log p(wf|l)$: $(1672 \div 1682) \times \log(1672 \div 1682)$

Now we sum the $p(wf|l)$ of all wordforms and the negative result is H of this lemma.

Figure 8: En example of calculation of H, step by step, for one lemma.

more complex a paradigm (i.e. the greater the number of different wordforms that belong to it), the higher the entropy of the lemma will be. This is why we can say that higher value of H indicates higher morphological diversity – it simply means that the inflectional paradigms of our lemmas tend to be more complex, diverse, and in a way, informative. A lemma with one wordform has H of 0.0, i.e. is not considered informative at all.

For the calculation of H, we use the code from Vanmassenhove et al.’s 2021 GitHub repository[34]. H is calculated using the following formula[35]:

$$H(l) = -\sum_{wf \in l} p(wf|l) \log p(wf|l) \tag{1}$$

where $H(l)$ is the Shannon entropy for lemma l; and $p(wf|l)$ is the count of one wordform divided by the count of all wordforms.

The input must be a nested Python dictionary of lemmas and all their wordforms. H is calculated per lemma. Vanmassenhove et al. (2021) report that while calculating H for lemmas with single wordforms is useful for analyzing the paradigm in question on its own, such lemmas affect the overall score of H and do not give useful information for quantifying the morphological diversity. This is why they do not take lemmas with only one wordform into account (but do report the number of such lemmas), and we did the same. The final result is the mean value of H across all lemmas.

[34] `https://github.com/dimitarsh1/BiasMT/blob/4b4012c3cb117b7229120afe976ec10fc03228fb/scripts/diversity/biasmt_metrics.py`, lines 144-164; 322-336.

[35] Formula taken from Vanmassenhove et al. 2021, p. 2209.

Dictionaries of lemmas and their wordforms were created using the same lemmatizers as for lexical diversity experiments, i.e. SpaCy's[36] `es_core_news_lg` pipeline for Spanish, and a fork of Stanford Stanza pipeline for (some) South Slavic languages named `CLASSLA`[37] (Ljubešić and Dobrovoljc, 2019). For English, Spacy's `en_core_web_lg` pipeline was used.

4.7.2 Simpson's Diversity Index

Similarly to Shannon's entropy, Simpson's diversity index (D) (Simpson, 1949 as cited in Vanmassenhove et al., 2021) also measures categorical data diversity (Vanmassenhove et al., 2021). In general, this measure seems to be used for ecological diversity of an area – if there are multiple species who live in the area and have similar (large) numbers of organisms per species, the area is considered more diverse than if fewer organism and fewer species were to inhabit it[38].

Vanmassenhove et al. (2021) apply Simpson's diversity index to lemmas and their wordforms and adapt the original formula to lemmas and wordforms calculations. If we draw a parallel between the original and the author's impletentation, it could be said that D is used to calculate the diversity (potential) of a lemma: if the lemma has more wordforms, it will be considered more diverse. Note that this metric was first implemented by the authors and there is not much frame of reference or detailed explanations as to why they change the formula the way they do. We conducted some tests of the calculation of D and it seems like it favors lemmas with wordforms that have equal counts as more diverse than lemmas with the same number of wordforms, but different counts. Since in their formula the authors calculate D by dividing one with the wordform calculations, greater values of D indicate lower morphological diversity. A lemma with one wordform has D of 1.0.

Vanmassenhove et al. (2021) calculate D using the following formula:

$$D(l) = \frac{1}{\sum\limits_{wf \in l} p(wf|l)^2} \tag{2}$$

Just like for Shannon's entropy, the input is a Python dictionary with lemmas and all their wordforms. The final score is the mean of all scores per lemma. Lemmas with

[36] `https://spacy.io/`

[37] `https://pypi.org/project/classla/`

[38] `https://geographyfieldwork.com/Simpson'sDiversityIndex.htm`

single wordforms are not included in the calculation of the text's score. For building the dictionaries of lemmas and their wordforms, we used the SpaCy's[39] `es_core_news_lg` pipeline for Spanish, a fork of Stanford Stanza pipeline for (some) South Slavic languages named `CLASSLA`[40] (Ljubešić and Dobrovoljc, 2019), and Spacy's `en_core_web_lg` pipeline is used for English.

4.7.3 Inverse Simpson's Diversity

In the code used for calculating Shannon's entropy and Simpson's diversity index, Vanmassenhove et al. (2021) also calculate Inverse Simpson's Diversity by dividing 1 with Simpson's diversity score, and the mean value for all lemmas is the final result. Although Vanmassenhove et al. (2021) do not report this score in their paper, here we do report it since we calculate it along with Shannon's entropy and Simpson's Diversity. The presumable use of this metric is to make the comparison between Shannon's Entropy and Simpson's diversity easier, since these metrics move in different directions to signify the loss of morphological diversity (Shannon's entropy score goes up with morphological diversity, while Simpson's diversity index goes down with morphological diversity); or it was used during the testing of both metrics, given that they are applied for the purpose of analyzing morphological diversity for the first time.

4.7.4 Part of Speech Distribution

Another simple analysis was added to measuring morphological richness or diversity, and it is a simple approach to look into the part of speech (POS) distribution of reference and translation texts. We are interested to see if some parts of speech are over- or under-used in translations. Furthermore, a look into POS distributions can perhaps help explain some lexical diversity results.

Firstly, we count the parts of speech per text.

Secondly, we count the number of times each POS has appeared in the reference and the translation text. Using these numbers, we establish for each POS whether it appears more often in the reference or in the translation text. We also report the difference in counts between texts, based on the text where it is more common (i.e. if the translation text has the higher count for `NOUN`, we report `reference text count - translation`

[39] `https://spacy.io/`

[40] `https://pypi.org/project/classla/`

POS: *NOUN*

EP EN-ES REF count: 2 298 403
EP EN-ES TR count: 2 275 400

EP EN-ES REF normalized count = $2298403 \div 57391728 \times 100 = 4.005$
EP EN-ES TR normalized count = $2275400 \div 56667959 \times 100 = 4.015$

EP ES-EN TR > EP ES-EN REF ; add one point to EP EN-ES TR
the difference is EP EN-ES TR - EP EN-ES REF, i.e. $4.015 - 4.005 = 0.011$

Figure 9: Examples of calculation of POS distribution.

`text count`; and inversely if the reference text has a higher count we report `translation text count - reference text count`).

All part of speech tags are obtained using SpaCy[41]: using `es_core_news_lg` pipeline for Spanish, `en_core_web_lg` pipeline for English, and `hr_core_news_lg` for Croatian. This time, we used SpaCy for processing Croatian as well. The reason was speed of processing: CLASSLA seems to be much slower than SpaCy; lemmatizing texts with CLASSLA took around 4 hours. We used the same pipeline to tag parts of speech and parse, and since parsing is a much slower process than lemmatizing, we decided to use SpaCy for Croatian POS and parse analysis. At the time of conducting tests involving lemmatization, there was no SpaCy pipeline for Croatian. After testing the Spacy's lemmatizer, it seemed like SpaCy's Croatian pipeline was less accurate than CLASSLA.

Before calculating the counts, `SPACE` and `PUNCT` (punctuation) tags were removed, since we are interested only in POS tags for words. However, we do report `SYM` (symbol) tag counts. The counts were normalized by dividing the count for each POS tag with the total number of POS tags in the text and multiplying this result by 100. This way, we can compare the relative distributions of reference and translation texts. An example of calculation of distribution of POS tags per text can be seen in Figure 9.

[41] https://spacy.io/

4.8 Syntactical Diversity

For syntactical diversity, a pretty simple and naive approach was taken, inspired by Vanmassenhove et al. (2019). In this paper, the authors deal with the loss of lexical richness due to machine translation, and amongst other analyses report frequency increases and decreases in frequent and non-frequent words in the translation texts. The hypothesis is that machine translation systems will over-use frequent words, and drop the less or least frequent words due to (over)generalization. This approach is partially applied here, but on counts of parses. Parses are obtained with SpaCy's dependency parsers[42], and every translation unit is parsed separately.

Firstly, we count the number of unique parses in each text, as well as the number of parses which both texts have in common. We expect this analysis to show if there is a difference in amounts of unique parses between the reference and translation text, which can directly inform us of (potential) differences in syntactical diversity. Furthermore, by checking for how many parses do the reference and translation texts have in common, we can see how many of the parses are "taken" from the reference text, how many parses of the reference texts have been left out by the machine translation system, and also how many parses are novel as a result of the text going through the machine translation system.

Secondly, for the parses that are in common to both texts, we counted the number of times they appear in the texts in order to compare their representation and distribution. This can once again give us information of differences in syntactical diversity. For example, one question we can pose for this analysis is: do parses in common appear more frequently in the translation texts than they do in the reference texts?

Thirdly, we took the 1000 most and 1000 least common parses from the reference text, and compared their counts in the reference texts with the counts in the translation texts. This calculation is the closest to what is done in Vanmassenhove et al. (2019), and we can test their hypothesis of machine translation systems amplifying the most frequent items and dropping the least frequent items on parses. The hypothesis is the following: the 1000 most common parses will (tend to) be over-represented; while the 1000 least common parses will be under-represented in the translation.

Normalized counts of parses were used for calculating the distributions of parses in common, 1000 most frequent, and 1000 least frequent parses. To obtain a normalized

[42] https://spacy.io/usage/linguistic-features#dependency-parse

count, every parse's count was divided by the total number of parses of the text it is pertaining to, and multiplied by 100. Note that for parses, the normalization of counts was not necessary because both the reference and the translation texts have the same number of parses.

As mentioned above, all parses are obtained using SpaCy[43]: `es_core_news_lg` pipeline for Spanish,
`en_core_web_lg` pipeline is used for English, and `hr_core_news_lg` for Croatian. This time, we used SpaCy for processing Croatian as well, for the same reason explained in 4.7.4.

Punctuation (`PUNCT`) is removed from all parses in an attempt to generalize the linguistic content of the sentences, and avoid two sentences which are otherwise identical to not be considered as such due to differences in punctuation.

[43] `https://spacy.io/`

5 Analysis

In this section, the results of the work described in Section 4 will be displayed and discussed.

In all tables different translation pairs are colored with a different color, and the translation text has a lighter shade than the reference text. In most tables, results are grouped by language pair, except for the tables in Section 5.4 where they are grouped by corpora. REF indicates reference text, and TR indicates translation text. For example, in the language pair of `ES-EN`, the reference text is the test file in the language we are translating to, i.e. `test.en`. The translation file is the translation itself, i.e. `translation_es-en.en`.

Model and system are used interchangeably and refer to the machine translation model.

5.1 Translation

Translation quality was evaluated using BLEU score by `sacrebleu` (Post, 2018).

Table 5 shows BLEU scores of all trained systems, with the scores reported in Vanmassenhove et al. (2021) for reference. Please note that Vanmassenhove et al. (2021) translate the original training data to translate, and in this work we translate unseen test data.

BLEU scores on the Europarl corpus are a bit lower than results of Vanmassenhove et al. (2021), but could be considered comparable. On the other hand, BLEU scores on DGT EN-ES and ES-EN data are much higher than Europarl, and surprisingly high for EN-HR/HR-EN.

	EN-ES	**ES-EN**	**EN-HR**	**HR-EN**
Vanmassenhove et al. (2021)	40.9	41.3	–	–
EP	39.4	40.2	–	–
DGT	64.0	69.3	59.7	62.7

Table 5: BLEU scores for all recreated systems with scores reported in Vanmassenhove et al. (2021) for (approximate) reference.

5.1.1 BLEU Scores: Too good to be true?

Multiple settings for training translation systems were tested out before settling on the one reported here, and the scores for the DGT corpora were always in the same range, i.e.

much higher than for EP data. These scores might be caused by overfitting during training or the nature of the corpus (if the data is not well suited and prepared for training). The following possible causes of overfitting or data imbalances have been ruled out (to the best of our abilities):

1) Preprocessing steps, model architecture, and the training and translation procedures: are the same for all corpora used.
2) Differences in sizes of training, validation and test sets: the sizes are comparable and are split following roughly the same split, as shown in Table 2 in Section 4.3.
3) Duplicates in corpora: after training one set of systems, it was noticed that the DGT corpora have a lot of duplicates between training, validation, and test sets. An additional step was added during preprocessing to remove overlapping translation units between sets, as described in Section 4.3.2.
4) Different vocabulary sizes: vocabulary sizes of all training systems are comparable, as shown in Table 3 in Section 4.4.

Translation units in different data splits that are similar, but not identical, still might be causing overfitting and (falsely) increasing the BLEU score.[44] Python library `fuzzywuzzy`[45] was used in order to test string similarity on a random sample of 1000 sentences from the test sets, and 1000 sentences of training sets. This means that a random sentence of the test file was compared with 1) 1000 train sentences; and 2) a subset of random 1000 sentences that are comparable in length to the test sentence (+/- 5 words). Results displayed in Table 6 show that this seems to not be the culprit for such high BLEU scores since the similarity results for EP are in the same range as the results for DGT. Note that if a test sentence did not have any sentences comparable in length in the train sentence, it was not considered for calculations.

We have already stated that one feature of the raw DGT data is its repetitiveness (Section 4.2). Although we have removed all overlapping translation units between data splits, we did not remove duplicate translation units within the splits themselves, i.e. did not create unique sets of sentences for every data split. The reason for not doing this is the fear of losing too much of DGT data which is already smaller by roughly 200 000 translation units than EP. However, as we can see in Table 7, the ratio of duplicated sentences within

[44] Credit and a big thank you goes to Gorka Labaka for this idea and the code provided in order to test it.

[45] `https://pypi.org/project/fuzzywuzzy/`

Source	**Average – all**	**Average – comp.**	**Max – all**	**Max – comp.**
EP EN	85.8	52.5	100	100
EP ES	85.7	52.7	100	100
DGT EN	85.4	51.7	97	97
DGT ES	85.5	53.1	95	95

Table 6: Average and maximum score of string similarity results for testing sentences against all training sentences (all) and a subset of training sentences comparable in length (comp.).

Data split	**# of duplicates**	**% of duplicates**
EP train EN	43 178	2.89
EP train ES	41 124	2.76
DGT train EN	391 512	30.37
DGT train ES	373 136	28.94
EP test EN	9 677	2.09
EP test ES	9 340	2.02
DGT test EN	17 914	4.32
DGT test ES	16 605	4.01

Table 7: Number and percentage of duplicates per training and test sets of EP EN-ES and DGT EN-ES corpora.

splits is much larger in DGT than in EP, especially for training sets. This might be the cause of our high BLEU scores. In future work, it would be interesting to further investigate the effect of duplicates in training data on machine translation quality; and even to recreate the work outlined here and compare data with and without (a significant number of) duplicates.

5.2 Lexical Diversity

In order to analyze lexical diversity, TTR, MTLD, Yule's I, LFP, PTF, and CDU were calculated. We compare the results of translated texts with the result of the reference texts in order to assess the effect of machine translation. Note that we are not focused on comparing results of manual or human translation versus machine translation - we do not know the original language of many of our texts, and therefore cannot state whether or not they are translations. We are merely interested in the differences between the input to the machine translation system and its output. This stands true for all analyses carried out in this work.

5.2.1 TTR, MTLD, Yule's I

Results of TTR, MTLD and Yule's I for all systems trained can be seen in Table 8. The main value in a cell is the metric's score, the value in parentheses is the p-value, and the value in square brackets is the percentage of how many times the text in question scored better results. TTR was multiplied by 1000, and MTLD was multiplied by 10 000. All results reported in the table are statistically significant.

Text	**TTR * 1000**	**MTLD**	**Yule's I * 10 000**
EP EN-ES REF	8.616 (4.1e-33) [66.7%]	77.989 (3.4e-36) [66.1%]	56.669 (4.0e-14) [57.2%]
EP EN-ES TR	6.425 (4.1e-33) [32.7%]	77.063 (3.4e-36) [33.1%]	32.319 (4.0e-14) [42.8%]
DGT EN-ES REF	12.247 (5.5e-25) [62.3%]	72.350 (1.1e-21) [59.6%]	89.141 (1.1e-22) [60.4%]
DGT EN-ES TR	11.392 (5.5e-25) [37.0%]	70.941 (1.1e-21) [39.2%]	76.765 (1.1e-22) [39.6%]
EP ES-EN REF	5.674 (2.1e-10) [58.6%]	81.669 (5.4e-13) [59.4%]	26.041 (6.9e-67) [69.7%]
EP ES-EN TR	4.597 (2.1e-10) [40.9%]	77.116 (5.4e-13) [40.3%]	15.719 (6.9e-67) [30.3%]
DGT ES-EN REF	12.679 (1.0e-38) [65.8%]	99.708 (5.8e-40) [66.6%]	117.626 (3.6e-71) [74.9%]
DGT ES-EN TR	11.656 (1.9e-38) [33.0%]	95.994 (5.8e-40) [32.6%]	93.420 (3.6e-71) [25.1%]
DGT EN-HR REF	21.931 (2.67e-20) [60.3%]	381.072 (1.64e-14) [58.5%]	1076.499 (6.23e-24) [60.9%]
DGT EN-HR TR	20.187 (2/67e-20) [38.7%]	386.976 (1.64e-14) [40.6%]	894.341 (6.23e-24) [39.1%]
DGT EN-HR REF	12.720 (3.11e-68) [71.7%]	99.976 (3.51e-51) [67.8%]	118.389 (7.23e-63) [74.2%]
DGT EN-HR TR	11.712 (3.11e-68) [27.5%]	94.455 (3.51e-51) [31.2%]	95.477 (7.23e-63) [25.8%]

Table 8: Scores of TTR, MTLD, and Yule's I metrics for each model trained. REF signifies the reference text (the test split of the target language), while TR signifies translation text. For better readability, different translation pairs are colored with different colors, and the translations are indicated with lighter shades.

TTR is consistently smaller in translation texts than it is in reference texts. Furthermore, the samples of reference texts on average score higher than translation texts (see square brackets in TTR). The largest difference in TTR is between Europarl EN-ES reference and translation texts and is 2.191; and the smallest TTR difference is in DGT HR-EN reference and translation texts and is 0.558. TTR values of Spanish texts have greater variation between splits than English ones do. TTR for Croatian is the highest overall, which is not surprising given that Croatian is the most morphologically complex out of all used languages. One simple example for this is the fact that since Croatian has 7 cases, one lemma could theoretically have 14 different forms (7 forms in singular, 7 forms in plural). In reality, this usually does not happen because of the fusional aspect of Croatian morphology, but it still has a great effect on the number of types in a text. It is surprising that Spanish texts consistently have lower TTR than English texts, despite having more complex morphology. A reason for this could be the use of articles (compare "de **las** páginas 9 a 10" and "on pages 9 and 10"[46]) and other particles ("**a** final **de** mes y **a** final **de** trimestre" vs. "end-month and end-of-quarter"[47]) which lower the TTR and are used more extensively than in English. This is, however, just a first impression hypothesis and needs further research.

MTLD is lower for translation texts in all cases but one: DGT HR-EN, where the MTLD score is higher for the translation text by 5.905. The largest difference in MTLD is between DGT HR-EN (5.521); and the smallest difference is in the MTLD of EP EN-ES (0.926). It is very interesting that MTLD indicates higher lexical diversity on the translation side of DGT EN-HR; even though the reference text scored higher on the majority of samples (see square brackets in Table 8). All reference texts score higher than translation texts in the majority of samples.

The value of Yule's I is smaller for all translation texts when compared to the reference texts. Like in TTR and MTLD, the greatest difference is in DGT EN-HR with the translation text's score being lower by 182.158. The smallest difference of 10.322 is between reference and translation texts of EP ES-EN.

Figure 10 shows differences between normalized scores of the reference and translation texts per every model that was trained. The results per metric were normalized to be in range between 0 and 1, and then the score of the translation text was deducted from the score of the reference text. Translations with English as source are colored with shades of blue, while translations with English as target are colored with shades of orange. With the help of this visualization, we can notice that TTR indicates greatest losses of lexical diversity in two out of three translations with English as source. This is not unexpected given that both Spanish and Croatian are morphologically more complex than English. TTR of the remaining four translations indicates roughly the same loss. Furthermore, lexical diversity measured by TTR shows greatest loss over all translations. Regarding MTLD, the loss of lexical diversity indicated by this metric is the lowest in all three translations with English as source, with one translation (DGT EN-HR) "gaining" lexical diversity. It is also the metric that indicates least loss of lexical diversity over all translations, possibly because it supposedly does not depend so greatly on text length as TTR does. On the other hand, a metric designed to mitigate text length issues, Yule's I, indicates the greatest decrease in DGT EN-HR, with the rest of translations being around the same range

[46] Examples taken from DGT translation texts, line 11.

[47] Examples taken from DGT translation texts, line 25.

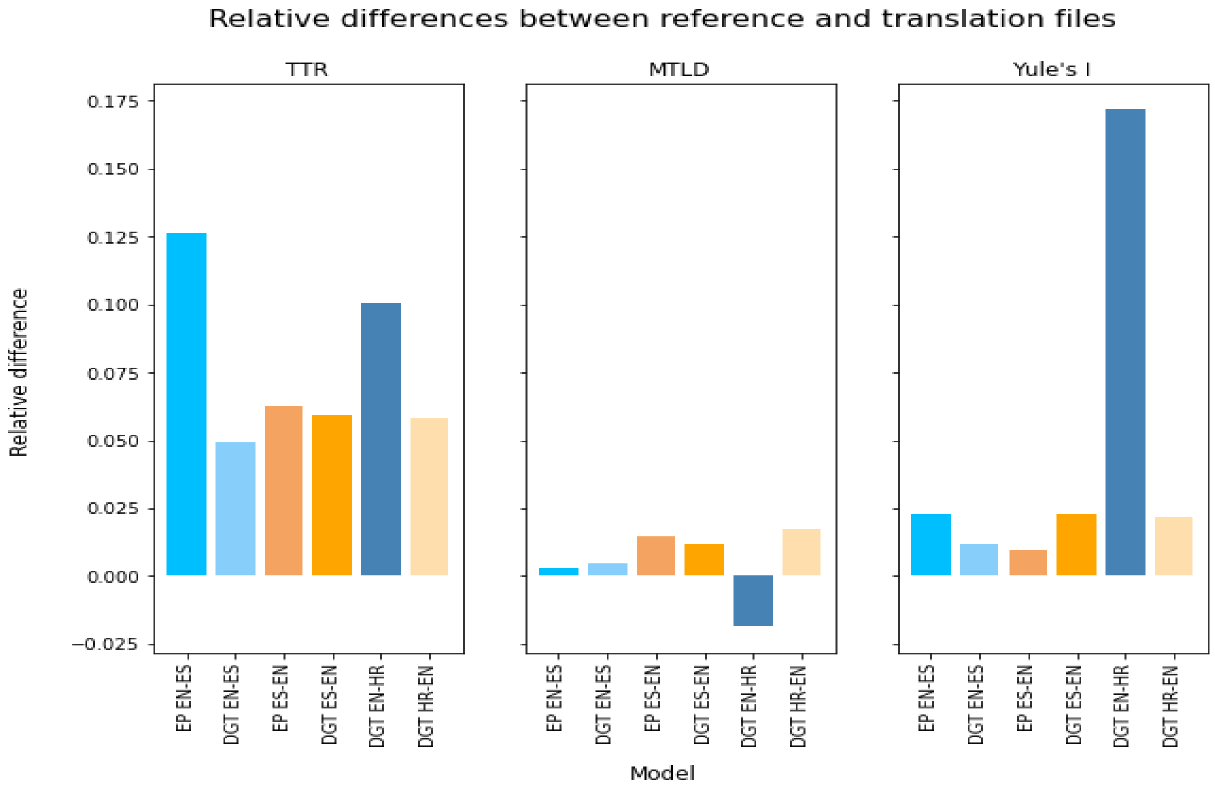

Figure 10: Normalized differences between reference and translation texts per trained model. Translations with English as source are colored with shades of blue, while translations with English as target are colored with shades of orange.

of change. With these results, it is clear that DGT EN-HR seems to be the most affected by the machine translation system, with it displaying (by far) the greatest loss in Yule's I and a gain in the MTLD score.

5.2.2 LFP

The results of LFP are reported in Table 9. As explained in Section 4.6.5, the word frequencies were calculated based on the training set of the language of the translation, i.e. the word frequencies of `train.es` were used to calculate LFP bands for both `reference.es` and `translation.es`. Band 1 stands for the 1000 most frequent words, band 2 stands for the 1001-2000 most frequent words, and band 3 is all other words.

The results show that we can see an increase of words in band 1 for all translation texts, meaning that the most common words are used slightly more frequently in the translations. Furthermore, all translations show a drop in percentage of words in band 3, meaning that there is a drop in usage of less frequent words when compared to the reference texts.

These results seem to indicate that machine translation models trained as part of this work do tend to use frequent words more frequently, possibly in place of their less frequent synonyms. A qualitative analysis in the future might help answer these questions and validate or invalidate the hypotheses laid out here.

Text	**Band 1**	**Band 2**	**Band 3**
EP EN-ES REF	77.099	7.098	15.146
EP EN-ES TR	78.224	7.182	14.082
DGT EN-ES REF	75.191	7.159	17.232
DGT EN-ES TR	75.757	7.212	16.650
EP ES-EN REF	80.630	7.140	11.667
EP ES-EN TR	81.270	7.290	10.893
DGT ES-EN REF	75.107	8.149	16.287
DGT ES-EN TR	76.115	8.180	15.275
DGT EN-HR REF	59.622	8.634	29.676
DGT EN-HR TR	60.323	8.810	28.880
DGT HR-EN REF	74.980	8.086	15.069
DGT HR-EN TR	75.980	8.130	14.102

Table 9: LFP results for reference texts and translations. All data is expressed in percentages. For better readability, different translation pairs are colored with different colors, and the translations are indicated with lighter shades.

5.2.3 Synonym Frequency

As explained in Section 4.6.6, synonym frequency is analyzed only in one direction, and the lower the score, the higher the lexical diversity quantified by these metrics.

The results of PTF and CDU can be seen in Table 10. For ease of readability, scores are multiplied by 100. PTF shows that in every translation analyzed there is an increase of primary translation frequency. This signals the tendency of machine translation systems to choose the first most frequent translation more often that they should. Similarly, all translations show a fall in CDU when compared to reference texts. EP EN-ES suffered the least lexical diversity loss measured by CDU.

Because these metrics are introduced by Vanmassenhove et al. (2021), for frame of reference we report the difference between their texts, i.e. the reference (ORIG) text and the text of translation done by a transformer model (TRANS) for Spanish: the PTF

Text	PTF ↓	CDU ↓
EP EN-ES REF	86.552	25.715
EP EN-ES TR	88.517	27.549
DGT EN-ES REF	88.813	28.084
DGT EN-ES TR	89.574	28.814
DGT EN-HR REF	83.337	40.483
DGT EN-HR TR	84.147	41.494

Table 10: Results of two synonym frequency metrics, PTF and CDU, multiplied by 100. For better readability, different translation pairs are colored with different colors, and the translations are indicated with lighter shades.

difference they report is -0.154, and for CDU it is -0.148. It is unclear whether or not the results displayed were multiplied by any factor or not; the PTF for the reference of Spanish is 9.131, for translation text it is 9.285; the CDU of reference text is 4.539, and 4.687 for the translation. Given our results, it is presumed that they multiply their results by 10, but this is not explicitly mentioned anywhere in their paper.

It is good to mention the fact that dictionaries and lemmatizers used might cause the result to falsely represent the synonym frequency of the texts. Table 11 shows some numbers to provide an insight into the data used to calculate PTF and CDU. Column "SD size" represents the size of the synonym dictionaries, i.e. the number of synonyms extracted from the bilingual dictionaries (not the number of source words for which the synonyms were extracted). Column "Unique lemmas" indicates the number of unique lemmas per reference text, and columns "Lemmas in SD" and "Lemmas not in SD" show the number of lemmas per text that were or were not represented in the synonym dictionaries. The percentages are based on the unique number of lemmas, therefore indicate the percent of unique lemmas that are (not) in the synonym dictionaries. These calculations were only carried out for reference texts. We can see that Croatian has the greatest number of lemmas found in the synonyms dictionaries, while the Spanish DGT's lemmas suffered the most due to under-representation.

File	SD size	Unique lemmas	Lemmas in SD	Lemmas not in SD
EP ES	28 738	40 218	12 003 (29.9%)	28 215 (70.3%)
DGT ES	28 738	38 212	10 122 (26.5%)	28 090 (73.5%)
DGT HR	43 904	36 989	15 363 (41.5%)	21 626 (58.5%)

Table 11: Synonym dictionary sizes, number of unique lemmas of texts and their (non)-representation in the synonym dictionaries.

EP ES: *extrañado* (∼surprised), *lisín* (Lysine), *instituimos* (we instituted), *quite* (quite), *a5-0177/2003* (—), *surquir* (to plough), *obliguemos* (we obligate (subjunctive)), *gasto-utilidad* (∼value for money), *maquinilla* (razor), *recogido* (∼collected (adj.))

DGT ES: *potentially*, *turbopropulsado* (turboprop), *национальный* (—), *daltons* (Dalton), *empujado* (pushed (adj.)), *preseleccionar* (to preselect), *resincronización* (resynchronization), *vego* (—), *boissons* (—), *y1379* (—)

DGT HR: *lepezast* (in the shape of a fan), *nepržen* (not fried), *propizamid* (Propyzamide), *prispijevati* (∼to be arriving), *dlizak* (—), *aminopiralid* (Aminopyralid), *stajanka* (∼ramp), *rasap* (dispersal), *rafinacija* (refinement), *stajaćica* (gillnet)

Figure 11: Examples of lemmas which were not part of the dictionaries and their English translation.

To further inspect how this might have influenced PTF and CDU scores, we also provide a random sample of 10 lemmas not represented in synonym dictionaries per reference text in Figure 11. In this Figure we can see that some of the lemmas not found in dictionaries are either numerical representation, not real lemmas (for example, Croatian ‘dlizak’), or are (presumably) taken literally from source text (see Spanish ‘quite‘).

We can conclude that both metrics indicate slight lexical loss in EP EN-ES and DGT EN-HR, but for DGT EN-ES the difference is very small for PTF and even indicates a gain in lexical diversity according to the CDU metric. A study of greater range would be useful in order to determine how big of a difference in PTF and CDU is relevant and can be considered important.

5.3 Morphological Diversity

5.3.1 Shannon’s Entropy, Simpson’s Diversity Index, Inverse Simpson’s Diversity Index

For the analysis of morphological diversity, we report the results of Shannon’s entropy (H), Simpson’s diversity index (D), and the inverse of Simpson’s diversity index (inverse D) in Table 12. As mentioned in Section 4.7, for H higher score indicates higher morphological diversity, while for D lower score indicates higher morphological diversity. Even though it is not specified in Vanmassenhove et al. (2021), we presume that higher Inverse D indicates higher morphological diversity, given that it is an inverse of D.

The greatest loss of morphological diversity is seen in EP EN-ES, with a difference of 3.996 points in H, 2.615 points in D, and 4.706 points in Inverse D. Furthermore, in this translation we also see the greatest difference in the number of lemmas with only one wordform. While the metrics of lexical diversity overall did not point in this direction, the results of these morphological diversity metrics might indicate that there is a correlation between a higher BLEU score and the loss of morphological diversity; or that training a model with a large portion of duplicates leads to smaller losses of morphological diversity. The fact that this is noticed in the translation from English to Spanish, but not Spanish to English, is in line with the idea that translating from a morphologically simpler to a morphologically more complex language could lead to higher losses in quality. Further research is needed to claim this with certainty, since only one of our results seems to indicate this effect and could be uncorrelated with BLEU and the possible overfitting caused by duplicates in the training data.

The results seem to confirm that these metrics do capture morphological richness or diversity given that Spanish and Croatian do have better scores in H, D, and Inverse D than English. Another interesting observation is that the DGT corpora seem to be more morphologically complex than EP.

5.3.2 Part of Speech Distribution

The distribution of POS tags is analyzed per reference-translation pair in Table 13, and for each tag we indicate if it appears more often in the reference or the translation text of the pair. Normalized counts were used and not rounded at the time of comparison, therefore results of `0.000` can be considered roughly equal and are colored green in the table. Some thought could be given to the issue of setting a threshold of which difference to consider relevant for analyses, but in this work we report and take into account all differences rounded to three decimals that are above `0.000`.

Throughout all the reference-translation pairs, adjective (ADJ), numeral (NUM) and subordinating conjunction (SCONJ) are split equally between appearing more often in references than in translations, meaning that they appear more often in three reference texts, and more often in three translation texts of all reference-translation pairs (six in total).

Text	H ↑	D ↓	Inverse D ↑	One WF Lemmas
EP EN-ES REF	28.687	81.838	130.461	24 422
EP EN-ES TR	24.691	84.450	125.755	17 683
DGT EN-ES REF	32.105	79.531	133.867	23 419
DGT EN-ES TR	31.658	79.851	133.218	20 193
EP ES-EN REF	25.119	85.043	123.246	15 019
EP ES-EN TR	25.727	84.428	124.934	11 229
DGT ES-EN REF	28.032	82.934	127.598	16 636
DGT ES-EN TR	27.318	83.457	126.877	14 263
DGT EN-HR REF	35.222	77.532	137.382	27 339
DGT EN-HR TR	35.338	77.472	137.675	24 582
DGT HR-EN REF	27.397	83.441	126.689	16 842
DGT HR-EN TR	27.865	83.114	127.051	14 132

Table 12: Results of morphological diversity metrics: H, D, and Inverse D, along with the number of lemmas with only one wordform. Results are multiplied by 100. For better readability, different translation pairs are colored with different colors, and the translations are indicated with lighter shades.

Adverb (ADV), auxiliary (AUX), interjection (INTJ), particle (PART), pronoun (PRON), proper noun (PROPN), verb (VERB), and other (X) are overall more represented in reference texts than they in translations. More specifically, ADV, INTJ, PRON, and X appear more frequently in reference texts of all reference-translation pairs, AUX, PART, and PROPN appear more frequently in four reference texts, while VERB appears more often in five reference texts.

All remaining POS tags (adposition (ADP), coordinating conjunction (CCONJ), determiner (DET), noun (NOUN), and symbol (SYM)) are represented more in the translations than they are references of the majority of the reference-translation pairs.

Interestingly, NOUN appears more often in every translation text, but PRON appears more often in every reference text. This might be indicative of the explicitation effect which is identified as one of the universals of translation in Laviosa-Braithwaite (1998).

Table 14 lists the total number of POS tags per text (spaces and punctuation excluded). The parentheses indicate the percentage of the "missing" reference words, i.e. we subtracted the number of translation POS tags from the number of reference POS tags and divided by the number of the reference POS tags. We can see that in every translation there is less POS tags/words than in the reference text, and the loss of reference text words is between

POS	EP		DGT			
	EN-ES	ES-EN	EN-ES	ES-EN	EN-HR	HR-EN
ADJ	REF 0.002	REF 0.005	TR 0.012	TR 0.013	REF 0.012	TR 0.016
ADP	REF 0.092	TR 0.053	REF 0.012	TR 0.027	TR 0.013	TR 0.028
ADV	REF 0.023	REF 0.068	REF 0.006	REF 0.026	REF 0.012	REF 0.024
AUX	TR 0.063	TR 0.007	REF 0.006	REF 0.003	REF 0.009	REF 0.006
CCONJ	TR 0.020	REF 0.006	REF 0.002	TR 0.007	TR 0.000	TR 0.003
DET	TR 0.036	TR 0.094	TR 0.039	TR 0.073	REF 0.019	TR 0.058
INTJ	REF 0.003	REF 0.001	REF 0.000	REF 0.000	REF 0.000	REF 0.000
NOUN	TR 0.002	TR 0.011	TR 0.045	TR 0.055	TR 0.049	TR 0.081
NUM	TR 0.019	TR 0.020	TR 0.001	REF 0.006	REF 0.003	REF 0.001
PART	REF 0.004	REF 0.030	TR 0.001	REF 0.002	TR 0.001	REF 0.002
PRON	REF 0.082	REF 0.011	REF 0.028	REF 0.004	REF 0.020	REF 0.034
PROPN	TR 0.042	REF 0.027	REF 0.015	REF 0.063	TR 0.002	REF 0.045
SCONJ	TR 0.010	TR 0.012	REF 0.006	REF 0.004	TR 0.004	REF 0.005
SYM	TR 0.000	TR 0.019	TR 0.001	REF 0.009	TR 0.001	REF 0.004
VERB	REF 0.007	REF 0.055	REF 0.011	REF 0.022	TR 0.005	REF 0.036
X	REF 0.000	REF 0.003	–	REF 0.004	REF 0.023	REF 0.003

Table 13: Distribution of POS tags, per tag and reference-translation pair. For every pair, it is indicated whether it is more frequent in the reference text (REF), or in the translation text (TR). Differences that, when rounded, are equal to 0.000% are colored in green.

Text	Total number of POS tags	REF - TR
EP EN-ES REF	58 104 831	723 769 (1.261%)
EP EN-ES TR	56 376 715	
DGT EN-ES REF	55 525 184	1 728 116 (2.974%)
DGT EN-ES TR	53 459 901	
EP ES-EN REF	57 391 728	2 065 283 (3.720%)
EP ES-EN TR	56 667 959	
DGT ES-EN REF	50 094 813	1 133 794 (2.263%)
DGT ES-EN TR	48 961 019	
DGT EN-HR REF	45 816 340	1 546 633 (3.376%)
DGT EN-HR TR	44 269 707	
DGT HR-EN REF	50 290 433	1 141 797 (2.270%)
DGT HR-EN TR	49 148 636	

Table 14: Total count of POS tags per text and difference between reference and translation text counts. The percentage of "missing" reference words are expressed in parentheses. For better readability, different translation pairs are colored with different colors, and the translations are indicated with lighter shades.

1 and almost 4 percent. The highest and lowest loss of words happened in the same corpus (EP), with the highest loss in the ES-EN direction, and the lowest in the EN-ES direction. This analysis shines a new light on the lexical diversity metrics discussed in Section 5.2, especially TTR which is known to be affected by the text length. We noticed a drop in TTR values for every translation text, and here we see that all translations are also shorter in terms of words than their reference texts. This could mean that the lexical richness loss measured by TTR could be even greater if TTR wasn't affected by the text's length. It would be interesting to qualitatively analyze which words got omitted in the translation. This trend could also signify another translationese universal existing in our translations, the one of simplification (Laviosa-Braithwaite, 1998) by making longer sentences shorter and possibly more robust.

5.4 Syntactical Diversity

The three analyses of syntactical diversity can be seen in Tables 15, 16, and 17.

We created a dependency parse for every translation unit using SpaCy's parsers[48], as

[48] https://spacy.io/usage/linguistic-features#dependency-parse

explained in Section 4.8.

Table 15 shows that the number of parses is lower for every translation text when compared to the reference text. The greatest difference (6632) can be seen between DGT EN-HR. This language pair in the opposite direction has the least difference in the number of parses. The consistent loss of the number of parses in translation texts might indicate a loss in syntactical diversity. One more interesting thing we can notice is the fact that the DGT corpora all have around 100 000 parses less than EP, and DGT texts have more than double the parses in common than EP texts have. These facts also reaffirm the presumed repetitiveness[49] of the DGT corpus, which persists even after the removal of duplicates, and also exists on a more general level as these parse patterns show.

Text	**Unique parses**	**Parses in common**	**Total**
EP EN-ES REF	435 196	19 193	462 711
EP EN-ES TR	428 938		
EP ES-EN REF	431 238	23 049	
EP ES-EN TR	427 235		
DGT EN-ES REF	331 030	50 861	414 590
DGT EN-ES TR	325 626		
DGT ES-EN REF	324 349	56 143	
DGT ES-EN TR	323 146		
DGT EN-HR REF	335 425	60 446	416 402
DGT EN-HR TR	328 793		
DGT HR-EN REF	325 494	55 957	
DGT HR-EN TR	325 341		

Table 15: Unique parses vs. parses in common per text. For better readability, different translation pairs are colored with different colors, and the translations are indicated with lighter shades

Table 16 shows results for the second analysis done for syntactical diversity. For every corpus, we took sentences that appear in both the reference and translation text and counted the number of times their count is higher in the reference than it is translation, and vice-versa. The scores represented in the column "Higher count" are not real counts of parses in any of the texts, but the number of times a parse that appears in both texts has a higher count in the text. This was done in the same manner as POS calculations and one example of it can be seen in Table 9 in Section 4.7.4. The results show that the

[49] Discussed in Section 4.2.

majority of parses that are in common to both the reference and the translation have the same distribution. To directly answer the question posed in 4.8 (Do parses in common appear more frequently in the translation texts than they do in the reference texts?), the shared sentences appear more frequently in all translation texts than they do in reference texts. This seems to indicate some degree of syntactical loss and could mean that systems tend to over-use those sentence structures that have a) (presumably) been seen before and b) are prompted by the sentence currently being translated (from the test dataset, i.e. the reference). A more detailed comparison between parses of the training text and the translation could shine a light on this question and the results could further exemplify this effect (or could disprove it).

Text	**Higher count**	**Equal count**	**Total number of parses**
EP EN-ES REF	1884	14 301	19 193
EP EN-ES TR	3008		
EP ES-EN REF	2168	17 816	23 049
EP ES-EN TR	3065		
DGT EN-ES REF	3402	42 807	50 861
DGT EN-ES TR	4652		
DGT ES-EN REF	4043	47 729	56 143
DGT ES-EN TR	4371		
DGT EN-HR REF	3312	52 375	60 446
DGT EN-HR TR	4759		
DGT HR-EN REF	4172	47 501	55 957
DGT HR-EN TR	4284		

Table 16: Comparison of the distribution of parses that appear in both the reference and translation text. For better readability, different translation pairs are colored with different colors, and the translations are indicated with lighter shades.

Table 17 shows the comparison between the distributions of the 1000 most and 1000 least common parses from the reference text in both the reference and translation texts. These results were calculated in the same manner as the results for POS distribution, as exemplified in Figure 9 in Section 4.7.4. The results for most common sentences show that the sentences that do not have the same distribution in both texts appear more often in all translation texts than they do in the reference text. Just like the results of sentences in common, this could indicate some degree of loss of syntactical diversity. Note that we did not remove sentences in common from the 1000 most common reference sentences and that both of these results might be pointing to the same conclusion due to this overlap.

Furthermore, models trained on EP dropped the most frequent parses far more often than models trained on DGT did. This could be caused by the nature of the DGT corpus: either its general repetitiveness, smaller syntactical diversity to begin with, or by the possible overfitting caused by duplicates in the training data. The results of the 1000 least common sentences are on par with the results Vanmassenhove et al. (2019) get for testing the loss of least frequent words caused by machine translation and show that the same effect happens in parses - in all translations, the majority of least frequent parses do not appear at all. On the other hand, a very interesting occurrence is the fact that there is if a parse that appears in both the reference and the translation text, in none of our models does it appear more often in the reference than it does in translation. This might speak to the amplification of certain features and total disregard of others; what features get amplified and what get disregarded seems to depend on the model itself.

Text	Higher count	Equal count	Not in text
1000 most common sentences			
EP EN-ES REF	372	77	0
EP EN-ES TR	479		72
EP ES-EN REF	399	77	0
EP ES-EN TR	450		74
DGT EN-ES REF	390	148	0
DGT EN-ES TR	461		1
DGT ES-EN REF	404	136	0
DGT ES-EN TR	460		0
DGT EN-HR REF	372	163	0
DGT EN-HR TR	462		3
DGT HR-EN REF	410	145	0
DGT HR-EN TR	444		1
1000 least common sentences			
EP EN-ES REF	0	21	0
EP EN-ES TR	4		975
EP ES-EN REF	0	22	0
EP ES-EN TR	5		973
DGT EN-ES REF	0	115	0
DGT EN-ES TR	6		879
DGT ES-EN REF	0	136	0
EP ES-EN TR	10		854
DGT EN-HR REF	0	147	0
DGT EN-HR TR	6		847
DGT HR-EN REF	0	142	0
DGT HR-EN TR	4		854

Table 17: Comparison of distribution of the 1000 most and 1000 least frequent parses of the reference text in reference and translation texts. For better readability, different translation pairs are colored with different colors, and the translations are indicated with lighter shades.

6 Conclusions

In this work we trained six neural machine translation systems and analyzed their output on two language pairs in both directions, as well as compared the translated texts to the source text. The translations were analyzed in terms of their lexical, morphological, and syntactical diversity or richness, totaling eight different metrics or methods of analysis. The procedure of pre- and postprocessing data, training neural machine translation systems, translating the texts, and assessing lexical and most of morphological richness is done following Vanmassenhove et al. (2021). The goal of this work was to analyze the output of machine translation systems with regard to their lexical, morphological, and syntactical richness or diversity.

We used two corpora for conducting our experiments. The first one is Europarl (Koehn, 2005), and the second one is DGT, a corpus we constructed from files of Directorate-General for Translation Translation Memory. The languages chosen for analysis were English, Spanish, and Croatian, i.e. translation pairs of English ↔ Spanish, and English ↔ Croatian. Spanish and Croatian were chosen as languages that are morphologically more complex than English and therefore considered good candidates for analyzing the effect of translation.

In most of our metrics we notice a decrease in richness or diversity caused by machine translation. Regarding lexical richness, TTR and Yule's I (the inverse of Yule's K, a measure designed to be better suited for longer texts) both indicate loss in translations for all our systems, languages and pairs, while MTLD shows loss in the majority of translations and a gain in one (DGT EN-HR). LFP, a metric usually used to score a language learner's level of the language, adapted by Vanmassenhove et al. (2021) for measuring lexical richness and further amended in this work, show that machine translated texts use frequent words more frequently than the reference texts. PFT and CDU, two metrics Vanmassenhove et al. (2021) use to score synonym frequency, both show loss in all texts and systems.

For morphological richness or diversity we used three metrics that Vanmassenhove et al. (2021) adapted for this purpose: Shannon's entropy, Simpson's diversity index and Inverse Simpson's diversity index. These metrics both measure the diversity of categorical data, with Shannon's entropy measuring the entropy of a lemma's inflectional paradigm, and Simpson's diversity index measuring the diversity of a lemma's inflectional paradigm (and its inverse does the same, but the results move in the opposite direction). We also performed part of speech analysis distribution. Shannon's entropy showed slight to no difference between reference texts and their translations, and so did Simpson's diversity index and its inverse. The latter two even showed improvement of morphological richness in two out of the six translated texts. Given that these three metrics are first used for this purpose by Vanmassenhove et al. (2021), further research needs to be done in order to determine how big of a difference can be considered relevant and meaningful when comparing texts. Part of speech distribution analysis seemed to confirm allegations of

Vanmassenhove et al. (2019) that machine translation systems use frequent items even more frequently.

For syntactical richness or diversity, we compared the distribution of parse distributions in texts of all reference-translation pairs. This analysis also confirmed the increase of frequent and decrease of infrequent items in translations as a product of machine translation.

We did not notice a difference in the effect of machine translation systems had on Spanish and Croatian texts., except in the lexical diversity metrics where some of the most dramatic changes happened in the direction of English-Croatian, but not in English-Spanish (TTR, MTLD). We did, however, notice differences between the two that are not caused by translation, but by the languages' typology. This is not directly relevant to this work, but does confirm that the metrics used do behave as expected (see Sections 5.2, LFP; and 5.3, H, D, Inverse D).

Overall, the majority of our metrics have confirmed that neural machine translation caused a decrease in lexical, morphological, and syntactical diversity. Unfortunately, in this work we did not have time to test the metrics in depth nor to qualitatively evaluate the translations. In the future, it would be beneficial to test the behaviour of many of the metrics used in this work and establish ranges of increments or decreases that can be considered relevant for each of the metrics (but especially for the novel and adapted ones). For example, it would be interesting to compare the results of Shannon's entropy and Simpson's diversity index with a qualitative analysis of certain lemmas and their wordforms. Both of these metrics indicated the average entropy or diversity of the texts' inflectional paradigms, but it not clear how the changes in these numbers actually reflect the effect of a machine translation system and to what extent. A qualitative study of neural machine translation output would certainly also be helpful in indicating what exactly is affected by the systems. This could help us answer questions such as why do all translation texts have less words than the reference texts and whether or not that is very apparent to the average reader. Furthermore, such an analysis could answer the question if the many novel parses found in translations are actually valid and would be considered natural to a native speaker of the language.

References

Mona Baker, Gill Francis, and Elena Tognini-Bonelli. *Text and technology: in honour of John Sinclair*. John Benjamins Publishing, 1993.

Marco Baroni and Silvia Bernardini. A new approach to the study of translationese: Machine-learning the difference between original and translated text. *Literary and Linguistic Computing*, 21(3):259–274, 2006.

Shoshana Blum and Eddie A Levenston. Universals of lexical simplification. *Language learning*, 28(2):399–415, 1978.

Shoshana Blum-Kulka, Shoshana Blum-Kulka, and Julian House. Shifts of cohesion and coherence in translation. 1996.

Martin Gellerstam. Translationese in swedish novels translated from english. *Translation studies in Scandinavia*, 1:88–95, 1986.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P17-4012`.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86, 2005.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180, 2007.

Moshe Koppel and Noam Ordan. Translationese and its dialects. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 1318–1326, 2011.

Svenja Kranich. Translations as a locus of language contact. In *Translation: A multidisciplinary approach*, pages 96–115. Springer, 2014.

Kristopher Kyle. Measuring lexical richness. In *The Routledge handbook of vocabulary studies*, pages 454–476. Routledge, 2019.

Batia Laufer. The lexical profile of second language writing: Does it change over time? *RELC journal*, 25(2):21–33, 1994.

Batia Laufer and Paul Nation. Vocabulary size and use: Lexical richness in l2 written production. *Applied linguistics*, 16(3):307–322, 1995.

Sara Laviosa-Braithwaite. Universals of translation. *Routledge encyclopedia of translation studies. London: Routledge*, pages 288–291, 1998.

Gennadi Lembersky, Noam Ordan, and Shuly Wintner. Language models for machine translation: Original vs. translated texts. *Computational Linguistics*, 38(4):799–825, 2012.

Nikola Ljubešić and Kaja Dobrovoljc. What does neural bring? analysing improvements in morphosyntactic annotation and lemmatisation of Slovenian, Croatian and Serbian. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 29–34, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3704. URL `https://www.aclweb.org/anthology/W19-3704`.

Philip M McCarthy. *An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (MTLD).* PhD thesis, The University of Memphis, 2005.

Philip M McCarthy and Scott Jarvis. Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42 (2):381–392, 2010.

Michael P Oakes and Meng Ji. *Quantitative methods in corpus-based translation studies: A practical guide to descriptive translation research*, volume 51. John Benjamins Publishing, 2012.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL `https://aclanthology.org/W18-6319`.

Georg Rehm, Hans Uszkoreit, Ido Dagan, Vartkes Goetcherian, Mehmet Ugur Dogan, and Tamás Váradi. An update and extension of the meta-net study “europe’s languages in the digital age”. 2014.

Diana Santos. On grammatical translationese. In *Short papers presented at the Tenth Scandinavian Conference on Computational Linguistics*, pages 59–66, 1995.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Edward H Simpson. Measurement of diversity. *Nature*, 163(4148):688–688, 1949.

Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 454–459, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2012/pdf/814_Paper.pdf`.

Elke Teich. *Cross-linguistic Variation in System and Text: A Methodology for the Investigation of Translations and Comparable Texts.* Text, translation, computational processing. Mouton de Gruyter, 2003. ISBN 9783110176155. URL `https://books.google.es/books?id=wxjZO8\_sn68C`.

Sonja Tirkkonen-Condit. Translationese—a myth or an empirical fact?: A study into the linguistic identifiability of translated language. *Target. International Journal of Translation Studies*, 14(2):207–220, 2002.

Gideon Toury. Descriptive translation studies: And beyond. *Descriptive Translation Studies*, pages 1–366, 2012.

Rita Vanderauwera. *Dutch Novels Translated Into English: The Transformation of a "Minority" Literature.* Approaches to translation studies. Rodopi, 1985. ISBN 9789062038473. URL `https://books.google.es/books?id=8yJcAAAAMAAJ`.

Eva Vanmassenhove, Dimitar Shterionov, and Andy Way. Lost in translation: Loss and decay of linguistic richness in machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 222–232, Dublin, Ireland, August 2019. European Association for Machine Translation. URL `https://aclanthology.org/W19-6622`.

Eva Vanmassenhove, Dimitar Shterionov, and Matthew Gwilliam. Machine translationese: Effects of algorithmic bias on linguistic complexity in machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2203–2213, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.188. URL `https://aclanthology.org/2021.eacl-main.188`.

George Udny Yule. The statistical study of literary vocabulary. 1944.

Mike Zhang and Antonio Toral. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5208. URL `https://aclanthology.org/W19-5208`.