Supervisor	Prof. dr. Lieve Macken
	Department of Translation, Interpreting and Communication
Cosupervisors	Prof. dr. Orphée De Clercq
	Dr. Arda Tezcan
	Department of Translation, Interpreting and Communication
Dean	Prof. dr. Gita Deneckere
Rector	Prof. dr. Rik Van de Walle

Syntactic Difficulties in Translation

Dissertation submitted in partial fulfilment
of the requirements for the Degree of

Doctor in Translation Studies

at Ghent University

Bram Vanroy

2021

GHENT
UNIVERSITY

FACULTY OF ARTS
AND PHILOSOPHY

Voor B&B

Abstract

Even though machine translation (MT) systems such as Google Translate and DeepL have improved significantly over the last years, a continuous rise in globalisation and linguistic diversity requires increasing amounts of professional, error-free translation. One can imagine, for instance, that mistakes in medical leaflets can lead to disastrous consequences. Less catastrophic, but equally significant, is the lack of a consistent and creative style of MT systems in literary genres. In such cases, a human translation is preferred.

Translating a text is a complex procedure that involves a variety of mental processes such as understanding the original message and its context, finding a fitting translation, and verifying that the translation is grammatical, contextually sound, and generally adequate and acceptable. From an educational perspective, it would be helpful if the translation difficulty of a given text can be predicted, for instance to ensure that texts of objectively appropriate difficulty levels are used in exams and assignments for translators. Also in the translation industry it may prove useful, for example to direct more difficult texts to more experienced translators.

During this PhD project, my coauthors and I investigated which linguistic properties contribute to such difficulties. Specifically, we put our attention to syntactic differences between a source text and its translation, that is to say their (dis)similarities in terms of linguistic structure. To this end we developed new measures that can quantify such differences and made the implementation publicly available for other researchers to use. These metrics include word (group) movement (how does the order in the original text differ from that in a given translation), changes in the linguistic properties of words, and a comparison of the underlying abstract structure of a sentence and a translation.

Translation difficulty cannot be directly measured but process information can help. Particularly, keystroke logging and eye-tracking data can be recorded during translation and used as a proxy for the required cognitive effort. An example: the longer a translator looks at a word, the more time and effort they likely need to process it. We investigated the effect that specific measures of syntactic similarity have on these behavioural processing features to get an indication of what their effect is on the translation difficulty. In short: how does the syntactic (dis)similarity between a source text and a possible translation impact the translation difficulty?

In our experiments, we show that different syntactic properties indeed have an effect, and that differences in syntax between a source text and its translation affect the cognitive effort required to translate that text. These effects are not identical between syntactic properties, though, suggesting that individual syntactic properties affect the translation process in different ways and that

not all syntactic dissimilarities contribute to translation difficulty equally.

Samenvatting

De kwaliteit van machinevertaalsystemen (MT) zoals Google Translate en DeepL is de afgelopen jaren sterk verbeterd. Door alsmaar meer globalisering en taalkundige diversiteit is er echter meer dan ooit nood aan professionele vertalingen waar geen fouten in staan. In zekere communicatievormen zouden vertaalfouten namelijk tot desastreuse gevolgen kunnen leiden, bijvoorbeeld in medische bijsluiters. Ook in minder levensbedreigende situaties verkiezen we nog steeds menselijke vertalingen, bijvoorbeeld daar waar een creatieve en consistente stijl noodzakelijk is, zoals in boeken en poëzie.

Een tekst vertalen is een complex karwei waarin verschillende mentale processen een rol spelen. Zo moet bijvoorbeeld de brontekst gelezen en begrepen worden, moet er naar een vertaling gezocht worden, en daarbovenop moet tijdens het vertaalproces de vertaling continu gecontroleerd worden om te zorgen dat het ten eerste een juiste vertaling is en ten tweede dat de tekst ook grammaticaal correct is in de doeltaal. Vanuit een pedagogisch standpunt zou het nuttig zijn om de vertaalmoeilijkheid van een tekst te voorspellen. Zo wordt ervoor gezorgd dat de taken en examens van vertaalstudenten tot een objectief bepaald moeilijkheidsniveau behoren. Ook in de vertaalindustrie zou zo'n systeem van toepassing zijn; moeilijkere teksten kunnen aan de meest ervaren vertalers worden bezorgd.

Samen met mijn medeauteurs heb ik tijdens dit doctoraatsproject onderzocht welke eigenschappen van een tekst bijdragen tot vertaalmoeilijkheden. We legden daarbij de nadruk op taalkundige, structurele verschillen tussen de brontekst en diens vertaling, en ontwikkelden verscheidene metrieken om dit soort syntactische verschillen te kunnen meten. Zo kan bijvoorbeeld een verschillende woord(groep)volgorde worden gekwantificeerd, kunnen verschillen in taalkundige labels worden geteld, en kunnen de abstracte, onderliggende structuren van een bronzin en een vertaling vergeleken worden. We maakten de implementatie van deze metrieken openbaar beschikbaar.

De vertaalmoeilijkheid van een tekst kan niet zomaar gemeten worden, maar door naar gedragsdata van een vertaler te kijken, krijgen we wel een goed idee van de moeilijkheden waarmee ze geconfronteerd werden. De bewegingen en focuspunten van de ogen van de vertaler en hun toetsaanslagen kunnen worden geregistreerd en nadien gebruikt in een experimentele analyse. Ze geven ons nuttig informatie en kunnen zelfs dienen als een benadering van de nodige inspanning die geleverd moest worden tijdens het vertaalproces. Daarmee leidt het ons ook naar de elementen (woorden, woordgroepen) waar de vertaler moeilijkheden mee had. Als een vertaler lang naar een woord kijkt, dan kunnen we aannemen dat de verwerking ervan veel inspanning vergt. We kunnen deze gedragsdata dus gebruiken als een maat voor moeilijkheid. In ons

onderzoek waren we voornamelijk benieuwd naar het effect van syntactische verschillen tussen een bronzin en een doelzin op dit soort gedragsdata.

Onze resultaten tonen aan dat de voorgestelde metrieken inderdaad een effect hebben en dat taalkundige verschillen tussen een bron- en doeltekst leiden tot een hogere cognitieve belasting tijdens het vertalen van een tekst. Deze effecten verschillen per metriek, wat duidt op het belang van (onderzoek naar) individuele syntactische metrieken; niet elke metriek draagt even veel bij aan vertaalmoeilijkheden.

Acknowledgements

A lot has changed in the last couple of years, so much so that sometimes none of this feels real. I am thankful for the opportunities that were given to me and for all the people that touched my life in any way. While working on this PhD, I have had the privilege to enjoy the love and support of people that I would like to extend my gratitude to. With a cliché: I couldn't have done it without you.

Prof. dr. Lieve Macken. Lieve, the patience that you have is unlike any I have ever seen. You gave me the freedom to wander off into whatever direction that had caught my interest on any given week – probably something related to neural networks or linguistic trees... But when I wandered too far, you were there to help me re-focus my mind and support me along the way. I learnt so much during this PhD, precisely because you allowed me the liberty to discover new things. Not in the least, I learnt *from* you. I admire you, your work ethic, and the way in which you lead, support and work with "your team" specifically, and LT3 as a whole. Thank you for guiding me through this adventure.

Prof. dr. Orphée De Clercq and dr. Arda Tezcan. The PhD gods blessed me with not one but with two cosupervisors on my quest. Working on a topic that touched on many related fields, I am thankful for the insights I gained from your expertise, for your critical reflections and continuous feedback. For the patience that you had when I worked (too) close to a deadline. For the times that you calmed me down when I was fed up with responding to reviews, was stuck with a programming problem, or had yet another writer's block. Thank you for always having my back.

Dr. Vincent Vandeghinste. As part of my doctoral advisory committee, we met once a year where you guided me, suggested new ideas, and – perhaps most importantly – always spoke your mind. Your role in me becoming a doctor is much greater than that, however. What feels like an age ago (allow me some exaggeration while I near my 30s), you were the supervisor of my internship in Artificial Intelligence. I remember well how you suggested the position for a PhD at Ghent University. Perhaps unknowingly you boosted my confidence greatly, and I do not think I would have ever had applied for the position were it not for the faith that you put in me. In a melancholic sense, not much has changed between now and the time at the CCL – and in a more real sense *a lot* has changed. I still enjoy working with linguistic trees, and that will never change. Perl, I do not miss so much... Thank you for the part that you have played in what now is my future. I am indebted to you.

Dear colleagues of LT^3. It was my pleasure to work with you. A team where everyone has their own speciality but where knowledge and skills are shared. Thank you for the laughs and talks, and the frequent pats on the back. I wish you all the best in your future endeavours.

In my publications, I have had the continuous support of my supervisors, but I've had the pleasure of working with dr. Joke Daems and dr. Moritz Schaeffer as well. Thank you for your understanding whenever I came up with alternative ideas or modifications to my dataset, and thank you for your patience to educate me on topics that were new to me.

A special thanks to Ann Van Daele and Chantal Van de Veire for always providing calm yet efficient help with anything related to administration, going from vending machine issues to missing payment information. Also my gratitude to the unnamed employers of Ghent University Library who must have scanned dozens of book chapters for me to use during this PhD. You are a lifesaver.

My family. It feels surreal for me to be writing a, chronologically, last section in a PhD thesis. Who would have imagined ten years ago that this is what I would do and would love doing? *The only reason that I am here today, is you.* Because of your unrelenting support, your ever-lasting curiosity, and your inexhaustible enthusiasm. My gratitude and unconditional love are forever yours.

Finally, my friends. I appreciate your understanding for all those times where I was absent, where you did not ask questions, and yet always encouraged me with positivity. Thank you for lifting me up whenever I got down.

Contents

Chapter 1

Introduction

1.1 Preamble

The digital age has facilitated communication to an astounding level. We are at a point in time where we can use machine translation (MT) relatively reliably to get the gist of a text in many different languages, have PowerPoint subtitle our presentation in real time in a different language, and we can even install apps on our phone that can serve as a multilingual interpreter, listening to and translating what is being said. Whereas these examples do a good job of making translation tools available to the masses, no guarantee of quality is given. If a translation contains an error, then that is almost expected, “it is a machine after all”. One can easily imagine specific cases or general domains where mistranslation can lead to catastrophic consequences such as the translation of legal, medical, and technical documentation. In other scenarios, such as literary translation, poor translations miss their mark of conveying an idea or atmosphere and can destroy the immersion that the original text was striving for. There is no question about it: machine translation has its flaws and in many cases a post-editor is required who makes changes to the MT version, or preferably, if the available resources allow it, a human translator should translate the text from-scratch. Expert translators are needed, even in this digital age.

Translation, human or otherwise, comes with its own set of problems. Most notable for this thesis is the difficulty of translation and possible errors that may result from it. Professional translators would undoubtedly agree that some texts or even specific constructions are more difficult to translate than others. Being able to predict the translation difficulty (or *translatability*) of a source text (ST) before translating would have some considerable applications. It could be a helpful tool in both educational and production environments: (i) in (machine) translation research to select texts of similar or contrasting translation difficulty; (ii) in the translation industry to deliver source texts to translators with the appropriate expertise; (iii) in an educational environment where texts of a relevant difficulty level needs to be attained. For instance during exams and assignments to ensure that all students receive texts of similar translation difficulty. Unfortunately such a general-purpose system does not exist. However, in research focused on machine translation, Underwood and Jongejan (2001) proposed a statistical tool that can highlight possibly problematic elements in a source sentence that may be difficult for MT. Features such as “long sentence of more than 25 words” or “one or more nominal compounds” are used to this end. (It should be noted that this tool dates from 2001 and that it probably is not applicable to contemporary MT systems.) In this paper we are only interested in human translation, however.

This PhD project set out to contribute to the field of translatability. Initial goals were conceived, and due to follow-up research findings, and perhaps particularly my own interests, the course of the project changed over time. At the start of the project, the intent was to create a translatability system that could quantify the translation difficulty of a given text as well as highlight specific constructions in that text that contribute most to translation issues. However, it quickly became clear that translation difficulty is not easily reducible to only specific constructions or language properties (as Chapter 2 will show). Translation difficulty is influenced by a range of factors and I chose to take a deep dive into a couple of them. Therefore, a translatability prediction system has not been created and instead more time was dedicated to lay out the syntactic, methodological foundation for such a tool.

In what follows a brief overview of the individual papers and book chapter is given. These publications follow each other chronologically and they each start where the previous left off. The works themselves are provided in full as they were published (or submitted) in the following chapters. Small (formal) changes have been made for consistency purposes or to add glossary references, but the content itself was not altered. I apologise in advance to the reader: because these publications all touch on the same topic, overlap is inevitable. Particularly literature studies and data set descriptions will be similar across studies.

Chapter 2: Correlating Process and Product Data. Keeping in mind that the initial goal was to create a translation difficulty prediction system that *exclusively* makes use of source text features, we first tried to answer the question How does translation process data correlate with the translation product? The crux here, and in what follows, is that translation process measures (such as eye-tracking data) are often used in the field as a proxy for cognitive effort. By analysing someone’s behaviour during a task, you can get a good idea about the cognitive effort that was required and, by extension, the difficulties that they were facing. The idea behind this question is that if we know which product features correlate with cognitive effort (difficulty), then we can try to model and use such features as part of a translatability prediction system. The product-based measures that we used were number of annotated translation errors, word translation entropy (certainty of a lexical translation choice), and word reordering. Those measures were correlated with process features of three different categories: keystroke information, eye-tracking data, and duration. As a pilot study of sorts we investigated whether we could confirm the findings of related research where a relationship was proven between product-based measures and data of the translation process.

Weak, but significant, correlations were found. Low correlations are no surprise: the mental processes involved in translation are complex and highly interactive, and high correlations are hard to expect. Significant correlations for word translation entropy and syntactic equivalence indicated to us that these features could be used as difficulty predictors in future work. In this and the next chapter, "syntactic equivalence" is reduced to a single facet of syntax, namely word (group) order. Also see the literature section (Sec. 1.2.1) below on equivalence.

Chapter 3: Predicting Syntactic Equivalence. The results of Chapter 2 suggested that syntactic equivalence, which in this case means "word reordering", is correlated with cognitive effort (as other research also confirmed). It would therefore make a fitting feature for a translatability system. Put differently: if we know before translating that a source sentence will need a lot of word reordering, we can assume that it will be more difficult to translate. We created two new metrics to measure this reordering on the sentence-level: `word_cross` and sequence cross. The former is different from the word reordering metric used in Chapter 2 (bidirectional and absolute instead of directional and relative, as the paper will explain), and the latter of which is word *group* based. The (applied) research question in this article is: How well can a machine learning system predict word (group) reordering by only making use of source text information? Because we know from Chapter 2 that word reordering correlates with cognitive effort, it would be helpful if we could predict such a feature to eventually use in a translation difficulty prediction system. We therefore built different machine learning systems and trained them on a large parallel corpus of Dutch and English to predict such word (group) reordering. The best system, a neural architecture that combines semantic and morphosyntactic features, achieved a moderate Pearson r correlation of 0.54 for word reordering and 0.58 for word group reordering. These results made it clear to us that word (group) reordering for a given language pair is predictable, even if only the source text is available. But surely, syntactic equivalence is more than reordering alone?

Chapter 4: Metrics of Syntactic Equivalence. In this book chapter, we built on the previous work and introduced different measures to calculate the difference between syntactic properties of a source sentence and its translation. The emphasis lied on trying to disentangle the broad concept of syntax into more specific syntactic properties. More formally, Which fine-grained metrics can quantify syntactic divergences between a source and target text on the sentence level? and additionally, What is the effect of sentence-level, syntactic metrics on process data? First, the word groups in Chapter 3 were determined merely by constraints on word alignment. These groups were therefore not guaranteed to be linguistic entities. In Chapter 4, we further restrict those word groups to ensure that a word group must consist of a meaningful linguistic unit as determined by the word's position in their linguistic, structural tree. Second, we measure the changes in dependency label (e.g. subject, adverbial clause) for a word compared to its translated word(s). The last, but most complex, measure, is ASTrED (Aligned Syntactic Tree Edit Distance). It compares the abstract, linguistic structure of the source sentence to its translation. Similar to Chapter 2, we related these product features to process features. We found that, on the sentence level, label changes had a significant, positive effect on the production duration.[1] In addition, linguistically motivated word group reordering positively and significantly affect the total reading time on the source sentence. So, the more divergent a source sentence is from its translation in terms of linguistic labels and word group order, the more cognitive effort is required to translate it.

Chapter 5: The Effect of Product-based Metrics. Up to this point, our previous work focused on the sentence as a single unit. We were particularly interested in which syntactic, quantifiable phenomena lead to difficulties when translating a sentence. However, a fine-grained approach is bound to provide a more detailed insight of specific difficulties on smaller linguistic units. With a translatability prediction system in mind, word-level features would be helpful because they would allow a machine learning system to better identify sub-sentential difficulties that translators may be faced with in addition to coarse-grained, sentence-level problems. Rather than aggregating our metrics on the sentence level, as we did in the previous chapters, Chapter 5 re-implemented all the previous work on the word level. In succession to the previous chapter, the research question to answer was What is the effect of word-level, syntactic product-based metrics on process data? Particularly, we measured the effect of word (group) reordering and the structural comparison metric ASTrED alongside other existing metrics on different stages of the translation process. Significant, positive effects were found, indicating the impact of word (group) reordering and structural divergences on translation difficulty. This suggests that if a given word (or its group) needs to be reordered, the difficulty to translate that word will be higher. Similarly, higher cognitive effort can also be observed when the position of a word in its respective linguistic tree differs from its related position in the target tree.

Before presenting these publications in full, three more preliminary sections will be given. In Section 1.2, a summary of the most related research is given. Next, the data sets that were used in different chapters will be introduced (Sec. 1.3). Even though these are described in their respective publications as well, it is worth condensing them in one place for reference. Finally, an extensive section is preserved for the description of the open-source library that is the culmination of this PhD project (Sec. 1.4). In that section, the library itself, its uses, and the metrics involved are exemplified and explored. The section brings together the different metrics that have been introduced over time in Chapters 3, 4, and 5.

[1] Positive, or negative, effects in this thesis always refer to the statistical concept where an effect indicates that as a given predictor variable increases, the dependent variable increases or decreases in a significantly meaningful way, respectively.

1.2 Literature

Even though the literature sections of the following chapters each refer to topic-specific related research, three specific topics deserve emphasis. First and foremost, the concept of equivalence has a special meaning in TS, but also in (computational) linguistics. It is therefore necessary to explain how it is used in this thesis. Second, a general section discusses (mostly theoretical) approaches and models of how language is represented in the mind, particularly in light of translation and bilingualism. In a final section, specific research on translatability is summarised to illustrate how it has been explored. This section shows overlap with the literature sections in the separate publications.

1.2.1 Equivalence

In the early chapters that follow, the term equivalence is discussed in a limited context within Translation Studies (TS) (Sec. 2.2.3.2, Sec. 3.2), but it is never clarified in light of my own research. As it might not be clear to the reader what is intended with "(syntactic) equivalence" in the backdrop of Translation Studies, this section hopefully disambiguates its meaning for the remainder of this work. Below I will first provide a non-exhaustive overview of how the concept has been notably studied in TS and end with a motivation of how I intend to use it throughout this thesis. For a more complete discussion, see the work by Panou (2013) and Pym (2014), which served as a reference for this section.

As early as 1958, Vinay and Darbelnet mention equivalence to mean a semantic correspondence where the meaning representation of a situation is the same for both the source and target expressions. From the translated edition in 1995: "The equivalence of the texts depends on the equivalence of the situations" (Vinay & Darbelnet, 1995, p. 5). As one of seven methods of translation (Vinay & Darbelnet, 1995, p. 30 and onwards;), equivalence aims to transfer the meaning of one text to its translation as closely as possible, even if that implies the complete abandonment of the original style or form. Idioms, for example, illustrate such an equivalence well, where a direct (rather literal) translation often does not have the desired meaning in the target text (TT).

From the point of Structural Linguistics, where form and meaning are deeply intertwined, Jakobson argues that full equivalence rarely exists in translation "between code-units" (Jakobson, 1971, p. 261; originally published in 1959), similar to how synonymy is never *complete* equivalence because of that intimate relationship between a (lexical or linguistic) item and its meaning. That does not mean, however, that two texts cannot be equivalent: when translators choose to focus on the transfer of meaning of the whole source message, rather than systematically translating such "code-units" individually, a (non-formal) equivalent target text can be created. This interpretation is largely similar to Vinay and Darbelnet (1995) above.

In a more strict sense than Jakobson (1971), Nida (1964) claims that there can be “no fully exact translation” (p. 156). The job of the translator, then, is to find the translation that is as close as possible to the original, but this closeness (or sameness) can be approached from different angles, and is determined by what the translation is intended to achieve. Formal equivalence, according to Nida, emphasises both form and content and is for instance typical when translating poetry or when one is concerned with the one-on-one correspondence of sentences and concepts. On the other end of the scale lies dynamic equivalence, where a translator focuses on the “naturalness of expression” (p. 159) and conveys the source meaning into the target context and culture. This type is closer to what Vinay and Darbelnet and Jakobson understand as equivalence, with a focus of a meaning equivalent of the whole message instead of a formal one.

Observing translation as a (formal) linguistic phenomenon, Catford distinguishes between textual equivalence and formal correspondence. Whereas (textual) equivalence is similar to (dynamic) equivalence as it has been discussed so far, formal correspondence is the language-systematic comparison of a source language unit to its translated unit, or put differently “[a] formal correspondent [...] can be said to occupy, as nearly as possible, the ‘same’ place in the ‘economy’ of the TL [target language] as the given SL [source language] category occupies in the SL” (p. 27). Crucial to this thesis, Catford continues to discuss two major types of translation shifts where a translator deviates from formal correspondence. In level shifts, a linguistic property of the source text is translated by means of a characteristic on a different level, typically a shift from grammar to lexis or vice-versa. Such shifts are for instance used when the valence or implication of a verb cannot easily be transposed grammatically into the target language, requiring a lexical change or addition instead. In addition, four types of category-shifts are discussed. Structure-shifts are cases where the linguistic structure of the target text has changed compared to its original. This involves alterations to language structure across different levels (such as phrases and clauses) as well as word (group) order changes. Class-shifts occur when the classification of a given linguistic unit (such as a word) differs from its translated unit, for instance when its word class or function in the sentence has changed. When the linguistic unit in the source text differs from the unit of its translation (e.g. word-to-phrase), a unit-shift has taken place, for instance when a single lexical item is translated as a phrase. Finally, intra-system shifts describe those cases where the two language systems involved correspond formally, i.e. their monolingual language rules are approximately similar, but where a correct translation requires non-correspondence. As an example, Catford refers to plural versus singular surface forms of the same concept (EN “the dishes” vs. FR “la vaisselle”; p. 80) or the explication of articles (EN “He is *a* teacher” vs. FR “Il est professeur”; p. 81). Panou (2013) notes that Catford was criticised for his pinpoint-focus on the almost exclusive role of linguistics on translation with disregard for the importance of extra-textual context.

Baker (2011, first published in 1992) tears open equivalence and uses it in a very broad context to touch upon a variety of levels in which equivalence can be present. She states that equivalency relations can exist on the (sub-)word level, in syntax and grammar, semantics, pragmatics and beyond. For this literature overview, especially her discussion on grammatical equivalence is relevant, which is related to the structure and unit shifts of Catford, above. Particularly, nonequivalence of grammar systems (morphology and syntax) can, perhaps unsurprisingly, pose difficulties for translators. If the SL has grammatical rules or categories that are not present in TL (e.g. concerning gender, number, voice), the translator has to add or remove additional linguistic units (morphological or lexical) in the target text to compensate for this difference in expressiveness.

Pym (2014, first published in 2010), finally, considers two opposing paradigms of equivalence in translation literature. On the one hand, directional equivalence is based on the assumption that translation is not a symmetric relationship between a source text and its translation (Pym, 2014, Chapter 3). In other words, when back-translating a translation, the generated text will not have the same equivalence relation to TT as TT has to the original ST. In this light, Pym refers to the work of Kade (1968) and his different types of equivalence on the word and phrase level. One-to-one (or “total”) equivalence is relatively rare and restricted to technical terms. This type of relationship holds in two directions and is therefore more keen to natural equivalence, which will be mentioned next. One-to-several equivalence occurs when translators have a number of alternatives to choose from, and one-to-part where a translation equivalent only covers part of the source concept. Lastly, one-to-none occurs when a translation is simply not available and the creation of a neologism or borrowing of an existing term in another language is necessary. Natural equivalence (Pym, 2014, Chapter 2) dictates that, in principle, “the things of equal value are presumed to exist *prior* to anyone translating” (p. 6). In other words, language systems have corresponding tools to their disposal to make the same meaningful statements. Note that such a view clashes with aforementioned (structuralist) convictions by Jakobson (1971) and Nida (1964) that posit that the lexical surface forms of different languages *per definition* cannot be equivalent because language, in structuralism, is a realisation of a world view which inherently must be different between languages and cultures. Furthermore, natural equivalence does not depend on directionality but rather the equivalency relationship between ST and TT holds regardless of the translation direction. From a product-oriented, methodological viewpoint, the operationalisation of syntax in this thesis is most related to natural equivalence. *Not* because I am of the conviction that only one correct translation can ever exist but because given a final product (i.e. the given translation), the quantifiable properties in my view are not dependent on the translation direction. This is also in line with the interpretation of equivalence in Formal Language Theory (FLT).

(Formal) Grammars in FLT are considered weakly equivalent if (and only if) they generate the same string language and strongly equivalent if and only if they have the generative power to create the same tree language (Chomsky, 1963, p. 395; and Bod (1998) for an overview). As such, equivalence in FLT indicates a mathematical and algorithmic property. I use equivalence in a similar way applied to syntax to mean the relationship between a source unit and an aligned target unit (word, phrase, sentence etc.) that can be measured by means of quantifiable properties such as the suggested metrics introduced in this thesis. These metrics are independent of the translation direction, which follows Pym's natural equivalence. As is hopefully clear by now, literature tends to focus greatly on the importance of equivalence of meaning whereas I almost exclusively emphasise the "relevant similarity" (Chesterman, 2011, p. 26) between syntactic structures – as one does in FLT. In the case of the suggested metrics lower values indicate a higher equivalence, or – synonymous, here – higher similarity. Note that in such an interpretation a syntactically equivalent, and hence syntactically literal, translation is not necessarily a correct one, neither semantically nor grammatically.

Early chapters in this thesis will focus on (my quantification of syntactic) equivalence between a source text and its translation, and gradually the related concept of literal translation takes the spotlight. Literal translation can be operationalised in very specific ways. Important to repeat, though, is that literal translation is often thought of as the early, default mode of translation, or as Balling et al. (2014) put it: "literal translation is likely to be a universal initial default strategy in translation" (p. 234). The metrics that will be introduced in this thesis all, in a specific but practical sense, model parts of the syntactic component of literal translation. Put differently, by quantifying (syntactic) differences between a source text and its translation, we are also measuring how close the translation is to a literal alternative, i.e. how syntactically equivalent they are. This will be explained in greater detail in Chapters 4 and 5.

1.2.2 Bilingual Models of the Mind

Although each of the following chapters deals with a specific angle of the translation process, mental processes involved, and the differences with other language-involved tasks, the current section will address different dichotomies of translation as a means to set the stage on which the following chapters will be played. Particular attention will be paid to the historical difference between horizontal and vertical processing.

While discussing the processes of interpreting and translation, Seleskovitch (1976) compares two different approaches of reformulation, namely code switching versus the transfer of ideas. Code switching here refers to the almost purely technical task of switching a set of sequential symbols in one language to a corresponding sequence in another. In the extreme case, understanding the complete sequence is not a requirement to substitute the source symbols for the target symbols (although there is no guarantee that that leads to a correct translation). On the other end of the spectrum, deverbalization of the meaning on the source side is key. The relevance of surface forms of the source and target text (or speech) are superseded by the transfer of semantics and prosody. Either the symbols are translated between one language and the other (like a dictionary where one consciously goes from the source lemma to the target translation), or the source message is first understood and deverbalized before being transferred to the target language (TL). Seleskovitch stresses that translation and interpreting are not to be identified as these extremes, but rather each individual instance of such a task is positioned on the continuum between the end points (p. 96). de Groot (1997) describes two different points of view that she calls vertical and horizontal translation. In the vertical view, the different stages in the translation process are considered self-contained in terms of the activated language. Particularly, the source language text needs to be understood within its context and pragmatic intention, after which the deconstructed meaning can be reformulated in TL. This is similar to the deverbalisation process of Seleskovitch (1976), above. In contrast, the horizontal view holds that translation is a transcoding process where specific units in the source text (words, phrases, clauses) are replaced by their TL equivalent, similar to what Seleskovitch called code-switching. In such an interpretation, de Groot suggests, translated words have a shared representation in the bilingual lexicon of the translator. It follows, then, that during translation the two involved language systems are co-activated, because lexical entries are shared in memory. Although a vertical orientation is often strongly defended, "[e]ven strong advocates of the vertical view acknowledge that translation involves some horizontal processing" (de Groot, 1997, p. 30-31). Even though Seleskovitch (1976) does not discuss co-activation and bilingual, shared representations, de Groot notes that Seleskovitch is one of many "advocate[s] of the vertical view" (p. 31), who consider the parallel process to be inferior, because in such a view the target text is (or can be) influenced too greatly by the source text through shared (active) representations.

de Groot (1997) was definitely not the first to suggest that representations are shared between a bilingual's languages, but her work provides an insightful overview of such literature that has suggested different views on shared (or independent) representations of the mental lexicon (particularly in the section on "Bilingual memory representation", p. 34-37). Important to highlight is that in this specific case, the emphasis lies on the lexicon and conceptual representation, and that other linguistic aspects were not directly involved. Perhaps most notable for us, is the Revised Hierarchical Model (RHM) of Kroll and Stewart (1994). The authors discuss two different viewpoints that have been described in related work: either a bilingual's language systems are independent of each other or "common" (i.e. shared). Alternatively, more coordinated work suggests that a hierarchical model is perhaps a more natural approach to the translation process. In such a view, it is argued that *words* are stored in language-specific memory systems but that *concepts* are shared between languages. Links exist between the language systems themselves and between the systems and the shared conceptual memory. The RHM, then, claims that the strength of the connection between the languages themselves and the shared lexicon is determined by the dominance of one language (L1) over another (L2). The link between L1 and the conceptual memory is particularly strong and the lexical link from L2 to L1 is most prominent. That means that translating from L2 to L1 is more straightforward by using lexical transfer, and that translating from L1 to L2 requires so-called "concept mediation", making translation slower from L1 to L2 than from L2 to L1.

In a similar contrasting study, Grosjean (1985) makes the distinction between two views of research in bilingualism. In the monolingual or fractional view a bilingual is considered to have two separate language competences, and the "covert or overt contact between their two languages should be rare" (p. 470). Either one language is used (and activated) or the other, but simultaneous use is accidental or comes down to language borrowing and code switching. In this context, code switching is the bilingual concept of switching languages rather than the technical symbol-transfer process that Seleskovitch (1976) refers to. An important implication in this view is that language activation is seen as a conscious choice by the bilingual and that automatic, unconscious access to both languages in the mind is rare or at least unexplained. On the other side, and further discussed in Grosjean (1989, 1997, 2001), stands the bilingual or holistic view. In this perspective of bilingualism, a bilingual is not only the combination of two equally monolingual parts but rather they contain a blended linguistic system that encompasses both their languages in co-existence. This view is most similar to the horizontal view discussed above. Bilinguals can activate the desired language systems alongside a continuum depending on the situation. On one side of the spectrum is monolingual mode, where they only activate one of their languages. On the other end, a person can mix the languages when in bilingual mode (e.g. code switching and borrowing). The bilingual speaker can position themselves anywhere on this continuum between these two end points depending on the situation.

More recent, experimental studies provide evidence to support the idea that horizontal processes are more highly involved in translation than vertical ones. Experiments by Macizo and Bajo (2004, 2006), for instance, compare reading for comprehension with reading for translation for the Spanish-English language pair and find that reading for translation is significantly slower. They argue that these results support the claim that horizontal processes are active during translation. Such processes integrate code switching and target language forms before ST comprehension has concluded, which means that both language systems are active at the same time during the translation process. In a vertical view, one would expect that reading for translation has no negative impact on reading time: it should be very similar to reading for comprehension because the level of co-activation in such a view is small or non-existent. Instead, the authors find that accessing the target language occurs before the comprehension phase of the source text is finished, thus imposing additional cognitive load on the translator. These findings concerning the simultaneous activation of both lexical and syntactic levels of processing, were confirmed in Ruiz et al. (2008).

Such results about shared representations are in line with the conclusions of priming studies in cognitive psychology. In (monolingual) structural priming, the choice between two equally grammatical and correct syntactic structures is influenced by a previously processed prime structure (Bock, 1986; Levelt & Kelter, 1982). Extending such paradigm to bilingualism and related to the shared representation implication of the horizontal view on translation, Hartsuiker et al. (2004) experimented with Spanish-English bilingual participants and showed that syntactic priming (active or passive constructions) occurs across languages and that syntax of two languages can be shared in the mind. This so-called shared-syntax account implies that language rules that are shared between language systems lead to a facilitating effect and priming of the same rule in the different languages. This theory was confirmed in Hartsuiker et al. (2016). In later research, Hartsuiker and Bernolet (2017) suggest that shared representations of syntax are not static and that syntactic priming works in function of language proficiency. Put differently, language learners will be primed differently than bilinguals, and the way in which syntax is shared in the mind develops in function of the language learner's proficiency. The underlying warning is that researchers should be cautious about making claims involving priming depending on the proficiency level of participants. Jacob et al. (2017) find that in German speakers of L2 English, both the level of embedding (whether a construction is part of a subordinate or main clause) and the constituent order influence the choice of target structure. They conclude that, because both these variables play their own part in causing cross-lingual priming, "the effect can presumably only be caused by a linguistic representation. ... an obvious candidate for this is a hierarchical syntactic tree representation" (p. 279). For an in-depth overview of research on cross-linguistic structural priming see Hartsuiker and Bernolet (2017).

From a production-oriented viewpoint, Gile (1995) introduces the Sequential Model of translation. The model entails that for a meaningful translation unit (be it a word or a larger unit), the translator creates a mental Meaning Hypothesis. If the hypothesis is found to be plausible in terms of the source language and world knowledge, this meaning can be reformulated into the target language. During this reformulation phase, the unit of translation is translated using TL knowledge as well as any required extralinguistic information. The translation of this particular unit is then verified to make sure that the information of the source side is transferred correctly. In addition, the translation is tested for contextual, stylistic, and pragmatic appropriateness. During the translation process, the translator occasionally verifies that larger groups of translation units are correctly translated, too. This model is "not designed to be an accurate description of the actual translation process ... it represents an idealized process in which pedagogically important components are stressed" (p. 106) and Gile does not go into shared representations or co-activation: the Sequential Model solely serves as a pedagogical tool that illustrates the individual, sequential phases that contribute to the production of a translation. It is for instance possible that during this sequential process of translation the languages or specific language properties are both active (perhaps to different degrees). This model does, however, highlight the step-by-step, sequential view of understanding the source text and reformulating it into the target language. Such an approach is common in (pedagogy involving) interpreting (Gile, 1995; Lederer, 1994, 2003) where the deverbalization process is a marked step in the translation process (e.g. Lederer, 2003, Section 1.5). “Ideas” (as Seleskovitch calls it) should be transferred after abstracting the meaning from the form of ST and then they can be reformulated in the target language.

Rather than exclusively contrasting vertical and horizontal processes Schaeffer and Carl (2013) propose a Recursive Model of translation that implies that translation consists of both. In recursive cycles horizontal and vertical processes are integrated in the translation process. Early, horizontal processes access shared representations which, as Seleskovitch (1976) warned, lead to a target text that is highly influenced by the source text. By default the source text is, almost automatically, very closely followed, leading to a literal translation as suggested in the literal translation hypothesis of Tirkkonen-Condit (2005). Vertical monitor processes verify the acceptability of the translation and ensure equivalence between the source and the target text, and interrupt the automatic translation processes if a problem is encountered. The authors provide supporting evidence of such interconnected view of horizontal and vertical processes from a priming study where they tested whether reading and translating a sentence primes shared representation more than reading for comprehension only. Because participants could recall previously seen sentences better in the translation condition, it is suggested that during translation the shared representations are activated alongside monolingual representations. If a monolingual and bilingual representation overlaps then such aspects are activated twice, allowing for easier recall. Later experiments by Schaeffer and Carl (2017) confirm this view. That study also reaffirms the aforementioned shared-syntax account (Hartsuiker et al., 2004) by observing an effect of word order differences among translators on the eye-key span (EKS; Dragsted, 2010; Dragsted & Hansen, 2008) and on the likelihood of concurrent ST reading and TT typing. EKS measures the time between the first or last fixation on a word and the first keystroke that contributes to the prediction of that word (Schaeffer and Carl (2017) calculated EKS from the first fixation). The authors conclude that their results confirm the shared-syntax account because “when the word order in the ST and the TT is dissimilar, also the eye-key span (EKS) is shorter and fewer different word orders are observed” (p. 147). The Recursive Model of translation of Schaeffer and Carl (2013) is inspired by an interpreting model by Christoffels and de Groot (2005) in which the source and target lexicon is activated simultaneously to allow for parallel comprehension and monitoring of the produced translation. It is also heavily influenced by the Monitor Model by Tirkkonen-Condit (2005), who revived the “monitor”-moniker from earlier work by Toury (1995, p. 191-192), who in turn used it to describe a statement by Ivir (1981) who famously wrote that a translator deviates from literal, formal correspondence “only when the identical-meaning formal correspondent is either not available or not able to ensure equivalence” (p. 58).

1.2.3 Translation Difficulty

This thesis focuses specifically on syntactic difficulties in translation, but issues in translation can manifest itself on different linguistic levels. In this section I will summarise relevant literature in the field of (human) translatability. Most content of this overview is also spread out over the following chapters, where each chapter highlights specific aspects of its topic with respect to translation difficulty. This overview section should therefore be seen as complementary to the following chapters rather than an independent whole. Translatability in this section does not refer to the, almost philosophical, discussion whether or not a source text can truly and fully be translated (see Sec. 1.2.1 in this thesis, as well as Catford, 1965, Ch. 14 “The Limits of Translatability”). Instead, the focus lies on difficulties that can hinder a translator. For a deep-dive into (source) text complexity and translation difficulty, also see the overviews in Akbari and Segers (2017); Heilmann (2020); Sun (2012, 2015).

Although not directly using the term translation “difficulties”, Ervin and Bower (1952) discuss translation distortions where the meaning of the source text has been changed in the translation due to a number of language-related categories. As a first category, direct lexical translations may not share the exact conceptual meaning with the original text, leading to an incorrect translation. Second, grammatical rules and requirements may differ between the source and target language system, which may cause either a loss of information or, conversely, uncertainty or vagueness (e.g. languages where the gender of the speaker is part of the grammar compared to those where such information is not represented). Syntactic variations may also result in unintended emphasis or even other, unwanted meaning. Cultural factors, finally, can have an important effect on which translation should be produced, and depend on the languages and cultures involved.

Nord (2005, p. 167; first published in 1991) makes the distinction between translation difficulties and translation problems, the former of which, she argues, is subject to the specific translator. Translation problems, however, can be categorised as follows. Pragmatic problems are caused by a difference in the source and target situation in which the text and its translation are used. Convention-related problems are cultural-bound, similar to the cultural factors of Ervin and Bower (1952). Linguistic problems relate to structural differences between source and target language *systems*. Text-specific problems, finally, arise from specific properties of (parts of) the *source text*.

In addition to such broad, categorical approaches translation difficulty has also been investigated with empirical methods. A pioneer of sorts, Campbell (1999) defines translation difficulty in terms of the cognitive processing effort involved with the task – a methodological approach that is assumed in this thesis as well. In addition to text-intrinsic difficulties, the author mentions translator competence and the mode of translation as contributing factors to translation difficulty. Those will not be discussed in this thesis (or barely, cf. the distinction between students and professional translators in Ch. 2) and the focus lies on properties of a text and its relation to a potential translation. Campbell specifically refers to two cognitive approaches to translation difficulty. First, limitations of a translator's working memory with respect to the task are indicative of difficulty (Gathercole & Baddeley, 1993). In terms of (source text) syntax, the author gives the example of grammatically difficult items where a lexical item, perhaps ambiguous, needs to be held in the limited capacity of the working memory until another disambiguating element is encountered, for instance its grammatical head. Because the working memory has a limited capacity, it can only hold (and process) a limited amount of information at a time. Supplementary to that approach, a lexis-driven language processing paradigm can be utilised as seen in follow-up research to the speech production model of Levelt (1989), particularly de Bot and Schreuder (1993). In such a view, difficulties in the source text are either those lexical items for which no lemmatised form is available in the mental lexicon or whose lemma is underspecified so that the (semantic) concept cannot be readily retrieved. In an initial experimental study, Campbell hypothesises that translation alternatives across translators of the same text can serve as an indicator of difficulty and motivates that decision by its correlation with the number of edits that translators made to a segment. Number of edits are indicative of dealing with problems or difficulties. Such a focus on translation variation left its mark and was later refined by means of an entropy-based component (both on the lexicosemantic and syntactic plane; Carl & Schaeffer, 2014; Carl et al., 2019), as will be discussed later in this thesis. Among Campbell's results is the impact on translation difficulty of word class (particularly verbs and adjectives), complex (and ambiguous) noun phrases, and the level of abstractness. Similarly, considering difficult lexical items, Dragsted (2005) emphasises the effect that difficult terminology has on the translation procedure.

Campbell continued research into translatability, most notably with his colleague Hale. Initially in Campbell and Hale (1999) and later in Campbell (2000), they present the Choice Network Analysis (CNA), which is a continuation of Campbell's earlier work involving variation amongst the generated translators. The underlying assumption here is that "target texts are a tangible source of evidence of mental processing in translation" (Campbell, 2000, p. 32) and that a larger sample of possible translations approximates the total number of possible translations and, as such, must reflect the decision-making process of the subject-translators. However, contrary to the hypothesis in Campbell (1999), Hale and Campbell (2002) found that it is not necessarily the case that higher variation (multiple translation choices) lead to higher difficulty, which in this study is taken to be reflected by translation accuracy.

Campbell (and Hale) above touch on two different aspects that can contribute to translation difficulty. On the one hand, source text specific properties can be responsible for a number of problems and on the other, translation-specific issues may give rise to higher translation difficulty. Translation is a complex process that involves interacting mental sub-processes (cf. Sec. 1.2.2) but, broadly speaking, we can say that the source text needs to be read, understood, and its meaning translated. Text reading and comprehension is therefore an important component of the translation process. The relation with the field of readability should be clear. It has been suggested that formulas to quantify the readability level of the source text can to some extent be used as approximates for translation (Jensen, 2009, and see Sec. 2.2.2 in this thesis for more on readability). These formulas typically use textual information such as number of (difficult words), sentence length, number of syllables, and so on. Such a claim that translation difficulty relates to translation difficulty was later confirmed in experimental research by Sun and Shreve (2014), who found that a source text's readability explains its translation difficulty, but – and this should be emphasised – only partially. Translation difficulty in this research was measured by (a variation of) the NASA task load index (TLX) (Hart & Staveland, 1988), a subjective rating scale for assessing workload where participants fill out self-assessment surveys. Similarly, Liu et al. (2019) used TLX scales to assess subjects' translation difficulties, and they used readability formulas alongside frequency metrics and non-literalness (e.g. idioms) to quantify text complexity. This way of measuring text complexity (readability, frequency, and non-literalness) is in line with previous work by Hvelplund (2011). Other studies have also used readability scores to control for text complexity in experimental translation studies (e.g. Daems, 2016; Sharmin et al., 2008).

Liu et al. (2019) found that a moderate correlation exists between the self-assessment of TLX data and behavioural data. In translation process research such data is often used as an approximation of cognitive processes, and hence difficulty. By observing translators' behaviour through their process data, a window is provided to look into the difficulties they face. This measure is bound to reflect cognitive problems more accurately than only using translation accuracy as a measure of difficulty, as was frequently done before eye-tracking technology was available (e.g. Hale & Campbell, 2002). The first to use eye-tracking in studies related to the translation process was O'Brien (2007) who made use of this technology to identify cognitive effort during the use of translation memories. Since then, many researchers have followed this experimental approach (Alves & Vale, 2009; Balling et al., 2014; Daems et al., 2017; Dragsted & Hansen, 2009; Jakobsen, 2011, and many more). In fact, most references to translation process research in this thesis make use of eye-tracking data as an approximation of cognitive effort. Another type of process data that can provide insights into the mental processes of a translator, is keystroke logging. In relation to cognitive effort, they are frequently used to measure pauses during the production process of a translation. If a translator pauses their typing activity for a given amount of time, these pauses can be indicative of problem-solving mental processes (e.g. Carl et al., 2008; Daems et al., 2015; Immonen & Mäkisalo, 2010; Lacruz et al., 2012; O'Brien, 2006). There is some debate about how long a meaningful pause is supposed to be, however (Lacruz, 2017).

Some examples of studies that use behavioural data to investigate translation difficulty follow. Daems et al. (2017) found that from-scratch translation is more effortful than post-editing a machine translated text, as measured by the number of eye fixations on source text tokens. Immonen and Mäkisalo (2010) used pauses to show that the processing of main and subordinate clauses differs depending on the task (monolingual text production or translation). During translation the length of the pauses (and hence effort) preceding a subordinate or main clause are almost identical. The conclusion that they make is that during translation main and subclauses are processed independently, even though the latter is grammatically embedded in the former. In his thesis, Heilmann (2020) finds that it can be generally shown in empirical data that a syntactically more complex source text requires more cognitive processing effort but with the important side note that "[t]he human cognitive system seems to be equipped to deal with some types of complexity better than with other" (p. 254). Here as well, syntax refers to source text operationalisations such as number of main/embedded clauses or non-clausal chunks, and word group length. Keystroke and eye-tracking metrics served as a proxy for translation effort and difficulty. As said before, in addition to source text difficulties such as the ones discussed above, translation-specific properties that involve both the source and target text likely play a crucial role in translation difficulty as well. Sun (2015) notes that, indeed, many translation difficulties come down to problems of equivalence between a source text and (a) possible translation(s) (cf. Sec. 1.2.1). The Choice Network Analysis of Campbell (2000) was discussed before and is a prime example of how the process of selecting a translation from a variety of options can be modelled. As alluded to before, some work has continued in that direction. Dragsted (2012), for instance, confirmed that target text variation across subjects relates well with process data indicative of cognitive effort. Carl and Schaeffer (2014) adapt the concept of CNA by applying entropy, a measure of uncertainty based on the distribution of probabilities, to lexical choices (HTra). Their goal was to model the literality of a translation. The concept of literality and ways to operationalise it, has been referred to before in Section 1.2.1 and will be discussed in more detail in the following chapters, especially Section 5.2.1. Carl and Schaeffer find that word translation entropy correlates with gaze duration and that more uncertainty in lexical choice thus has an effect on cognitive effort. These results have been replicated a number of times, and supplemented by similar findings about the positive effect of word translation entropy on the translation process (Carl & Schaeffer, 2017; Schaeffer & Carl, 2017; Schaeffer, Dragsted, et al., 2016).

More interesting for this thesis on syntax, are those efforts that quantify syntactic, translation-specific characteristics. Cross is a word-based metric to quantify for a source word how its translation has been reordered relative to the translation of the previous word Schaeffer and Carl (2014). Cross can either be negative (regression), positive (progression), or zero (no movement). Word order divergences, as measured by this metric, have been shown to have a positive effect on eye-tracking data indicating that word reordering leads to increased cognitive effort (Schaeffer, Dragsted, et al., 2016). The measure has also been adapted to consider variations of multiple translators, similar to word translation entropy above. Entropy of word reordering (HCross) correlates with word translation entropy, which is perhaps unsurprising: different lexical realisations may require different positions in the sentence and different word orders may need different lexical items. In addition, HCross affects eye-key span (Schaeffer & Carl, 2017), a cognitive measure that combines eye-tracking information with keylogging data. The idea is that the time between the first visual contact with a word and the first keystroke that contributes to its translation quantifies the processing time of that word (Dragsted, 2010; Dragsted & Hansen, 2008). The results of Schaeffer and Carl suggest that higher entropy in word order leads to more difficulties. Bangalore et al. (2015) define syntactic entropy (HSyn) based on three manually annotated properties of the target text: clause type (independent, dependent), voice (active, passive), and valency of the verb (transitive, intransitive, ditransitive, impersonal). Similar to the other entropy metrics above, they found that more variance has a positive effect on both keystroke and eye-tracking measures. A last entropy-based metric is joint source-target alignment / translation distortion entropy (HSTC; Carl, in press, further discussed in Sec. 5.3.3). It incorporates both semantic and syntactic information (similar to word translation entropy and HCross, above) and also involves word groups rather than only single words. In line with the other entropy metrics, a positive effect on the translation process was observed, specifically on production duration. More variance in translation choices thus corresponds to a slower translation process.

As stated before, this thesis contributes to the field of translation difficulty, particularly by providing new ways of measuring the syntactic relationship between a source text and a given translation. Quite some work involving syntactic complexity of the source text exists, but here I focus on translation-specific difficulties. Some work has been done in this respect, too, particularly Cross, HCross, HSyn, and HSTC as mentioned above. HCross, HSyn, HSTC are entropy-based and require multiple translations, which may not always be available. Carl (in press) finds that around ten translations are needed to approximate word translation entropy values of a real statistical population. In addition, HSyn requires manual annotations. Of these metrics only HSTC can take word group information into consideration, but the metric also explicitly includes semantic information, so it cannot be used as a purely syntactic measure. The suggested metrics therefore fill in some gaps in how syntactic translation difficulties can be quantified (Sec. 1.4). They should give researchers more control over and insights into specific syntactic differences. They are intended to complement the previously discussed metrics by providing a fine-grained means to measure syntactic divergences. They differ in some respects from the other metrics that were just described. First, none of my suggested metrics require multiple translations (although one can easily apply entropy to them[2]). Second, they are intended to cover more syntactic ground than previous measures did, including word and word group based information as well as structural linguistic features. An alternative approach to word reordering is included, accompanied by ways to quantify word group reordering, shifts of linguistic word groups, changes in linguistic label and finally modifications in (aligned) deep linguistic structure. Finally, manual annotations are not required – but if researchers have access to highly accurate annotations those can be used as well.

1.3 Data

In the following chapters we will make use of two existing data sets. For experiments involving translation process data we used the ROBOT data set that was created during the PhD project of Daems (2016). It contains post-edited and from-scratch translations by ten student translators and twelve professional translators. Out of eight texts, each participant post-edited four texts and translated four others so that in total every text had around ten translations and around ten post-edited versions. These texts were all chosen to be comparable in terms of complexity and readability. In addition to the final translations, the translation process of the translators was also recorded, giving us access to valuable keystroke and eye-tracking information. The translation process was recorded using an EyeLink 1000 eye tracker and the software packages Inputlog (Leijten & Van Waes, 2013) and CASMACAT (Alabau et al., 2013). The final translations were manually word aligned with YAWAT (Germann, 2008) and finally processed with the Translation Process Research DataBase (TPR-DB; Carl et al., 2016)[3], which creates useful overview tables of the translation product and process measures on both the word and segment level. In our studies we only made use of the from-scratch data because the focus is on written translation without access to a prior translation (such as in post-editing or revision). The processing steps of this data set differ between studies, though. For instance, in Chapter 4 the outlier exclusion of data points was specifically chosen to be identical to a previous study by Bangalore et al. (2015), and in Chapter 5 residual outliers were excluded. In Chapters 2, 3 and 4 we made use of sentence measures whereas word-based metrics were investigated in Chapter 5.

It should be noted that we did consider creating and using our own trans-

[2] Rather than categorical entropy Shannon (1948), which the aforementioned metrics are based on, I would suggest entropy implementations that handle continuous data better, such as differential entropy, particularly the formulation by Jaynes (1963, 1968).

[3] `https://critt.as.kent.edu/cgi-bin/yawat/tpd.cgi` (You can access public studies with username: TPRDB, password: tprdb)

lation process data. We contemplated using data that was generously given to us by language educators (with students' consent) and that contained corrections of students' assignments and exams, but at that time our attention had shifted to a need for process data as a proxy for cognitive effort in addition to the translations themselves. Our following efforts to gather process data was greatly hindered by the COVID-19 situation because we could not set up eye-tracking experiments, which would be the prime source of translation process information. Instead, we asked ten participants to translate the multiLing dataset[4] from home while using the key-logging software Translog-II (Carl, 2012). They were not allowed to make use of external resources such as dictionaries or Wikipedia. The participants were native Dutch speakers who had recently obtained a master's degree in translation involving English into Dutch translation. In addition to these human translations, machine translations of these texts were collected with Google Translate and DeepL, but of course no process data is available in the latter case. The tokenisation of the data was manually corrected[5] and the source and target tokens were manually aligned with YAWAT (Germann, 2008). The final data was processed with the TPR-DB. Still, in the research that followed we decided to use the ROBOT dataset because it contained those interesting eye-tracking measures that are frequently used as a representation of cognitive effort. Even though we did not use the created data set, it might prove useful in future research. The data is also publicly available as a public study in the TPR-DB, so fellow researchers are free to make use of it. ENDU20 contains the human translations and ENDU20-MT contains the translations by DeepL (P20) and Google Translate (P21).

For the machine learning systems that were built in Chapter 3, a large parallel corpus was required. Because the goal was to predict word (group) order differences between a source sentence and its translation, no process data was needed. We could therefore make use of the Dutch Parallel Corpus (DPC; Macken et al., 2011), which contains more than 140,000 parallel English-Dutch sentence pairs. This data set was automatically tokenised and lower-cased by using the preprocessing scripts provided by Moses (Koehn et al., 2007).[6] Word alignment was also done automatically by means of GIZA++ (Och & Ney, 2003).

1.4 Metrics and Implementation

Over the course of the PhD project, a number of equivalence metrics have been suggested. The focus was mainly on syntax as that aligned most with my

[4] https://sites.google.com/site/centretranslationinnovation/tpr-db/public-studies#h.p_iVVuCQOHJx2O

[5] This is an intricate procedure of manually changing individual XML files. Code was written to facilitate the process. This code is available on GitHub: https://github.com/BramVanroy/predict-translate-annotation.

[6] https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer

interests and background. Initially, these measures were limited to aggregated values on the *sentence* level (Vanroy et al., in press; Vanroy, Tezcan, & Macken, 2019). Those metrics were implemented in such a way that one output value was given per source sentence. However, when analysing translation process data, it is desirable to have access to metrics on the *word* level because it provides a more fine-grained perspective of the translation process. With that in mind, those metrics were re-implemented and extended in Vanroy et al. (2021) so that for each word a meaningful value can be extracted. This section provides a detailed overview of all the equivalence measures that were created during the PhD project. A lot of information provided here will return in some form or other in the following chapters, but because different articles highlight different aspects, it seems useful to have a complete description in one place. The implication is that this section is quite dense. The reader may prefer to use this section as a reference later on, and first read the following chapters which build up to the final metrics discussed here.

The full implementation of this library, called `ASTrED` after one of the metrics, is available on GitHub.[7] I am a staunch proponent of open source software (OSS), and I hope that by making this library publicly available fellow researchers can make use of it to push the field further. The library is easy to use and is specifically designed to make it accessible for researchers with a variety of backgrounds (including non-technical ones). By providing tokenised source and target sentences and their word alignments, all equivalence measures can be calculated. Functionality is present to automatically create the word alignments for the given source and target sentence, and in addition the tokenisation process can be automated, too. The implication is that one only needs a parallel, sentence-aligned corpus and the library does the rest: tokenisation, parsing, and word alignment are all taken care of. To be able to do this, the library integrates the tokenisation and parsing capabilities of the Stanford pipeline `stanza` (Qi et al., 2020)[8] as well as word alignment capabilities of the recently introduced library Awesome Align (Dou & Neubig, 2021)[9]. Both libraries claim to be state-of-the-art in their respective field. Although such a highly automated approach may be useful, and even required, for large corpus studies, other researchers may choose to provide manually tokenised data and verified word alignments instead because their quality will undoubtedly be higher. Taking it a step further, it is also possible to add linguistic information such as dependency labels and part-of-speech tags (POS) manually. In sum, the automated capabilities of the library can be (dis)used to the user's preference.

In what follows the available metrics will be discussed but first a short detour is required. Some metrics rely on a linguistic representation of the source and target sentence. In our case, these representations need to be

[7] `https://github.com/BramVanroy/astred`

[8] See `https://stanfordnlp.github.io/stanza/available_models.html` for all 66 available languages

[9] `https://github.com/neulab/awesome-align`

comparable across languages. To this end, we will use and discuss Universal Dependencies (UD). During my PhD project I have only investigated English-to-Dutch translation (because I have a strong formal linguistics background in those languages), but because of the dependency on UD, any language pair that can be represented with this annotation scheme can be used in the tool.

1.4.1 Universal Dependencies

Particularly, we are interested in a dependency structure where each word has a dependency relationship to one other word. That means that each word has a *to*-relationship with another word (its head). For example, in Figure 1.1, the word *baker* is a subject (`nsubj`) to *tastes*. Some terminology that will be used throughout this thesis: *baker* is the child (or *direct* descendant) of *tastes*, and *tastes* is the parent of *baker* (or *direct* ancestor or head). Every sentence must have exactly one `root`.

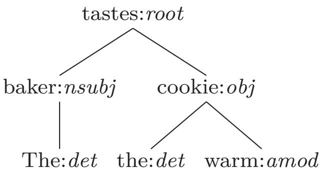


Figure 1.1. Example of a dependency tree of the sentence "The baker tastes the cookie"

For our purposes, these representations need to be comparable between languages, that is, they must use the same label set in the two different languages. That is not straightforward because historically, different annotation schemes have been created for different (families of) languages or different use cases (e.g. de Marneffe et al., 2006; Kromann, 2003; Nivre & Megyesi, 2007). For this reason, we make use of the Universal Dependencies annotation scheme[10] (Nivre et al., 2016). It was created to allow cross-linguistic analyses by constructing one linguistic annotation scheme for as many languages as possible and is heavily inspired by previous efforts to this common goal, such as the Google Universal Dependency Treebank project (UDT; McDonald et al., 2013) and Stanford Dependencies (SD; de Marneffe et al., 2014, 2006; de Marneffe & Manning, 2008).[11]

One of our word-based metrics (Sec. 1.4.2) makes use of the universal part-of-speech (POS) tagset[12] which is also part of the Universal Dependency an-

[10] All UD labels and their descriptions can be found on the website `https://universaldependencies.org/u/dep/index.html`. To minimize the effect of slight differences between languages, we only consider main dependency labels and not their subtypes

[11] For the history of UD, see the historical overview on the website `https://universaldependencies.org/introduction.html#history`, particularly the history sections in McDonald et al. (2013); Nivre et al. (2016).

[12] `https://universaldependencies.org/u/pos/`

notation scheme and which builds on previous work by Petrov et al. (2012). Part-of-speech tags indicate the word category that a word belongs to given its meaning in the sentence (e.g. adjective `ADJ`, noun `NOUN`, auxiliary verb `AUX`). Whereas dependency labels indicate the relationship of a word to its parent, the POS tag of a word does not depend on any other words but the word's own meaning in its current context. This also means that a sentence can be represented as a tree by using the dependency label of each word (Figure 1.1), which is not possible with POS tags because there is no relationship between the POS tag of one word and another. The example sentence visualised as a dependency tree in Figure 1.1 "The baker eats the warm cookie" is enriched with POS tags in Example 1.[13]

(1)	The	baker	tastes	the	warm	cookie
	det	nsubj	root	det	amod	obj
	DET	NOUN	VERB	DET	ADJ	NOUN

1.4.2 Changes in Dependency Label and POS Tag

A simple way to see whether the syntax of a word has changed, is by comparing its dependency label to the dependency label of its translation(s). The intuition here is that we expect that a change in dependency label, and thus a change in the relationship between words, requires more processing effort from the translator. Note that this change compares the "flat" labels. As described above, dependency labels are assigned to a word to indicate its relationship to the word's head. When we simply compare the labels of the word, we do not take into account the head. The metric is thus superficial and does not take the specific word relations of words to their head into account but merely the label. On the word level, the number of label changes L of a word w can be formulated as Equation 1.1.

$$L = \# \{t \in T : t \neq l\} \tag{1.1}$$

where:

T the collection of labels of all words aligned to w
l the label of word w

A similar method of checking the linguistic equivalence between the source and target text is comparing the universal POS tags of a source word with the POS tag of its translation. A change in part-of-speech tag indicates that a word has been translated by a word of a different word group.

[13] From a formal perspective, the difference between the POS tag *DET* and the dependency label *det* for "The (baker)" is that the POS tag is simply a word category, not in relation to any other word, whereas its dependency label specifically expresses the relationship between the word and its parent "baker"

Although these metrics are not complex nor innovative (see the use of word class changes in Serbina et al., 2017), they measure interesting changes between a source text and its translation. A change in dependency label highlights a functional shift where a source word has changed its grammatical relation or function in the dependency tree, e.g. passivisation where a subject `nsubj` *she* is translated as a case marker `case` *by* accompanied by an oblique nominal `obl` *her*. A different POS tag on the other hand indicates a change in word class, often caused by a different choice of word surface form or by a radically different translation. It is therefore likely that a change in POS tag goes hand in hand with a change in dependency label because a different word class will often require a change in dependency structure.

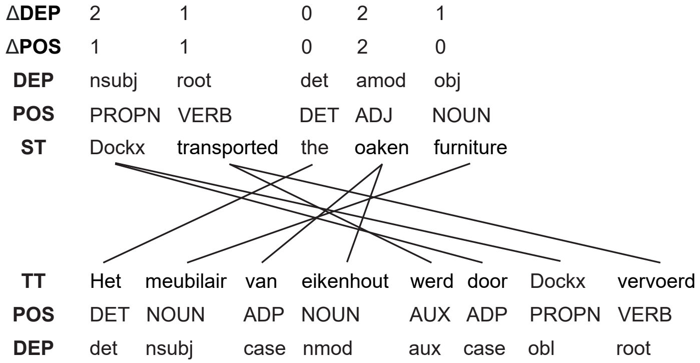

ΔDEP 2 1 0 2 1
ΔPOS 1 1 0 2 0
DEP nsubj root det amod obj
POS PROPN VERB DET ADJ NOUN
ST Dockx transported the oaken furniture
TT Het meubilair van eikenhout werd door Dockx vervoerd
POS DET NOUN ADP NOUN AUX ADP PROPN VERB
DEP det nsubj case nmod aux case obl root

Figure 1.2. Changes in dependency label and POS tag for an English-to-Dutch translation. The Δ rows indicate the number of differences between the source word's label and the label of its translation(s)

Figure 1.2 illustrates how these two metrics, dependency label change and POS tag change, differ. This translation contains two noteworthy phenomena. First, the active sentence is translated as a passive. That means that the dependency label subject `nsubj` turns into an object (technically a case marker `case` and oblique nominal `obl`). In terms of POS tag, however, it only differs from the added case marker, which is a preposition `ADP` and not a proper noun `PROPN` like the company name "Dockx". Additionally, the source-side object `obj` turns into the subject `nsubj` without a change to its POS. The second phenomenon involves modifiers: the translator has chosen to translate the adjectival modifier "oaken" as a marked nominal modifier `case nmod` "van eikenhout" *of oak-wood*, which is a valid translation alternative to "eiken meubilair" *oaken furniture*. Hence both the POS tag and the dependency labels are different from those of the source word in the two translated words.

Around the same time that we created and implemented these two measures, Nikolaev et al. (2020) suggested similar metrics. Their approach is somewhat more complex as they measure dependency *path* differences in the dependency tree rather than merely dependency labels. It is therefore more abstract than the suggestion above, and similar to (but less intricate than) our ASTrED measure described below.

1.4.3 Cross and `word_cross`

Schaeffer and Carl (2014), reiterated in Carl et al. (2016), introduced a local, word-based metric to calculate the similarity between a source and target sentence in terms of their word order. These values can be calculated from the source to the target side (CrossS) and vice-versa (CrossT). In this thesis, the focus always lies on the source side. The metric is generated for each word incrementally. That means that the Cross value of a word is relative to the position of the translation of the previous word. Figure 1.3 shows an example of CrossS for the sentence “Killer nurse receives four life sentences”, translated as “El enfermero asesino recibe cuatro cadenas perpetuas”. “Killer” is translated by “asesino” which is the third word in the sentence ($CrossS = 3$). For “nurse” the translator has to jump backwards by two words to translate the first word in the one-to-two alignment “El enfermero” ($CrossS = -2$). “receives” is translated as “recibe” which is three positions further than the previous translation (“El”) ($CrossS = 3$), and so on. Cross is not implemented in our library but serves here to show that other word reordering metrics exist.

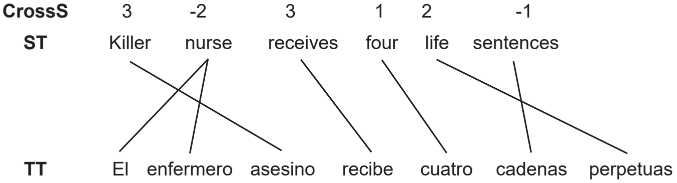


Figure 1.3. `CrossS` values for an English-to-Spanish translation with word reordering. Taken from Carl et al. (2016), originally shown in Schaeffer and Carl (2014)

In Vanroy, Tezcan, and Macken (2019), we suggested an alternative approach to the Cross value. Rather than having an asymmetrical metric where the values differ on the source and the target side, we opted for a bidirectional/symmetrical one. This means that the number of reordering steps to change the source sentence into the target sentence is the same as the other way around, or, more formally, that the number of crosses in the source and the target sentence are equal. On the word level, the Cross value (Schaeffer & Carl, 2014) of a word is determined by the position of its translation relative to the position of the previous word’s translation. In our proposal, the `word_cross` values take the movement of *all* other words into account. In other words, whereas a word’s Cross value is determined by the reordering of its translation relative to the previous word’s translation, its `word_cross` value is impacted by the reordering of all words in the sentence, including its own.[14] If a word is aligned with multiple target words, that word's `word_cross` is the sum of all the word crosses attached to it. Whether two alignments a_1 and a_2 cross each other can be formulated as Equation 1.2 and should be interpreted as: two alignments cross each other if the positions of the involved source words relative to each other are reversed on the target side.

$$cross((k,l),(i,j)) = \begin{cases} 1, & \text{if } i < k \ \ \& \ \ j > l \\ & \text{or } k < i \ \ \& \ \ l > j \\ 0, & \text{otherwise} \end{cases} \tag{1.2}$$

where:

k	source index of the first alignment link
l	target index of the first alignment link
i	source index of the second alignment link
j	target index of the second alignment link

In Figure 1.4, both the `CrossS` and `word_cross` values are given for each word. Circles indicate a `word_cross`. The example should make clear that whereas `CrossS` models regression (negative values) and progression (positive values), `word_cross` does not. The metric is intended to measure the amount of word reordering that a word's translation has to undergo with respect to the other words and their translation in the sentence and not only relative to its preceding word. For example, the word *nurse* has a `CrossS` of -2 because its first aligned word is two positions back from the translation of the previous word *asesino*. It is translated as two words, *El* and *enfermero*. Both of these words are reordered and have crossed the translation of another source word *Killer*. Therefore, the `word_cross` value of *nurse* is 2.

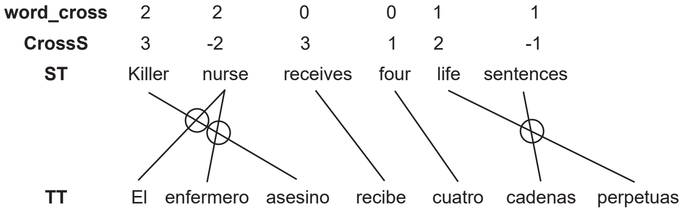


Figure 1.4. `CrossS` and `word_cross` values for an English-to-Spanish translation with word reordering. Circles indicate crossing alignment links that correspond to `word_cross`. Adapted from Carl et al. (2016), originally shown in Schaeffer and Carl (2014)

[14] When we use the term Cross we refer to the metric introduced in Schaeffer and Carl (2014). On the other hand, `word_cross` is in reference to Vanroy, Tezcan, and Macken (2019)

For both metrics, smaller (absolute) values indicate a translation where the word order of the source text can be largely maintained. A translation that has the exact same word order as its source text and where each word has a one-to-one translation will have zero word crosses. That means that the `word_cross` of each word is 0 and their `CrossS` value 1. Such a scenario is visualised in Figure 1.5.

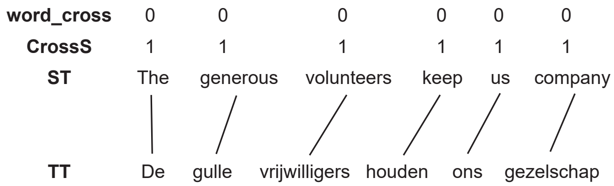

word_cross 0 0 0 0 0 0
CrossS 1 1 1 1 1 1
ST The generous volunteers keep us company
TT De gulle vrijwilligers houden ons gezelschap

Figure 1.5. `CrossS` and `word_cross` values for an English-to-Dutch translation without word reordering

1.4.4 Sequence Cross

Whereas previous research most often focuses on the properties of a word in solitude, we also provide a means to take larger units into consideration. `seq_cross` is calculated in the same way as `word_cross` (Equation 1.2) but the unit involved is different.

The fundamental property of the word groups that will be discussed in this and following sections is consecutiveness. This property can be defined as Definition 1.

Definition 1 (Consecutiveness)**.**

- All words on the source side need to be aligned with at least one word on the target side and vice-versa
- Words cannot be aligned to any word that does not belong to the aligned group (so-called "external alignments")
- All words on both the source and target side must be consecutive, that means that all the words between the first and the last word of the group must be included in that respective group

Note that the term "groups" is used interchangeably for a monolingual group itself and the alignment between such source and target groups. In other words, when a a set of aligned source and target words meet the requirements set by the definition, a source and target word group is created as well as an "alignment" between these two groups. Creating these word groups, then, is a way of increasing the unit size of the alignments, leading to fewer or equal alignments between the source and target text.

More specifically than Definition 1, a sequence group adds an additional requirement that states that word alignments within a group cannot cross each other. Words that do not form a valid group with other words form their own, singleton, sequence group.

Definition 2 (Sequence group)**.**

- Includes requirements for Consecutiveness (Definition 1)
- Word alignments of this group cannot cross each other (as per Equation 1.2)

As said before, the consequence of forming groups is that instead of having all the *word* alignments between all source and target words, the source and target *sequence* groups are connected with one alignment. This can lead to greatly reduced cross values compared to `word_cross` because the metric then measures crossing groups rather than individually moving words. The `seq_cross` value of a word is the cross value of its sequence group. Similarly, the SACr value of a word is the cross value of the SACr group that that word is part of but this will be discussed in Section 1.4.5.

Example 2, taken from our dataset, contains an English source sentence, a Dutch translation, and their word alignments (Ex. 2c). These word alignments are written in an often-used `i-j` format where `i` is an index of a source word and `j` the word that it is aligned with (also called the Pharaoh format; Koehn, 2004). If a word's index is not included in the alignments, it means that that word has no translation equivalent (also called a null alignment).

(2) a.

The	show	is	billed	as	the	museum	's	largest	ever	.
0	1	2	3	4	5	6	7	8	9	10

b.

Dit	is	de	duurste	voorstelling	ooit	in	het	museum	.
This	*is*	*the*	*most-expensive*	*show*	*ever*	*in*	*the*	*museum*	*.*
0	1	2	3	4	5	6	7	8	9

c. 0-0 0-2 1-4 2-1 3-1 4-1 5-7 6-8 7-6 8-3 9-5 10-9

In Figure 1.6, only the first part of the sentence in Example 2 is shown. "is billed as" is the only sequence of words that meets all the criteria (Definition 2) to form a sequence group (rectangle). Therefore, instead of having each word aligned individually (dashed grey lines), it is aligned as a single group to "is". All the other words are singletons, meaning that they constitute a group by themselves without any change to their alignments.

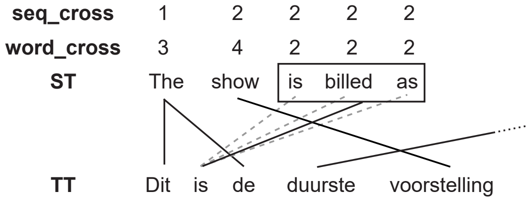


Figure 1.6. Partial visualisation of Example 2 illustrating sequence cross compared to `word_cross`. Solid box indicates sequence group of more than one word

If all words and their movement are considered individually, then the cross value is quite high. However, considering the word group as a candidate for the translation unit, leads to lower reordering values because we then consider that a translator has translated (and/or moved) a whole word group at once. In this thesis we do not make any statements about the size of the translation unit (cf. Chapter 5) but we accept that it is possible – and perhaps likely – that a translator also considers larger word groups and not only focuses on individual words. For instance, "show" has a relatively high `word_cross` value of 4 because it crosses all the words in "is billed as" as well as "duurste" which is aligned with "largest". In our sequence approach, however, we suggest that "is billed as" moves as a single sequence of words with no internal reordering. As such it could be seen as a single word *group* movement operation by a translator, which leads to a reduced cross value (`seq_cross`) of 2 for "show" as it only crosses the aforementioned group and "duurste". Note that the example also highlights the corner-stone of our implementation of how cross is calculated: the `seq_cross` values of the words in the larger group do not necessarily change compared with their `word_cross`, but forming such groups does impact the `seq_cross` values of the context. In other words, the movement of words and word groups impacts cross values of surrounding items as much as it impacts its own cross value.

1.4.5 Syntactically Aware Cross (SACr)

Sequence groups create larger units of alignment that inevitably lead to smaller cross values than `word_cross`. These groups, however, are seemingly arbitrary in terms of the syntactic properties of its words. We therefore suggested a linguistic correction of these groups (Vanroy et al., in press). To make sure that word groups constitute a linguistically meaningful whole, we add an additional requirement to Definition 2 as follows (Definition 3):

Definition 3 (SACr group)**.**

- Includes requirements for Sequence groups (Definition 2)

- Both the source group and target group must constitute a valid subtree in their respective dependency tree

This means that, from a monolingual perspective, the words in a group must form a meaningful part of the dependency tree of that sentence. In practice, that means that the parent of a word must be present in the same group with the exception of the topmost word in this subtree.

SACr can be illustrated with the same example as before (Example 2). It is clear that “is billed as” fulfils all the requirements of Definition 2 but it does not comply with the added requirement in Definition 3. Looking at the dependency tree of the source sentence in Figure 1.7, it is clear that the words of this group (underlined) do not constitute a valid subtree per the requirements described above. Particularly, the parent of “as” is not present in the word group. If “largest” had been part of the group, then the group would have been valid but it does not belong to the group because it violates the other requirements. Particularly, it does not directly follow the other words in the group as it is separated from them by the phrase “the museum’s”.

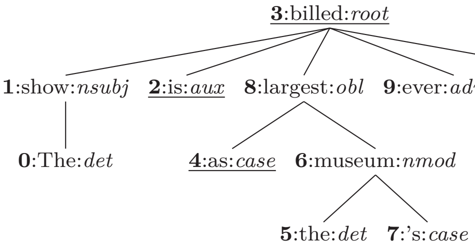


Figure 1.7. Dependency tree of the sentence “The show is billed as the museum’s largest ever .”

The consequence is that the initial sequence group “is billed as” needs to be corrected and split up into smaller groups that are valid. Because “is billed” does fulfil all requirements in Definition 3 to constitute a SACr group, the initial group is split up into “is billed” and “as”, as shown in Figure 1.8. This in turn leads to an increase of SACr cross values compared to sequence cross values for involved words because sequence groups that do not form a valid subtree are split up into smaller SACr groups. And more groups, and thus more alignments, lead to more crossing alignments.

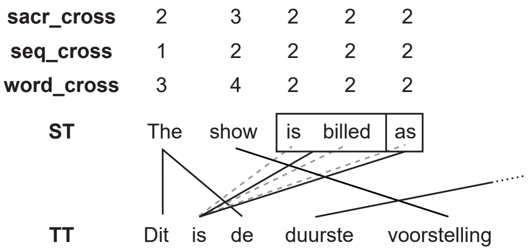


Figure 1.8. Partial visualisation of Example 2 illustrating SACr cross compared to sequence and `word_cross`. Solid boxes indicate SACr groups created out of the sequence group "is billed as"

Whereas sequence groups provide a way to model word group movement, ultimately lowering cross values compared to `word_cross` due to its larger units (thus less alignments), SACr can split those sequence groups up again to ensure that they are linguistically motivated. As such, sequence cross is always less than or equal to both SACr and `word_cross` due to the size of the units that are taken into account.

1.4.6 Multi-word Groups (MWG)

The observant reader may have noticed that our requirements for creating sequence groups has one important shortcoming when applied to real-word translations. It will occur that a specific construction of multiple words is difficult to transfer to the target language in terms of alignment, i.e. where a one-to-one mapping of meaning of source to target words is tedious. Even if such one-to-one or one-to-few alignments are possible, the translator may simply opt for a less straightforward construction, either because of personal preference or to create a target text that is more natural. Ultimately, such translation choices will lead to m-to-n alignments where all m source words are aligned to all n words in the group because all words contribute to the meaning that is being transferred and a smaller compositional alignment is not possible.

We consider all m-to-n alignments to be MWG candidates in cases where both m and n are greater than 1 (if $m = 1$ or $n = 1$, the group is a valid sequence group anyhow as there can not be an internal cross). That means that when calculating the sequence cross and SACr cross metrics for words, we can consider whether MWGs are allowed or not and base our calculations on the groups that are formed on said condition. If MWGs are considered valid groups, all MWG candidates are interpreted as valid sequence and SACr groups, even if they do not meet the criteria of the aforementioned Definitions 2 and 3. Instead, an alternative Definition 4 defines what we consider to be MWGs. If MWGs are not considered in the calculation of word group crossings, the words in m-to-n alignments do not constitute valid groups according to our criteria (Definitions 2 and 3 above) and each alignment will be considered individually.

Definition 4 (Multi-word group)**.**

- Includes requirements for Consecutiveness (Definition 1)
- All words in the source group need to be aligned with all the words in the target group and vice-versa

Note that our inclusion of MWGs for word groups only has an effect on the sequence and SACr cross values and not on `word_cross`. When considering groups of words, we calculate cross on the sequence level or on the SACr level, and in both cases we can (dis)allow the creation of MWGs as a alternative group type.

It is no surprise that not considering MWG as valid group alignments can lead to incredibly high sequence and SACr cross values, similar to the `word_cross` value for the same construction, because in such an event the unit of alignment does not increase in size. Instead, rather than forming a group, all alignments would constitute their own singleton group because they do not meet the requirements to form a sequence or SACr group. When we do allow MWGs, however, such alignment constructions are considered as valid coherent units, which leads to a considerably lower cross value. This is illustrated in the following example.

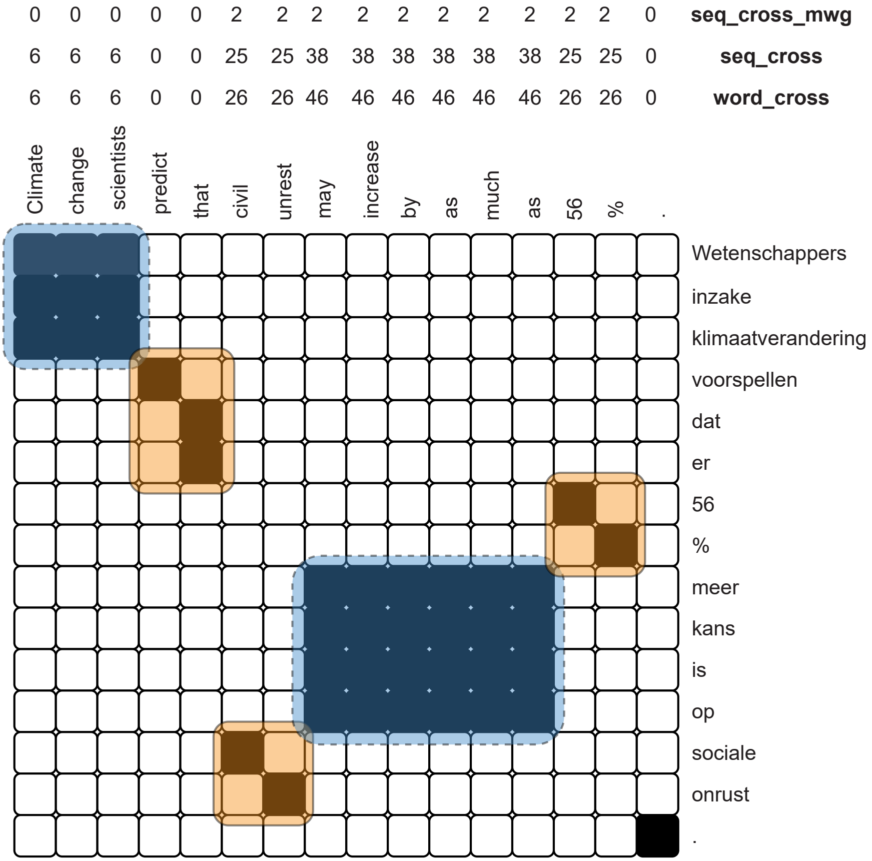


Figure 1.9. Alignment table of Example 3. Blue squares (dashed lines) indicate MWG candidates, orange squares (solid line) are valid sequence groups

In Figure 1.9, an English source sentence is translated as a Dutch target sentence. For clarity's sake, the example sentence and its translation and alignments are also given in text in Example 3. The example contains two MWG candidates, "Climate change scientists" aligned with "Wetenschappers inzake klimaatverandering" and "may increase by as much as" aligned with "meer kans is op".

(3) a. Climate change scientists predict that civil unrest may increase
0 1 2 3 4 5 6 7 8
by as much as 56 % .
9 10 11 12 13 14 15

b. Wetenschappers inzake klimaatverandering voorspellen dat
Scientists of climate-change predict that
0 1 2 3 4

er	56	%	meer	kans	is	op	sociale	onrust	.
there	*56*	*%*	*more*	*chance*	*is*	*of*	*civil*	*unrest*	*.*
5	6	7	8	9	10	11	12	13	14

c. 0-0 0-1 0-2 1-0 1-1 1-2 2-0 2-1 2-2 3-3 4-4 4-5 5-12 6-13 7-8 7-9 7-10 7-11 8-8 8-9 8-10 8-11 9-8 9-9 9-10 9-11 10-8 10-9 10-10 10-11 11-8 11-9 11-10 11-11 12-8 12-9 12-10 12-11 13-6 14-7 15-14

In the figure, black squares indicate word alignments, blue squares (dashed lines) are MWG candidates, whereas orange groups (solid lines) are valid sequence groups (no internal or external cross; consecutive source and target words).[15] At the top, the cross values for each word are given. `word_cross` serves as a baseline. `seq_cross*` values show the cross value of the group that a word belongs to, as before in Section 1.4.4. The distinction between the two versions is that for `seq_cross_MWG` we allow MWG candidates (blue squares) to be valid groups but in the other variant, they are not. That means that rather than having large units that possibly cross each other, the words themselves constitute their own group. Because every word in a MWG crosses every other word, that leads to large values. The number of internal crosses in an m-to-n aligned group scale with the number of words on the source (m) and target side (n) according to Formula 1.3. The proof for this formula is given in Appendix A.

$$cross_{MWG} = \frac{1}{4} \cdot mn(m-1)(n-1) \tag{1.3}$$

where:

m number of words on the source side of the MWG
n number of words on the target side of the MWG

In the kind of word alignment visualisation in Figure 1.9, a literal, one-on-one translation would be a straight diagonal descending line from left to right. Disruptions in word order are those parts where an alignment deviates from that diagonal. Looking at the individual word alignments in the figure, that is the case for instance for "civil" (positioned relatively early in the source sentence) which is aligned with "sociale" (near the end of the target sentence). The corresponding alignment point does not directly follow the previous word "that" diagonally. Looking at the figure, all the alignments of the words of "may increase by as much as 56 %" needs to be crossed to align "unrest" with "onrust". That leads to a significant `word_cross` value of 26.

In terms of word groups, "civil unrest" (aligned with "sociale unrust") and "56 %" have swapped places between the source and target sentence. If we create sequence groups as per Definition 2, then only the orange groups

[15] For simplicity's sake we do not include SACr but the results would be identical to the `seq_cross*` values because the sequence groups "civil unrest" (aligned with "sociale onrust") and "56 %" ("56 %") are valid subtrees so the SACr groups are identical to the sequence groups, leading to the same respective cross values

(solid lines) are acceptable sequence groups: if we do not allow MWG, then the blue groups (dashed lines) are not valid groups and the individual word alignments constitute each their own individual sequence (and SACr) group. That means that on the sequence group level, even though "civil unrest" is a single group, it still has to cross with every single alignment (black box) in "unrest may increase by as much as" (aligned with "meer kans is op") and the newly formed sequence group "56 %" ("56 %"). That decreases the sequence cross value of "unrest" by one compared to the `word_cross` value (because it only crosses with the group "56 %" rather than the individual alignments), but it still is quite high at 25. If, however, MWGs are allowed and groups can be formed according to Definition 4, then the blue groups (dashed lines) are valid groups, too. If that is the case, the sequence group "civil unrest" only needs to cross two other alignments, namely the one between the MWG "may increase by as much as" and its translation, and the alignment between the sequence group "56 %" and its translation. This leads to a `seq_cross_MWG` value of only 2.

As is clear by the `seq_cross_mwg` values in Figure 1.9, interpreting MWGs as single units greatly reduces the word group cross values. Also here the principle holds that the larger the units of alignment, the lower the cross value can be. So the decision whether or not to allow MWGs in either sequence or SACr groups not only impacts the cross values in the word group at hand, but also the cross values of the surrounding groups. In the example discussed above the `seq_cross_MWG` was reduced greatly for all words inside the MWG "may increase by as much as" but as a consequence of this word group also the `seq_cross_MWG` value for words in "civil unrest" decreased significantly.

1.4.7 Aligned Syntactic Tree Edit Distance (ASTrED)

Words in a sentence establish a hierarchical structure where each word is a dependent to its head (except for the root node). It would therefore be an interesting endeavour to compare the source tree with the tree representation of the translation. To compare tree structures, we can make use of tree edit distance (TED), a metric that calculates the minimal edit operations that a source tree needs to be transformed into the target tree. Particularly, we make use of a Python implementation[16] of the APTED algorithm (Pawlik & Augsten, 2015, 2016). TED will recursively investigate each node and its position in the tree with respect to the target tree to see whether this node can be matched in the target tree (and no operations are needed) or whether something needs to change. Three operations are possible, namely deleting, inserting and substituting a node in the tree (the latter also referred to as "renaming"). Every operation has a cost attached to it, in our case all edit operations have a cost of 1 whereas matching has no cost (0). The goal of the algorithm, then, is to find a number of sequential operations with the lowest

[16] `https://github.com/JoaoFelipe/apted`

possible total cost to transform the source into the target tree.

A straightforward approach would be to calculate tree edit distance between the source and target dependency trees as-is. However, such a method would be rather naive as it does not consider alignment information. This is important because we are interested in structural differences between the source and target sentences while also taking the alignment between those two into account. In other words, when comparing a source tree to a target tree, we only want to match those nodes with each other that are translations of each other and find operations that need to occur to fill in the rest of target tree. We call this method Aligned Syntactic Tree Edit Distance (ASTrED). It was first introduced for sentence-level quantification of structural differences between a source sentence and its translation in Vanroy et al. (in press) (particularly Section 4.3.4) but has since been adapted to provide meaningful information for each individual word, as will be discussed below.

Terminology might be confusing here. Throughout the text, "TED" will be used as an indicator for tree edit distance. That is, the metric itself; the process to find differences between tree structures. Different implementations (algorithms) of TED have been proposed, mostly with a goal to make the metric as fast as possible or to consume less memory, but the results of all implementations should be the same. APTED is one of those implementations. ASTrED makes use of tree edit distance, particularly the APTED implementation, to calculate syntactic differences, but instead of just calculating TED on the source and target syntactic trees, those trees are first changed to incorporate word alignment information. So ASTrED is a name to indicate the preprocessing of the source and target trees to include word alignment information, followed by calculating TED on those modified trees.

Example 4, where an adverbial clause (`advcl`) has been translated as a nominal oblique (`obl`) and shifted to the end of the sentence, can illustrate the difference between regular TED matching and ASTrED matching.

(4) a. Scared and scarred by the global crisis , families hoard their
0 1 2 3 4 5 6 7 8 9 10
money
11

b. Gezinnen sparen intensief uit angst voor de wereldwijde crisis
Families save intensively out fear of the world-wide crisis
0 1 2 3 4 5 6 7 8

c. 0-3 0-4 1-3 1-4 2-3 2-4 3-5 4-6 5-7 6-8 8-0 9-1 9-2 10-1 10-2 11-1 11-2

Note that the example contains two MWGs ("Scared and scarred" – "uit angst", and "hoard their money" – "sparen intensief"). The source and target dependency trees of Example 4 are given in Figure 1.10 and 1.11 respectively. It is visually clear that the source tree is extended on the left side whereas the target tree extends on the right.

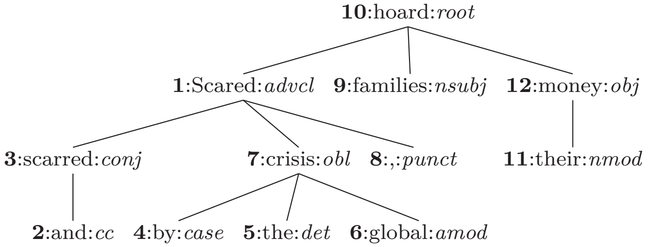


Figure 1.10. Dependency tree of the source sentence "Scared and scarred by the global crisis , families hoard their money"

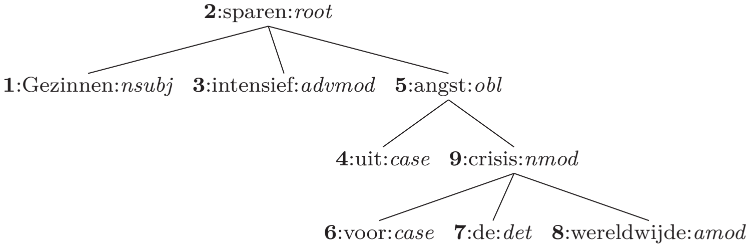


Figure 1.11. Dependency tree of the target sentence "Gezinnen sparen intensief uit angst voor de wereldwijde crisis"

In regular tree edit distance, 10 edit operations (Example 5) are required to transform the source tree into the target tree. These operations do not take into account word alignments and are a means to simply compare the structure of two (unrelated) sentences. Below, the full node representations are given for illustrative purposes, but in reality this example solely compares the dependency labels between trees (not the words or indices). When a node is deleted, its children are re-attached to the deleted node's parent. A node can be inserted anywhere, also between a parent and (a subset of) its children. On the word level, we keep track for every word what the edit operation is that is necessary to give this node a place in the target tree. As such, from a directional perspective (e.g. source to target or vice-versa), none of the starting words can be "inserted" as these are always the starting point and already present. We can only speak of "insertions" about words on the other side, i.e. the tree that we want to end up with.

(5)
1. Delete (src): **12**:money:*obj*
2. Delete (src): **11**:their:*nmod*
3. Delete (src): **9**:families:*nsubj*
4. Delete (src): **1**:Scared:*advcl*
5. Delete (src): **8**:,:*punct*
6. Delete (src): **3**:scarred:*conj*

7. Insert (tgt): **3**:intensief:*advmod*
8. Rename: **2**:and:*cc* – **1**:Gezinnen:*nsubj*
9. Insert (tgt): **9**:crisis:*nmod*
10. Insert (tgt): **6**:voor:*case*

These operations can be described more technically as follows, but keep in mind that for clarity's sake the words are used when in reality the tree only consists of the word's dependency label. The subtree of "their money" is deleted, as is the word "families". The latter needs to be deleted and re-inserted because on the target side it is on the left rather than the right of the deeper subtree of "angst". "Scared" is deleted, which means that the subtrees of "scarred", "crisis", and "," are now directly attached to the root. The punctuation "," is deleted as is "scarred", which means that "and" is now a direct child of the root. This node must then to be renamed because "and" is a coordinating conjunction `cc` and it needs to be renamed to `nsubj` (to turn into "Gezinnen"). A `nmod` (src: "crisis") needs to inserted between the `obl` (tgt: "crisis") and most of its children except for `case`, which becomes a sibling of the `nmod` rather than its child.[17] Finally, the missing `case` "voor" needs to be inserted.

Up to now, we have looked at the edit operations that are necessary. However, an important part of TED is finding identical nodes between the source and target tree, also called "matches". A match is preferred over any other edit operation as the cost for a match is 0. The matched nodes in the example are the following, which is evident from their identical dependency label.

(6)
- **4**:by:*case* – **4**:uit:*case*
- **5**:the:*det* – **7**:de:*det*
- **6**:global:*amod* – **8**:wereldwijde:*amod*
- **7**:crisis:*obl* – **5**:angst:*obl*
- **10**:hoard:*root* – **2**:sparen:*root*

Two of these are noteworthy. First, "by" is matched with "uit" but for our purposes of comparing a source structure with the syntactic structure of its translation, it does not seem to be a very sensible decision. "uit" is not the translation of "by" nor is it it in the same position with respect to its initial siblings `det amod` but rather one level higher. Second, "crisis" is matched with "angst" because they are both oblique nominals `obl`. They are not, however, the match that we may want because those two words "crisis" (src) and "angst" (tgt) have nothing to do with each other in terms of word

[17] Note how both "uit" and "voor" are `case`. Therefore, one may expect that `nmod` (src: "crisis") would be inserted between `obl` and *all* children, followed by the insertion of "uit" because that would mean that the sequence "by the global" – "voor de wereldwijde" could be transferred as a whole. However, because the TED algorithm procedurally finds the fewest number of operations needed, its decision ensures that the subtree "uit angst crisis" is correct before completing the lowest subtree "voor de wereldwijde"

alignment but because of the algorithm's naive look at the two syntactic trees, this is the most efficient match that it can find. With ASTrED we suggest an alternative approach that *can* take word alignment information into account with more sensible matches as a consequence.

The main idea behind ASTrED is keeping the tree structure of both the source and target sentence as "the skeleton" but changing the node labels to contain information about the word(s) that this node is (in)directly aligned with. So rather than dependency labels, the underlying TED algorithm – which is identical to regular TED used above – needs to compare structures whose labels consist of an aligned representation of the respective word (also called its "connected representation"). For a given group, this representation is created as a mapping from source dependency label to target labels for all alignments involved in the group,[18] separated by a colon 0-:. If a source word has multiple alignments, then the target tokens are separated with a comma, e.g. s:t_1,t_2. If the group consists of multiple alignments, then those source-to-target mappings are separated by a pipe |. In both the source and the target tree this "serialised" representation will then replace the initial node labels that belong to this group. An example follows below. By doing so, all items in the group will have the same label in both the source and target tree. As such, matching must be done on words that belong to the same group and cannot be incidental.

To find all word's connected representations, we group all words together that are directly or indirectly "connected" to each other. Note however, that the connected groups that we use in ASTrED are different from the criteria for sequence or SACr groups, or MWGs. Particularly, groups in ASTrED consist of *all* directly or indirectly connected items. In other words, sequence, SACr, and multi-word groups are valid ASTrED groups but ASTrED groups are not necessarily valid sequence groups as they do not have to comply with any of the given definitions. That means that in the theoretical example in Figure 1.12, "A B D" aligned with "E G" is a group, even though that is not a valid sequence, SACr or multi-word group according to our definitions. All words are connected to each other in a way, e.g. "A" is connected to "G" via "E" and then "B". The connected representation of this group is A:E|B:E,G|D:G.

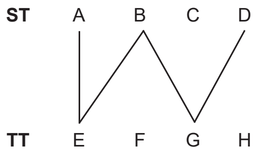


Figure 1.12. Minimal example of ASTrED groups

[18] In practice, the index of a word in its sentence is also included in this representation to avoid that groups that consists of the same dependency labels match each other. For illustrative and conciseness reasons, we only show the examples with dependency labels.

Back to Example 4, we will further focus on two of these groups and not discuss all of them. Figure 1.13 can help in visualising the connected groups. A simple, one-to-one group is “crisis” `obl` on the source side aligned with “crisis” `nmod` on the target side. These two words form one group because they are exclusively aligned to each other. Another group is “Scared and scarred” aligned with “uit angst”. This is an MWG but as explained above, such connected groups are valid ASTrED groups.

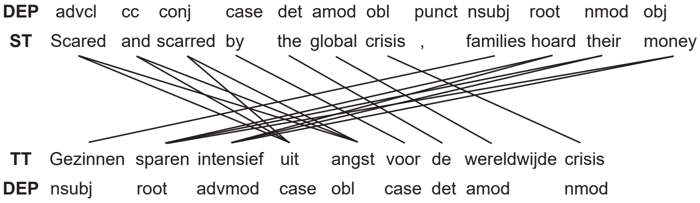

DEP advcl cc conj case det amod obl punct nsubj root nmod obj
ST Scared and scarred by the global crisis , families hoard their money
TT Gezinnen sparen intensief uit angst voor de wereldwijde crisis
DEP nsubj root advmod case obl case det amod nmod

Figure 1.13. Alignment visualisation of Example 4

Using the template that was discussed above, these groups have the following connected representations:

- `obl:nmod` (crisis:crisis)
- `advcl:case,obl|cc:case,obl|conj:case,obl` (Scared:uit,angst|and:uit,angst|scarred:uit,angst)

Now the labels in the source and target trees of words that belong to these groups are replaced with the connected representation. In Figure 1.14 and 1.15, the labels for the words in these groups have been replaced by the full group representation. The words are included below the labels for clarity only and are not actually used when calculating TED. Dots (. . .) are given for all nodes that are not part of the two groups that we discuss to save space.

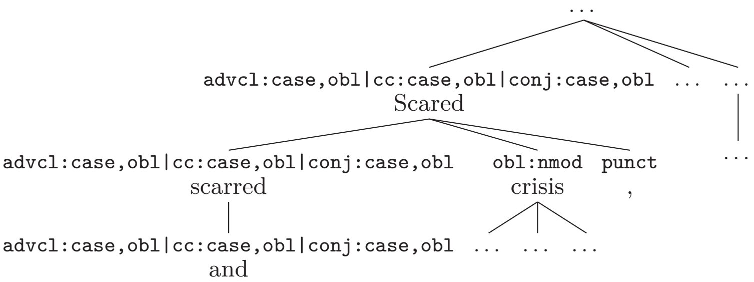

...
advcl:case,obl|cc:case,obl|conj:case,obl
Scared
...
...
advcl:case,obl|cc:case,obl|conj:case,obl
scarred
obl:nmod
crisis
punct
,
...
advcl:case,obl|cc:case,obl|conj:case,obl
and
...
...
...

Figure 1.14. Modified dependency tree of the source sentence “Scared and scarred by the global crisis , families hoard their money”

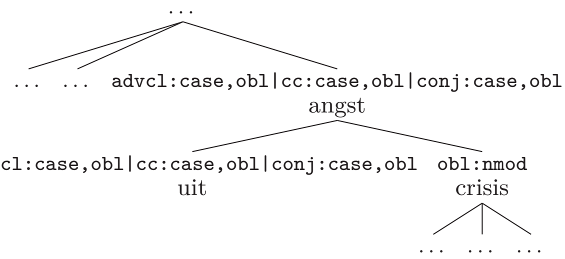

...
... ... advcl:case,obl|cc:case,obl|conj:case,obl
angst
advcl:case,obl|cc:case,obl|conj:case,obl
uit
obl:nmod
crisis
... ...

Figure 1.15. Modified dependency tree of the target sentence "Gezinnen sparen intensief uit angst voor de wereldwijde crisis"

It is clear now that the labels in the source and target tree are identical for words belonging to the same group. If we calculate tree edit distance on these modified trees, the algorithm will prefer matching a source word with an (indirectly) aligned target word because it has the same label and matches are preferred over other operations because a match has no cost attached to it. That means that the matching nodes now take alignment information into account, i.e. a source node can only match with a target node if it is (in)directly aligned with that word in the sentence. For the example at hand, that leads to the following matches in 7. (Exclamation marks indicate new or changed matches compared to regular TED discussed above in 6.)

(7)
- ! **1**:Scared:*advcl* – **5**:angst:*obl*
- ! **2**:and:*cc* – **4**:uit:*case*
- ! **4**:by:*case* – **6**:voor:*case*
- **5**:the:*det* – **7**:de:*det*
- **6**:global:*amod* – **8**:wereldwijde:*amod*
- ! **7**:crisis:*obl* – **9**:crisis:*nmod*
- **10**:hoard:*root* – **2**:sparen:*root*

The edit operations that are needed to "fill in the gaps" left by the matching nodes are not given for brevity's sake, but 7 are needed for ASTrED compared to 10 in regular TED.

The issues that were brought up earlier for regular TED have been averted. First, "crisis" (src) now successfully matches with "crisis" (tgt) rather than with "angst". Second, the preposition "by" matches with "voor" rather than with "uit", which makes more sense because of its position in the tree and because it is aligned with "by". In addition, new matches can be made such as "Scared" and "angst". This leads to a full subtree match of "Scared by the global crisis" and "angst voor de wereldwijde crisis".

On the word level, this approach provides us with two values for each word, namely the operation that is required on it (e.g. `match` for "crisis") and the cost attached to that operation (0 for matching, 1 for any other).

Consequences that may not be immediately clear from the example above is that ASTrED is also the only one of our metrics that can handle null alignments and differences in number of words in an *m*-to-*n* translation, because such occurrences will inevitably lead to insertions or deletions. For example, the unaligned comma **,** in the source tree is still present in the modified source tree in Figure 1.14 and will need to be deleted. However, ASTrED does not successfully catch word order differences because the word order position of a child relative to its parent is unknown, i.e. the tree structure does not indicate the position in the sentence of words by itself. The only guarantee in our implementation is that siblings are ordered in the same way as they occur in the sentence.

Chapter **2**

Correlating Process and Product Data to Get an Insight Into Translation Difficulty

Bram Vanroy* · **Orphée De Clercq*** · **Lieve Macken***
*LT^3, Language and Translation Technology Team, Ghent University

Abstract

Research in the field of translation studies suggests that translation product features can indicate translation difficulty. In the current pilot study, we investigate three of these features, namely the number of errors made in a translation, word translation entropy, and degree of syntactic equivalence. We correlate these translation product features with translation process features that can be put together into three categories: duration, revision, and gaze information. These features serve as a proxy for the cognitive effort required to solve difficulties in translation. The data that we used was gathered from professional translators as well as students of translation studies. The product data contains manual error annotations of the translations, automatically calculated entropy values, and syntactic re-ordering metrics. The process data are derived from keystroke and eye-tracking data gathered during the translation process. By correlating product and process data, we inspect how translation difficulty is reflected in the translation process and whether it is feasible to use product features to predict difficulties in translation. In addition, we also compare data of professional translators and students. We will show that correlations between process and product features exist, which opens many doors to further research on translatability.

Keywords: translation studies · translation difficulty · user activity data · eye tracking · keystroke logging

Publication: Vanroy, B., De Clercq, O., & Macken, L. (2019). Correlating process and product data to get an insight into translation difficulty. *Perspectives*, *27*(6), 924–941.

2.1 Introduction

Measuring and predicting translation difficulty is of use in a number of domains where it is required to have an objective indication of how difficult a given source text in language x is to translate to target language y. For example, (i) in (machine) translation research that depends on texts of a similar or contrasting translation difficulty; (ii) in the translation industry so that source texts can be dispatched to translators or machine translation (MT) systems of appropriate expertise; (iii) in an educational environment where grading students' translating competences requires texts of a relevant difficulty level.

The PreDicT project (Predicting Difficulty in Translation)[1] aims to build a system that can analyse a source text that has been written in language x and that subsequently can predict how difficult that text would be to translate to language y by returning a translation difficulty score. Furthermore, the PreDicT system should be able to indicate the segments (sentences, phrases, words) in the source text that can give rise to difficulties.

This paper reports on a pilot study that investigates three indicators of translation difficulty that have been suggested in related research, specifically the number of errors in the product (Daems et al., 2013), word translation entropy (Campbell, 2000), and the amount of syntactic (non-)equivalence between the source and target text (Sun, 2015). We correlate the difficulty indicators with translation process data. The process data that we used can be sorted into three categories, namely duration, revision, and gaze, which are all derived from keystroke and eye-tracking data.

Our goal is to answer the question whether correlations can indeed be found between translation process data, as proxy for cognitive effort, and product data. If so, product-like features can be confidently used to predict translation difficulty in future research. With the exception of the number of errors that were made in a translation, we believe we can model word translation entropy and syntactic equivalence off-line, i.e. before the translation process has taken place, by using large parallel corpora.

The current paper is structured as follows. First, we will give an overview of relevant research in readability and translation research in Section 2.2. Then we will discuss the methodology and the data that we used in our research (Section 2.3), followed by the results (Section 2.4) and their discussion (Section 2.5). We summarise and end the paper in the conclusion (Section 2.6) where we also hint towards future research.

2.2 Related Research

Readability prediction, which aims to predict the reading difficulty of a text, has been subject to extensive research by the development of readability formu-

[1] `https://research.flw.ugent.be/en/projects/predict`

las (see Benjamin, 2012; Collins-Thompson, 2014; DuBay, 2004; Klare, 1976, for overviews and discussions). Predicting translatability – that is, the difficulty of a translation task – on the other hand has not received the same attention.

It has been suggested that readability formulas can serve "to assess the relative amount of both production effort and comprehension effort needed during a translation process" (Jensen, 2009, p. 61-62). However, even though there is a clear overlap between readability and translatability, the latter cannot simply be solved by copying the methodology of the former because translation is not limited to monolingual reading. Rather, "coordinating reading and writing efforts seems to be an overarching activity in translation" (Dragsted, 2010, p. 43).

2.2.1 Cognitive Effort

In this paper we use translation process data as a proxy for cognitive effort. Data gathered from keystroke logging and eye tracking are used as indirect evidence of cognitive effort. With a focus on post-editing, Krings (2001) proposed three types of effort: temporal effort, technical effort, and cognitive effort. The first two are rather intuitive to understand. First, the longer it takes to complete a task, the higher the temporal effort. Second, from a practical viewpoint, activities that require a high amount of technical gestures, require more technical effort. For example, typing a text to convey a message involves a certain technical effort. The third type of effort, cognitive effort, is harder to define and measure, though. Activities that require mental effort where the brain has to process information or generate new information can be seen as high in required cognitive effort. These types of effort can overlap and oftentimes they even influence one another. For examples and a comprehensive overview of effort in translation process research, see Lacruz (2017). Because of the high inter-influence of the three types of effort, we use cognitive effort in a broader sense than Krings intends. In fact, we will use it exclusively as the overarching class of effort and will only refer to cognitive effort and cognition. Our reasoning is that technical and temporal effort are ultimately resolved by a mental process, hence we place technical and temporal effort under a broad umbrella-interpretation of cognitive effort.

2.2.2 Readability Research

Readability formulas have been vigorously drafted since the 1920s. According to Klare (1984) (as cited in Sun, 2015), more than 200 formulas have been proposed. Traditionally, these formulas use shallow source text statistics such as average word or sentence length. But over the past decade more complex language features have been proposed as well, for example lexical, syntactic, semantic and discourse text features (for an overview, see Collins-Thompson, 2014). Promising developments in computer science and, by extension, natural language processing, have led to innovative approaches that are inspired by statistical models and machine learning (Collins-Thompson & Callan, 2005; De Clercq et al., 2014; Francois & Miltsakaki, 2012; Hancke et al., 2012; Si & Callan, 2001).

Readability formulas are often used for monolingual reading, which means that the focus lies on a single language. To be able to compare different types of reading, Jakobsen and Jensen (2008) set up an experiment where they investigated the process data of four different reading tasks executed by six translation students (abbreviated as *stud.* below) and six professional translators (*prof.*). These tasks were (i) reading for comprehension; (ii) reading in preparation for translating; (iii) reading while speaking a translation; (iv) reading while typing a written translation. They focused their comparison on reading time and eye-tracking data such as number of fixations, total gaze time duration, and fixation duration. For our research especially the juxtaposition of tasks (i) and (iv) are of importance.

Because a translator has to switch between two separate areas of interest, namely the source text as well as the translated text under construction, it is to be expected that reading for translation takes more time and requires more eye movement than reading for comprehension. In the experiment of Jakobsen and Jensen (2008), reading for translation takes between 15 and 20 times longer than reading for comprehension (prof.: 771 seconds vs. 40s, stud.: 945s vs. 61s). Furthermore, eye-movement information shows that the effort that is required for translating is considerably higher than for reading. In particular, these features are the fixation count (1590 for translating vs. 145 for reading), gaze time (prof.: 288s vs. 29s, stud.: 223s vs. 31s), and fixation duration (218 milliseconds on source text + 259ms on target text vs. 205ms).

2.2.3 Translation Studies

In addition to language features that are typically analysed in readability research, predicting translation difficulty also has to take the source text and source language into account, as well as the process of translating from the source language to the target language. Research has focused on the issues that machine translation systems encounter, as well as the difficulties that human translators are faced with.

2.2.3.1 Machine translation

In the field of machine translation, researchers have focused on ways to improve an MT system's translation as well as analysing properties of the source text where MT systems are having difficulties coming up with a good and correct translation (Bernth & Gdaniec, 2001; Naskar et al., 2011; O'Brien, 2004; Underwood & Jongejan, 2001).

According to Underwood and Jongejan (2001), machine translation systems are prone to negative translatability indicators (NTIs, also referred to as “negative sentence properties”, or “translatability indicators”) that make translating a given text difficult. Some of these pointers are lexical ambiguity (e.g. polysemous words, homographs), structural ambiguity (e.g. caused by prepositional phrases or multiple coordination), complex noun phrases, and very long or very short sentences. Human translators are different from machines and not all the aforementioned negative translatability indicators will cause problems for human translators. Furthermore, translators may have to deal with difficulties that are non-existent in MT. Nonetheless, some overlap between translation difficulties of MT systems on the one hand and human translators on the other is to be expected. O’Brien (2004) applies negative translatability indicators on post-editing effort. She found that when less NTIs are present in a segment, the post-editing effort is reduced. However, the author also found that not all NTIs have the same effect on editing effort.

Bernth and Gdaniec (2001) discuss a number of ways to improve what they call *MTranslatability*, i.e. how well-suited a text is to be translated by an MT system. They provide suggestions for improving an input text in preparation of machine translation, which can greatly improve the output quality of a text. By doing so, they also highlight the problematic constructions that MT systems are faced with. Such source-text difficulties are, among others, ungrammatical constructions, ambiguity, coordination, and ing-words.

Rather than looking at how the source text should be tailored to the MT system, Naskar et al. (2011) improved an MT system by “relying on the advice of end-users on the basis of what they deem[ed] should be prioritized” (p. 529). The researchers used linguistic checkpoints to evaluate a system’s performance. A linguistic checkpoint is a point of importance that is required for an adequate translation. These points of interests are subjective and depend on the MT system as well as the end-users’ priorities. Examples of these checkpoints could be an ing-form, noun-noun compounds, and any part-of-speech tag. The checkpoints are sorted into a taxonomy that as a whole represents the important source-text features for a given task. By analysing an MT system’s performance at these linguistic checkpoints, a subjective performance test can be created where the importance of a correct translation of specific linguistic features is determined by the research set-up.

2.2.3.2 Human translation

From the perspective of human translation, Campbell (1999) took an empirical approach and found source-text specific constructions that were difficult to translate to different target languages, such as multi-word units, complex noun phrases, and abstract content words. Additionally including the type of the translation task and the competence of the translator, he presents a general framework that sheds light on translation difficulty. Particularly, the author concludes that “since common difficulties were encountered across subjects texts could be said to be inherently difficult to translate” (Campbell, 1999, p. 57). This statement implies that source-text features play a considerable role in a text's translation difficulty.

Building on those findings, and alluded to earlier by the same author (Campbell, 1998), Campbell and Hale (1999) aim to chart the choices that translators make during a task in a choice network. The idea of a choice network was later generalised to a framework, namely Choice Network Analysis (CNA) in Campbell (2000). CNA tries to model the mental processes of translation. It assumes that a translator's target text is the evidence and the product of his or her mental processes during translation. *Mental processes*, in this context, are in fact the choices that the translator had to make during the translation process. When multiple translators try their hand at the same source text, the product of these translation processes can be combined in a network that represents all the choices that can be made given an input text.

As an example (taken from Campbell, 2000, p. 36-37), one can imagine a choice network analysis of complex noun phrases in English translated to Spanish. Such an English construction of two nouns (N N) can be structurally re-factored into a Spanish translation that, for instance, consists of a single noun (N), a noun followed by an adjective (N Adj), a new complex noun phrase (N N), or a noun and a prepositional phrase (N PP) which entails many choices in itself (e.g. the choice of the preposition).

According to the author, "CNA [is] useful for estimating the relative difficulty of parts of source texts" (Campbell, 2000, p. 38). It follows that the plurality of choices in itself can be quantified as a difficulty indicator as "the more nodes and branches in the network, the more choices are faced".

The Choice Network model can be used to discuss another way of quantifying the choices that translators can choose from, namely word translation entropy (Carl et al., 2016; Schaeffer, Carl, et al., 2016). Word translation entropy indicates the uncertainty for a translator to choose (a) target word(s) for a source token. It revolves around the idea that a translator has multiple ways to translate a given source token. The more options that are available, the harder it is to make a decision. Word translation entropy is situated on the lexico-semantic level. The core idea of word translation entropy (i.e. the number of different translation options) can be modelled and visualised in a CNA. Figure 2.1 shows how *precipice* in the sentence *Residents have to catch a cable car to the top of a nearby precipice to get a dose of midday vitamin D* has been translated by different translators. Note that *afgrond* (abyss) is written in italics because this translation is semantically incorrect; an abyss does not have a top.

On the syntactic plane, syntactic equivalence can serve to indicate problems with translation. Sun (2015) proposes that difficulties that arise during translating from one language into another can generally be attributed to difficulties with equivalence, a concept from translation theory that has been around since the second half of the twentieth century (Pym, 2014, p. 7). Equivalence (or the lack thereof) can manifest itself on different levels in language. On a microscopic layer in morphology, lexicon, and syntax, up to a more global, macroscopic layer: a semantic, pragmatic and ultimately a cultural level (Baker, 2011). In this paper, we are mainly interested in syntactic equivalence. Problems with equivalence can occur when there is non-equivalence, one-to-several equivalence, and one-to-part equivalence according to Sun (2015). This categorisation is in line with an earlier grouping by Kade (1968, pp. 79-89) who uses the German terms *Eins-zu-Null*, *Viele-zu-Eins*, and *Eins-zu-Teil* respectively. Equivalence issues arise "especially for novice translators" (Sun, 2015, p. 36).

Carl and Schaeffer (2017) used both word translation entropy and syntactic equivalence to model translation literality or the lack thereof. They found "strong correlations of cross-lingual semantic [word translation entropy] and syntactic similarities [syntactic equivalence] and that non-literal translations were more difficult and time consuming [...] to produce than literal ones" (p. 55). From this we can assume that word translation entropy as well as syntactic equivalence give rise to higher cognitive effort. In other words, the more choices (or the more elaborate the choice network) or the more syntactic re-ordering has to take place, the more difficult a translation is to create.

2.2.3.3 Translation process research

In addition to the above similarity coefficients, cognitive effort is also often measured by analysing user-activity data (UAD) gathered during the translation process. Detailed information concerning duration (e.g. time to translate, pause information), revision (e.g. number of character insertions or deletions, number of self-corrections), and gaze information gathered with an eye tracker (e.g. number of fixations, fixation duration, regressions) are often used metrics in this type of research (see for instance Carl et al., 2008, 2010; Daems, 2016; Daems et al., 2017; Jakobsen, 2011; Lacruz et al., 2012; Schaeffer, Carl, et al., 2016).

Revision information (such as number of inserted or deleted characters, number of revisions) can shed a light on the cognitive effort a translator had to muster during translation. According to Leijten and Van Waes (2013, p. 360) "[t]he main rationale behind keystroke logging is that writing fluency and flow reveal traces of the underlying cognitive processes. This explains the analytical focus on pause [...] and revision [...] characteristics".

Initially touched upon by O'Brien (2005) and further developed in O'Brien

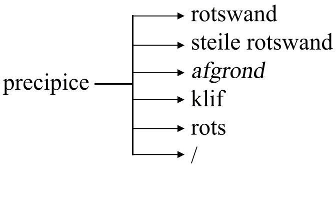


Figure 2.1. A CNA of the translation options to Dutch for English *precipice*

(2006), pauses and in particular pause ratio can be used as an indicator of cognitive effort for post-editing tasks (PE) where translators receive a machine-translated text and correct mistakes or improve the text in other meaningful ways. *Pause ratio* is the total time that a translator has paused (i.e. has not provided keyboard input) relative to the total production time of a translated segment.

The underlying idea is that the longer a translator pauses (i.e. does not provide keyboard input), the more cognitive effort is required to generate a suitable translation. It should be noted, though, that it is currently not possible to ensure that the cognitive effort related to a pause is in fact present and being directed towards the task at hand. In other words, it is nearly impossible to find out the motivation of a pause with certainty (see for instance Kumpulainen, 2015). With regards to our study, however, we assume that pauses that are not related to an increased cognitive effort (e.g. day-dreaming) are scarce and of small to no consequence for our results.

Lacruz et al. (2012, p. 24) reacted to O'Brien's duration metric by altering the concept of pause ratio and instead using *average pause ratio*, which is calculated as the average duration per pause divided by the average production time per word. The authors claim that "[pause ratio] does not take different patterns of pause behavior into account. In particular, it is not sensitive to the existence of clusters of short pauses".

Even though the researchers note that their research was limited to a single translator, they do believe that average pause ratio can be used as a valid measure of cognitive effort, and that it at least is a better indicator than O'Brien's aforementioned pause ratio metric. In her research on the effect of MT quality on PE effort indicators, Daems (2016, p. 131) confirms that, indeed, "average pause ratio is a better measure of cognitive effort than [...] pause ratio".

The methodological work by O'Brien (2005, 2006) and Lacruz et al. (2012), as well as the applied study by Daems (2016) above are restricted to post-editing tasks. A source text is fed into a machine translation system, and the generated output is then post-edited by a translator. Concerning cognitive effort, post-editing and translating are related but not identical tasks. The effort required to create a translation from start to finish is more intensive than post-editing an MT-translated text. Therefore, conclusions drawn in a PE setting are not necessarily applicable to translation. However, due to a lack of comparative research on pause ratio and average pause ratio in human translation tasks, and the tested preference for average pause ratio by Daems (2016), we will use this metric in the remainder of this study.

In Section 2.4, we will examine whether the number of translation errors and selected aforementioned product features reflecting similarity between the source and target text (word translation entropy and syntactic equivalence) correlate with process features that are generally accepted to reflect cognitive effort.

2.3 Methodology

2.3.1 Data Set

The data we use in this pilot study was collected in the ROBOT[2] project (Daems, 2016) and consists of process data in the form of keystroke and eye-tracking data, and product data in the form of the final annotated translations. Eight English source texts of seven to ten sentences each were translated to Dutch by 23 translators who were native speakers of Dutch and had English as (one of) their working language(s). The translators consisted of two well-defined groups: one group of 13 professional translators with minimally five years of translation experience (with one exception, who had been working for two years), and the other of 10 students of a Master in Translation programme at Ghent University. Every translator translated four randomly selected texts. Leaving corrupt or unusable data aside, the ROBOT data set that is used here consists of detailed process and product data of 690 segment translations (314 by students, 376 by professionals).

The ROBOT process data was recorded using CASMACAT (Alabau et al., 2013). This tool can track a user's mouse and keyboard activity in a controlled translation environment and can be extended with an eye tracker to also monitor a user's gaze. The researchers of the ROBOT project used an *EyeLink 1000* eye tracker. CASMACAT's output data is compatible with the CRITT Translation Process Research Database (TPR-DB; Carl et al., 2016). By means of freely available Perl scripts[3] CASMACAT's data could be converted into workable and analysable spreadsheets. These spreadsheets show the aggregated values of a magnitude of features for each translated segment. In this paper we will make use of these spreadsheets and some of the features, as we will discuss in the next section.

2.3.2 Features

As mentioned in section 2.2, literature suggests that process data such as duration, number of character insertions and deletions, and gaze information can mark translation difficulty. In addition, product data such as the number

[2] The ROBOT project compared post-editing (PE) and human translation (HT) by students (stud.) as well as professional translators (prof.). To this end, eye tracking and keystroke logging was used for data collection, but the author also worked with questionnaires to gauge participants' attitudes towards PE and HT. With respect to comparing PE and HT, research topics included (but were not limited to) task speed, task effort, product quality of tasks, and common error types of tasks. In all research questions, the differences or similarities between stud. and prof. is discussed as well. Some of the author's key findings are: PE is faster than HT but their output quality is comparable, PE is cognitively less demanding than HT, stud. behave differently than prof. with regard to processing texts, and the overall translation quality of stud. and prof. is comparable. The project page can be found at `https://research.flw.ugent.be/en/projects/robot`.

[3] See `https://sites.google.com/site/centretranslationinnovation/tpr-db` for guides and tools concerning TPR-DB.

of errors a translator makes, the number of translation choices a translator can choose from (entropy), and the amount of syntactic (non-)equivalence are plausible indicators of translation difficulty. We will calculate correlations between these process (section 2.3.2.1) and product features (section 2.3.2.2).

In the following sections, feature names are set in `monospace`. They are analogous with those used in Translation Process Research Database (Carl et al., 2016) with the exception of `AvgPauseRatio`, `Pausedur` and `EC_TOT`, which were added manually by the researchers of the ROBOT project.

2.3.2.1 Process features

The process data includes, but is not limited to, duration and pause information, textual segment statistics such as length (in tokens or characters), and keystroke and gaze information. In this paper we are only interested in a few that may point to translation difficulty, as found in related research. As Table 2.1 shows, our experiment includes a number of features that can be categorised into three groups, specifically DURATION, REVISION, and GAZE.

For DURATION, we use the features `AvgPauseRatio` (added manually during the ROBOT project and already discussed in Section 2.2.3.3), and the total production time `Pdur` that measures keyboard activity excluding pauses $\geqslant 1s$. These pauses are summed up in `Pausedur` (also added manually during the ROBOT project), which reflects the time that a translator did not use the keyboard. The threshold is motivated by work by Carl and Kay (2011, p. 969), who claim that one second or longer is the optimal duration to separate PUs (production units), which are segments in time where the target text is being produced. Therefore, the sum of all pauses longer than or equal to 1s is the meaningful, production-less keyboard pause.

REVISION categorises all features that have to do with keyboard input. `Mdel` and `Mins` respectively indicate how many characters have been deleted and inserted into the target window. `Nedit` is a broader concept, in the sense that it keeps track of how many times a translator has gone back to a translation and edited the translation. `Scatter`, finally, counts how often two consecutively typed characters do not belong to the same word or consecutive words. Put differently, how frequently a translator moves their cursor to a different word (earlier or later in the text) to make changes to that word.

Lastly, we draw on `FixS` and `FixT` to indicate the number of fixations that a translator has had on the source and target text respectively. These two features constitute GAZE.

Category	**Feature name in data set**	**Description**
Duration	`AvgPauseRatio` (`APR`)	the average duration per pause divided by the average production time per word
	`Pdur`	duration of coherent keyboard activity excluding keystroke pauses $\geqslant 1s$
	`Pausedur`	sum of all pauses $\geqslant 1s$
Revision	`Mdel`	number of characters deleted from the translation window
	`Mins`	number of characters inserted into the translation window
	`Nedit`	number of times the segment has been edited
	`Scatter`	amount of non-linear text production (i.e. when two consecutively typed characters do not belong to the same word or consecutive words)
Gaze	`FixS`	fixations on the source text
	`FixT`	fixations on the target text

Table 2.1. Process features

2.3.2.2 Product features

The produced translations in the data set were manually annotated according to an extensive error typology (Daems et al., 2013).[4] In the current study, we are only interested in the total number of errors (EC_TOT), though. In addition, we use two product features, word translation entropy and syntactic equivalence, that were created by the TPR-DB scripts during the ROBOT project prior to the current study. An overview of these three product features can be found in Table 2.2.

Feature	Feat. name in data set	Description
Error count	EC_TOT	total number of errors made in a segment
Entropy	HTra	word translation entropy
Syntactic equivalence	CrossS	Cross value for source tokens

Table 2.2. Product features

Translation difficulty can take place on different structural planes of language, ranging from phonology (e.g. homophones) and morphology (e.g. irregular verb inflexion) up to the textual level (e.g. coindexing ambiguity). In this study, we include select product features from the lexical as well as the syntactic level. These features are word translation entropy (HTra in TPR-DB) and syntactic equivalence between source and target text (CrossS) as they were touched upon in section 2.2 .

In the context of TPR-DB, word translation entropy is calculated as shown in Equation 2.1 (Carl et al., 2016, p. 31). Entropy is concerned with the impact of new information on the current knowledge.

$$H(s) = \sum_{i=1}^{n} p(s \rightarrow t_i) * I(p(s \rightarrow t_i)) \tag{2.1}$$

In this equation, $p(s \rightarrow t_i)$ stands for the word translation probabilities of a source token s and all its possible translations $t_i...n$. They are computed as how often a source token has been translated to the specified target token (Eq. 2.2).

$$p(s \rightarrow t_i) = \frac{count(s \rightarrow t_i)}{\#\text{translations}} \tag{2.2}$$

[4] This typology is divided into two main categories, namely adequacy errors and acceptability errors. Adequacy entails issues such as contradiction, word sense disambiguation, hyponymy and hyperonymy, deletion, addition and so on. Acceptability itself is divided into five sub-classes namely Grammar & Syntax, Lexicon, Spelling & Typos, Style & Register, and Coherence. These classes each contain even more fine-grained errors.

The information I that is present in a distribution with equal probability of an event p can be formulated as in Eq. 2.3. It is the smallest number of bits necessary to encode the probability p.

$$I(p) = -log_2(p) \tag{2.3}$$

The word translation entropy $H(s)$ of a source token s, then, can be phrased as "the sum over all observed word translation probabilities (i.e. expectations) of a given ST word s into TT words $t_i...n$ multiplied with their information content" (Carl et al., 2016, p. 31).

To apply the metric, all the translations of the segment concerned are put together to approximate the number of options a translator can choose from. As an example, if a source token is translated exactly the same by all translators then its entropy is $H(s) = 0$: there is only one option to choose from, so choice – in itself – is non-existent.

In this study, the calculation of word translation entropy is based on the final translations but in future research we intend to do away with the need of product data. In the case of word translation entropy, we plan to calculate it with information from large parallel corpora.

In contrast with word translation entropy, which operates on the lexico-semantic level, syntactic equivalence is a syntactic feature. In TPR-DB's generated features, there are two particularly interesting syntactic equivalence features that map the amount of word re-ordering that has to take place to transform the source text to the target text or vice versa. These features are called `CrossS` and `CrossT` respectively. In our study, we are only interested in going from the source text to the translation `CrossS`.

Figure 2.2 visualises such a re-ordering procedure. The higher the value for `CrossS`, the more re-ordering steps have to take place to generate the target text. The more syntactic transformations a translation requires, the higher the difficulty of that translation task.

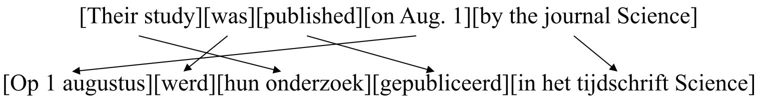

[Their study][was][published][on Aug. 1][by the journal Science]
[Op 1 augustus][werd][hun onderzoek][gepubliceerd][in het tijdschrift Science]

Figure 2.2. An illustration of syntactic re-ordering from English to Dutch

In the following section, a couple of methodological notes on the used correlation metrics are highlighted. They are necessary to provide a comprehensive overview of the results later on.

2.3.3 Correlation Metrics

In early tests it became clear that our data is not linearly distributed (e.g. Figure 2.3) and outliers are frequent. Therefore, we opted to use Kendall's tau

τ as correlation metric. When calculating correlations, all features have been normalised by the number of source tokens in the segment, hence the prefix `Norm` in the labels in Figure 2.3. For conciseness' sake, we do not prepend `Norm` to the feature names in the text, but it is important to keep in mind that they have been normalised by the number of source tokens.

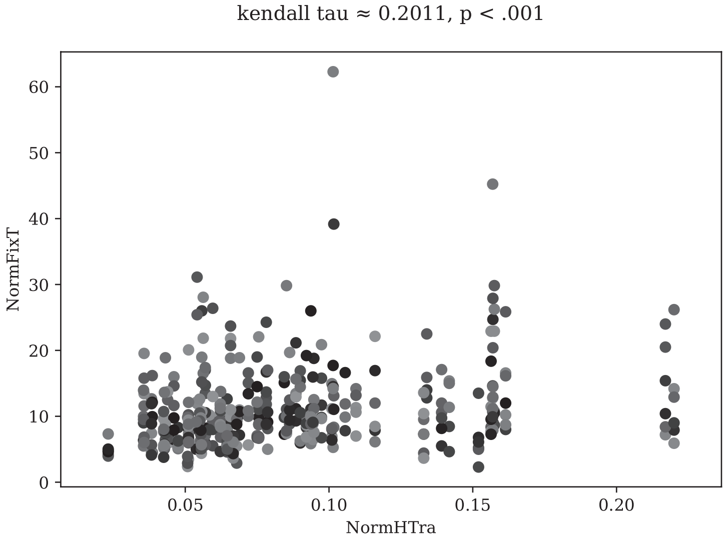


Figure 2.3. Scatter plot showing a non-linear distribution of data points over entropy (`HTra`, x-axis) and fixation on the target text (`FixT`, y-axis). Data set restricted to professional translators

For the feature `EC_TOT` that designates the number of errors that were annotated in a segment, we only look at a subset of the data (242 data points), namely only those segments where the error count is larger than zero. In other words, we are only interested in the final translations of segments that contain errors. When a segment has errors, it can be assumed that it was difficult to translate but when a segment has been translated without errors this does not imply that it was easy to translate.

Because we are interested in the difference between professional translators and students, both data sets have been analysed separately. By doing so, differences between professionals and students (if any) are emphasised.

2.4 Results

This section discusses the results of the correlation tests. Only features with a significant correlation (*$p < .05$) are discussed. The absolutely largest correlation between students and professionals is highlighted in **boldface**.

	Duration			Revision				Gaze	
	APR	Pausedur	Pdur	Mdel	Mins	Nedit	Scatter	FixS	FixT
prof	−.1746*	.1606*	.1240	.0556	.1498*	.2858*	.1204	.1545*	.1842*
stud	.0005	**.1918***	.0206	.0281	.0647	**.3651***	−.0312	**.1778***	.0992

$^*p < .05$

Table 2.3. Correlations between error counts (`EC_TOT`) and process features

2.4.1 Error Count

As mentioned before in Section 2.3, the number of errors made in each translated segment (`EC_TOT`) have been manually added to the data set using a fine grained error typology (Daems, 2016). In this study, we are not interested in the type of errors but in the number of errors.

Table 2.3 shows that the average pause ratio (`AvgPauseRatio`) is negatively correlated with the number of errors in the final translation. This correlation only exists for professional translators (−.1746) and it means that the more errors were made in a segment, the lower `AvgPauseRatio` would be. More verbosely, it means that the more errors occur, the smaller the average pause is relative to the average time per word. Note that this does not mean that a segment with a higher error count had less pause time. The opposite is shown in `Pausedur`, which is correlated with the number of errors too (prof. .1606, stud. .1918). There is no significant correlation between the total time of keyboard activity excluding keystroke pauses $\geqslant 1s$ (`Pdur`) and the number of errors. The number of typed characters (`Mins`) is significant only for professionals (.1498), meaning that the more errors have been made, the more characters the translator will have typed. More obvious is the strong correlation between `Nedit` and number of errors (prof. .2858, stud. .3651). This correlation implies that the more often a translator has gone back to a segment to revise it, the more errors will have been made, especially by students.

Finally, gaze is also moderately correlated with larger error counts. Fixations on the source text are correlated with error count for both professional translators and students (prof. .1545, stud. .1778). Fixations on the target text, however, are only of importance for professionals (.1842). This may indicate that professional translators spend consistently more time fixating on the target text when having difficulties than students.

2.4.2 Word Translation Entropy

Word translation entropy (`HTra`) is a quantifiable way to indicate the number of translation choices at word level that a translator is confronted with.

Table 2.4 shows that `AvgPauseRatio` has slightly lower absolute values for word translation entropy than it had for error count. More interesting, though, is that `AvgPauseRatio` negatively correlates with word translation entropy for professional translators as well as students of translation studies

	Duration			Revision				Gaze	
	APR	Pausedur	Pdur	Mdel	Mins	Nedit	Scatter	FixS	FixT
prof	**−.1160***	.1854*	**.1668***	**.1038***	**.2068***	.3729*	.0479	**.1567***	.2011*
stud	−.1119*	**.1864***	.1338*	.0930*	.1576*	**.4708***	.0568	.0991*	.0643

$^*p < .05$

Table 2.4. Correlations between word translation entropy (HTra) and process features

(prof. −.1160, stud. −.1119). Pausedur is correlated with word translation entropy (prof. .1854, stud. .1864) similar to its correlation with error counts: both product features correlate positively with Pausedur and the correlation is stronger for students than for professional translators. Where the correlation with Pdur was not significant for EC_TOT, it is for HTra (prof. .1668, stud. .1338). This means that the number of options a translator has to choose from influences the total time that a translator spends typing. In addition, the number of deletions and insertions is also indicative. For Mdel the correlation is rather small (prof. .1038, stud. .0930), but it is larger for Mins (prof. .2068, stud. .1576). Nedit scores high correlations again (prof. .3729, stud. .4708) in addition to be highly correlated with the error count shown before. Considering gaze it can be seen that there is a similar tendency as in the correlations with error counts, but here FixS is more strongly correlated for professionals (prof. .1567, stud. .0991). FixT is not correlated for students here either (prof. .2011).

2.4.3 Syntactic Equivalence

CrossS is a feature that indicates the amount of syntactic re-ordering that was needed to transform the source segment to the target segment. Moving phrases around and re-ordering the translation requires more cognitive effort (Sun, 2015). The correlations are given in Table 2.5.

	Duration			Revision				Gaze	
	APR	Pausedur	Pdur	Mdel	Mins	Nedit	Scatter	FixS	FixT
prof	**−.1526***	**.1482***	**.1901***	**.1371***	**.2661***	**.3098***	.0817*	.1460*	.2158*
stud	−.1168*	.1153*	.0926*	.0753*	.1398*	.1555*	−.0345	.0213	.0614

$^*p < .05$

Table 2.5. Correlations between syntactic equivalence (CrossS) and process features

AvgPauseRatio seems to correlate similarly with syntactic equivalence as it does with word translation entropy with a small improvement for professional translators (prof. −.1526, stud. −.1168). The correlation is negative, indicating that more re-ordering steps can be observed when there is a smaller average pause ratio. Pausedur (prof. .1482, stud. .1153) and Pdur (prof. .1901, stud. .0926) are also correlated with CrossS, meaning that syntactic non-equivalence generally occurs in segments with longer pauses and a longer production time.

Syntactic re-ordering is correlated with all revision features for professional translators. Mins is correlated especially for professional translators (prof. .2661, stud. .1398), and so is Mdel (prof. .1371, stud. .0753). Scatter (prof. .0817) is not correlated for students. For Nedit the correlation is almost twice as strong for professional translators as it is for students (prof. .3098, stud. .1555).

Gaze information, finally, is only correlated with syntactic equivalence for professional translators and not for students. In other words, when more syntactic re-ordering is required, professionals will have more focused fixations on the source (.1460) as well as on the target text (.2158).

2.5 Discussion

Going over the results presented above, we see that average pause ratio is correlated with all three product features, with the exception of a correlation with the number of errors (EC_TOT) for student translators. Because of the way average pause ratio is calculated, as the average duration per pause divided by the average production time per word, there are multiple ways to explain this correlation. The negative correlation for all product features, seems to imply that for high values for the product feature, the average duration of a pause relative to the average duration of a word decreases, or that the average time per word increases. What this means is that either the number of pauses in a segment increases (without changing the total duration of the pauses) when a product feature increases, or it means that the total pause duration decreases without influencing the number of pauses. In the case that the average time per word increases, it can only mean that translating the segment takes longer. This latter option is excluded for the number of errors (EC_TOT) as it does not correlate with the total production duration (Pdur), and we would expect a correlated connection between the average pause ratio (AvgPauseRatio) and the total production duration (Pdur) in that case.

The number of revisions that a translator made to a segment is the highest correlating feature (Nedit). It is correlated with all studied product features. This is especially true for word translation entropy with correlation values of .3729 for professional translators and .4708 for students. For error counts (prof. .2858, stud. .3651) and syntactic equivalence (prof. .3098, stud. .1555) the values are lower. In the latter case, the correlation is notably stronger for professionals, which is in contrast with the two other product features. This means that a higher translation difficulty is likely when translators have gone back to a segment's translation more often to revise it.

The correlations with other process features are also significant in most cases, with the exception of disrupted keyboard activity (Scatter), which is only slightly significantly correlated with syntactic equivalence (`CrossS`) and only so for professional translators (.0817), and with the exception of the total production duration (`Pdur`) and the number of deleted characters (`Mdel`), which are not significantly correlated with error counts.

Students and professional translators clearly behave differently. For students, correlations between process and product data are not significant more often. Of particular interest is the persistent lack of correlation between fixations on the target text (`FixT`) and product data for students. The statement by Sun (2015) that especially inexperienced translators have problems with equivalence cannot be verified nor refuted with our data. We can only conclude that there is indeed a clear difference between students and professional translators and that for students disrupted keyboard activity (`Scatter`) and gaze information are not correlated with syntactic equivalence.

We are aware that Mishra et al. (2013) developed a system to automatically predict translation difficulty. To the best of our knowledge, this has not been done before. In their set-up, the researchers rely on eye-tracking data. More particularly they use the sum of fixations and saccades on source and target text divided by the sentence length as an indicator of translation difficulty. They then correlate this translatability indicator with three intrinsic sentential properties, namely sentence length, degree of polysemy, and structural complexity. That allowed them to use these properties to train a Support Vector Regression system. However, "[their] claim is that translation difficulty is mainly caused by three features: *Length*, *Degree of Polysemy* and *Structural Complexity*" (Mishra et al., 2013, p. 348), which are all source-text features. The other side of the coin, namely translation-specific difficulties, is not taken into account. We are convinced that features derived from the relation between the source and target language (i.e. language-pair specific features) are important, too, and we will strive to include such features in our system.

2.6 Conclusion

In this paper, we have selected three difficulty-predicting features that were proposed in related research (number of errors, word translation entropy, syntactic equivalence) and we have correlated them with translation process data (features that can be categorised as duration, revision, and gaze information). By putting these data together, we have provided an insight in how particularities in the process data change with the difficulty of the translation task.

We have shown that the proposed difficulty-predicting features correlate with process data, as an intermediary indicating cognitive effort. In other words, when a segment is hard to translate for a translator, the translation process is different from a segment that is easy to translate. It follows that these product features can be used as predictors for translation difficulties. In our study, we have found that the impact of word translation entropy and syntactic equivalence is similar to that of the number of translation errors. With respect to future research, this is interesting information. Even though error counts can only be incorporated in an analysis after the translation process has finished, and after the target text has been annotated, entropy and equivalence may be modelled *off-line*, i.e. without the need of a translation. Future research will address this intuition. We will borrow techniques from computational linguistics and investigate the feasibility of calculating word translation entropy and syntactic equivalence values a priori with the help of parallel corpora and dependency parsers. If the outcomes of these studies are positive, word translation entropy and syntactic equivalence can be used as important difficulty indicators in the final translation difficulty predicting system.

Chapter 3

Predicting Syntactic Equivalence Between Source and Target Sentences[1]

Bram Vanroy* · **Arda Tezcan*** · **Lieve Macken***
*LT^3, Language and Translation Technology Team, Ghent University

Abstract
The translation difficulty of a text is influenced by many different factors. Some of these are specific to the source text and related to readability while others more directly involve translation and the relation between the source and the target text. One such factor is syntactic equivalence, which can be calculated on the basis of a source sentence and its translation. When the expected syntactic form of the target sentence is dissimilar to its source, translating said source sentence proves more difficult for a translator. The degree of syntactic equivalence between a word-aligned source and target sentence can be derived from the crossing alignment links, averaged by the number of alignments, either at word or at sequence level. However, when predicting the translatability of a source sentence, its translation is not available. Therefore, we train machine learning systems on a parallel English-Dutch corpus to predict the expected syntactic equivalence of an English source sentence without having access to its Dutch translation. We use traditional machine learning systems (Random Forest Regression and Support Vector Regression) combined with syntactic sentence-level features as well as recurrent neural networks that utilise word embeddings and accurate morpho-syntactic features.

Publication: Vanroy, B., Tezcan, A., & Macken, L. (2019). Predicting syntactic equivalence between source and target sentences. *Computational Linguistics in the Netherlands Journal*, *9*, 101–116.

[1] In the original publication, reproduced here, there is an error in the sequence alignments in Ex. 10. The unaligned word "me" was incorrectly not considered. The right alignments here are 0-0 1-2 2-1 4-3 5-5 6-4 7-6

3.1 Introduction

In translation studies, equivalence is a concept that indicates how a source text and its translation can be compared to each other. It works on low-level language features such as morphology, lexicon, and syntax, as well as on higher-level, general text properties such as semantic, pragmatic, and cultural planes (Baker, 2011). In the current study, we focus on syntactic — that is, *structural* — equivalence, which we operationalise as the amount of reordering that is necessary to transform a source sentence into a target sentence. When the syntactic equivalence between a source and target sentence is high (i.e. they are structurally similar), no or few reordering steps are needed to transform the source text's syntactic form into the target structure. When syntactic equivalence is low, many reordering steps are required to create a good translation. We investigate two approaches of quantifying syntactic equivalence. The first one is based on word alignment whereas the second focuses on the alignment of sequences of words. How exactly we calculate these metrics will be discussed in Section 3.3.1.

In light of the PreDicT project[2] (Predicting Difficulty in Translation), this study aims to estimate an English source sentence's syntactic equivalence to an implied Dutch translation without the need for that translation. Our previous study shows that syntactic equivalence is correlated with cognitive effort, and thus the translation difficulty of a text (Vanroy, De Clercq, & Macken, 2019). More related research will be discussed in Section 3.2. To predict syntactic equivalence, we use the wealth of data available in word-aligned source and target sentences of an English-Dutch parallel corpus to train machine learning models that can predict a source sentence's syntactic equivalence. The dataset used will be discussed in Section 3.3.2, followed by a detailed run-down of the machine learning systems that we have put to the test (Section 3.3.3). Then we show the results for both the approach based on a word-alignment corpus and the one utilising sequence alignment (Section 3.4), followed by a discussion in Section 3.5. Finally, we will provide some hints towards future research in Section 3.6.

3.2 Related Research

Translatability is a topic in translation studies that is still open to much debate. Historically, there has been much discussion of whether or not texts can in fact be truly translated, feeding an existential sense of untranslatability. This absolute viewpoint is much less present today, even though "it is assumed that the perfect translation, i.e. one which does not entail any losses from the original is unattainable" (de Pedro, 1999, p. 556). In this paper we understand translatability as the difficulty of translation rather than the

[2] `https://research.flw.ugent.be/en/projects/predict`

more philosophical possibility or attainability of translation. It has been suggested that the difficulty of translating a text can be measured by readability formulas (Jensen, 2009), and that it can be modelled by using only source text features (Mishra et al., 2013). However, empirical research found that a text's translatability is only in part related with its readability (Sun & Shreve, 2014). In addition to source text properties, the difficulties that translators are faced with can also be attributed to language-pair specific features. One such feature is equivalence, or the lack thereof, between a source text and its translation (Sun, 2015). As mentioned before, we put our attention to syntactic equivalence between a source and target text.

Even though quantitative research on syntactic equivalence with respect to human translation is scarce, syntactic equivalence is a much more discussed topic in the field of machine translation (MT), where it can be seen as synonymous for word reordering. Birch et al. (2008) show that the amount of reordering necessary between a source text and its translation is a strong predictor of the performance of a statistical machine translation (SMT) system. In other words, language pairs that require more reordering are more difficult to translate by SMT systems. The reaction to this particular translation difficulty was addressed by incorporating syntax into SMT systems. The interest for this problem was large, at the time, and many different solutions were proposed. Most of them require preprocessing the source text (often called *pre-reordering*) to better match the expected target sentence's structure. The extent of this topic surpasses the scope of the current paper, but for more details and different approaches see for instance Barone and Attardi (2013); Collins et al. (2005); Xia and McCord (2004); Yamada and Knight (2001).

In recent years, advances in deep learning gave rise to neural machine translation (NMT) systems, which outperform SMT in terms of translation quality and yield fewer errors across almost all error types, including word order errors (Bentivogli et al., 2016; Castilho & O'Brien, 2017; Van Brussel et al., 2018). Hence, researchers have posed the question whether pre-reordering steps are actually still necessary (Du & Way, 2017). Due to access to more context in NMT and the complexity and fine-grained feature analysis that neural networks are capable of, it is no surprise that NMT can implicitly learn word orders from training a translation model. In fact, Toral and Sánchez-Cartagena (2017) show that the reorderings that NMT introduces are closer to the reorderings in the reference translations than those by SMT. To further improve their performance, efforts have been made to teach NMT systems the linguistic nuances of natural language, particularly syntax and word (re)order(ing) (Huang et al., 2018; Zhang et al., 2017). Du and Way (2017) found that preprocessing a source sentence by reordering it actually lowers an NMT system's performance. Instead they suggest an alternative approach that improves translation quality by adding linguistic knowledge such as part-of-speech (PoS) tags and word class to Japanese-to-English and Chinese-to-English NMT systems. This addition was inspired by the work of Sennrich and Haddow (2016) on NMT for the language pair German and

English where they added lemmas, PoS tags, syntactic dependency labels, and morphological features to the input of the neural network, leading to an increased performance of the system. Similarly, on the task of detecting grammatical errors in SMT output, Tezcan et al. (2017) showed that word-level morpho-syntactic features, consisting of PoS, dependency and morphology information, yield better results than using word embeddings as a word representation technique. This approach is closely related to the recurrent neural network (RNN) architecture that we test in this study on the task of predicting syntactic equivalence (cf. Section 3.3.3.3). In a multi-task set-up, Niehues and Cho (2017) successfully used PoS-tagging as a secondary task next to neural machine translation. The idea being that the model learns the importance of part-of-speech tags, and that this information propagates to the MT task. Eriguchi et al. (2016) provided evidence that attentional NMT systems can be extended with a linguistic tree representation of the source text for English-to-Japanese translation. A similar idea was worked out by Bastings et al. (2017), who made use of graph convolutional networks (GCN) to encode the source text as syntax-aware word representations through syntactic dependency trees. This information contributes to improve over their baseline without syntactic information for English-German and English-Czech. Conversely, presenting the target text as a linguistic structure, Aharoni and Goldberg (2017) found that using a string-to-tree model can improve the performance of German-to-English NMT over a traditional string-to-string variant. Here, the target text is presented as a linearised and lexicalised constituency tree. See Currey and Heafield (2018) for a non-exhaustive yet comprehensive overview of efforts to incorporate syntax into RNN-based NMT systems. The authors also introduce their own approach of injecting syntactic information into an English-to-German NMT system by using both the source text as well as its linearised constituency parse as input. Especially the multi-task system performed well, improving over the baseline.

Generally speaking there seems to be an iterative process of linguistic features being added to existing translation systems to try to further improve their performance.

Even though the above only highlights the difficulties that MT systems experience as a result of the lack of linguistic knowledge, there is also some evidence that suggests that the syntax of a source and of its target text play a role for human translators. Bangalore et al. (2015) found that syntactic variation — that is, syntactic entropy – correlates with cognitive effort. In other words, the more possible variations in the target text structure, the more difficult the translation process is. In this paper, we are more interested in syntactic equivalence between a source and target text rather than the syntactic entropy of a given source sentence. Translation difficulty is often reduced to source text features such as its readability, but Sun and Shreve (2014) (reiterated in Sun, 2015) use the term *equivalence* to discuss translation-specific difficulty that is not solely related to the source text. The (syntactic) equivalence between a source text and its (possible) translation can cause difficulties for a translator when the source and target syntactic structure differ significantly. We refer the reader to a previous study (Vanroy, De Clercq, & Macken, 2019), where we correlated translation process features with syntactic equivalence. We found that, indeed, a correlation exists between word reordering and a number of translation process features (taken from duration, revision and gaze categories) as a proxy for cognitive effort and, thus, translation difficulty. We take this to mean that the more transformations the source word order has to undergo during translation, the more cognitive effort is required by the translator. Considering that the goal of PreDicT is to build a translatability predicting system, we wish to model syntactic equivalence of a sentence off-line, that is, without the need of a target sentence which brings us to the current study.

3.3 Methodology

First, we elaborate on how we quantify syntactic equivalence by distinguishing two approaches in Section 3.3.1: word alignment and sequence alignment. Then we discuss the used dataset (Section 3.3.2), followed by the experimental set-up consisting of a baseline (3.3.3.1), a traditional machine-learning (ML) approach using sentence-level features (3.3.3.2), and finally a neural network that uses word-level features (3.3.3.3).

3.3.1 Alignment Types

We present two similar approaches to quantify syntactic equivalence: the first quantifies how words have moved position during translation; the second takes the movement of word sequences into account rather than single words. The core idea is that we calculate syntactic equivalence as the number of times alignment links cross each other (hence its name *cross* value), averaged by the number of alignment links. Visual examples are given below.

Our first approach works on the word level: by looking at how each individual word has moved with respect to other words in the sentence, we calculate its `word_cross` value. Our measure of syntactic equivalence is bidirectional (or symmetrical) and applicable to either translation direction (English-to-Dutch or Dutch-to-English). This contrasts with Carl et al. (2016), who introduce a similar metric but which is asymmetrical, i.e. the resulting Cross value differs depending on the translation direction. Typically, a word-aligned corpus is represented in the Pharaoh format. The format requires that every alignment link is shown as a pair of source and target word indices $\mathrm{s_i}$-$\mathrm{t_j}$. The indices indicate the position of that token in its sentence. An example is given below where 8a is the source sentence, 8b the target sentence, and 9 a representation of the word alignments in Pharaoh format.

(8) a. Sometimes she asks me why I used to call her father Harold .

b. Soms vraagt ze waarom ik haar vader Harold noemde .

(9) 0-0 1-2 2-1 4-3 5-4 6-8 7-8 8-8 9-5 10-6 11-7 12-9

We can visualise this as follows in Figure 3.1. The arrows[3] indicate the alignment links, i.e. where a given source word has moved to in the target sentence. The circles highlight where alignments cross one another; these are the *crosses*. In this example, we count ten crosses. This value is then averaged by the number of alignments (arrows) to get our final, average cross value of the whole sentence. In this case that is $^{10}/_{12} = 0.8333...$.

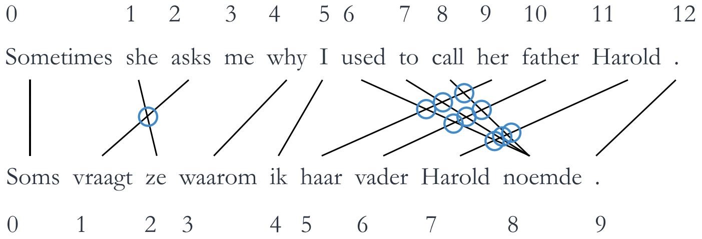

0 1 2 3 4 5 6 7 8 9 10 11 12
Sometimes she asks me why I used to call her father Harold .
Soms vraagt ze waarom ik haar vader Harold noemde .
0 1 2 3 4 5 6 7 8 9

Figure 3.1. A visual representation of word alignment and cross values

The second approach that we tested is based on sequential words that move together as a group (`seq_cross`). The intuition here is that words that move together are one unit and as such can count as one alignment link. To this end, we seek the longest possible word sequence alignments between the source and target sentences with the following criteria:

(i) each word in the source sequence is aligned to at least one word in the target sequence and vice versa;
(ii) each word in the source word sequence is only aligned to word(s) in the target word sequence and vice versa;
(iii) none of the alignments between the source and target word sequences cross each other.

The visualisation of Example 8 as sequence alignment in Figure 3.2 makes this clear. To illustrate: *why I* is seen as a sequence because they are sequential (there is no aligned word between them nor between their translations), and the order is the same (*why* is aligned with *waarom* which stands before the $I \rightarrow ik$ alignment link), so they do not cross each other. These alignments are also shown in Pharaoh format in Example 10.

[3] In the original publication, arrows were used instead of alignment *lines*. For consistency with following chapters, the figure has been replaced with a figure without arrows.

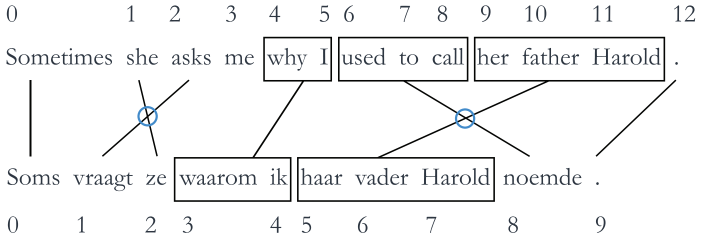

0 1 2 3 4 5 6 7 8 9 10 11 12
Sometimes she asks me why I used to call her father Harold .
Soms vraagt ze waarom ik haar vader Harold noemde .
0 1 2 3 4 5 6 7 8 9

Figure 3.2. A visual representation of sequence alignment and cross values

(10) 0-0 1-2 2-1 3-3 4-5 5-4 6-6

In the example, the cross value based on sequences is $^2/_7 = 0.286$. We would argue that, intuitively, using crosses of sequence alignment better represents the syntactic shifts that a source sentence has to go through to become the target sentence because it indicates crossing groups of words rather than single entities. Therefore, our hypothesis is that cross values based on sequence alignment can be modelled better than those based on word alignment. To distinguish systems that use word alignment as input from those using sequence alignment, we will identify the former with `WORD` and the latter with `SEQ`.

Even though not similar, Birch et al. (2008) propose a word reordering metric that works by maximising aligned block pairs, too. The authors use this metric as a predictor for the performance of machine translation systems. A large difference with our approach is that they average their metric by the number of source tokens in a sentence whereas we average by the number of alignments. This is important, because in our approach, both for word alignment and sequence alignment, calculating cross is direction-agnostic, meaning that going from source text to target sentence or vice-versa will yield the same cross value. As such it is, indeed, a syntactic equivalence metric that compares two sentences irrespective of the translation direction. Do note that sequences are not linguistically motivated entities but only sequences that move as a single unit. Sequences of words have been used as (structurally useful) phrases before, perhaps most notably by the (H)TER evaluation metric which takes sequence shifts into account to calculate the edit distance between a translation and a reference (Snover et al., 2006, (Human-targeted) Translation Edit Rate).

Because our aim is to predict syntactic equivalence automatically, which we measure with cross values in this study, we rely on automatic word alignment methods, which are error-prone. Therefore, we first analyse the quality of two widely-used automatic word alignment systems in order to (i) ensure that automatic methods are sufficiently accurate compared to human alignments; and (ii) to use the automatic method with the highest alignment quality in the remainder of this study. In the following test we compare the human, manual alignments (MA) of a small corpus, taken from the work of Macken (2010), to the output of GIZA++ (Och & Ney, 2003) and `fast_align` (FA) (Dyer et al., 2013) to see whether the quality of automatic word alignment systems is acceptable, and which tool performs best. For both GIZA++ and FA, we use the `grow-diag-final-and` algorithm, which starts with the intersection of the forward and backward alignments and then adds additional alignment points (Koehn et al., 2003). The dataset contains 143 sentences after filtering out outlying sentences that consist of less than three or more than 50 tokens in either source or target text. As word alignment tools work better with larger corpora, we obtained word alignments with GIZA++ and FA after appending the manually aligned data to the Dutch Parallel Corpus (DPC; Macken et al. (2011)). We evaluate the alignments between MA and GIZA++ and MA and FA with Alignment Error Rate (AER), as proposed by Mihalcea and Pedersen (2003), which is based on earlier work by Och and Ney (2000). We use the AER implementation of NLTK in Python (Bird et al., 2009).

	MA-GIZA++ ↓	**MA-FA ↓**
Min.	0.0	0.0
Max.	0.5758	0.6471
Mean	0.0822	0.1127
Median	0.0189	0.0691
SD	0.1110	0.1329

Table 3.1. Statistics about AER calculated on MA-GIZA++ and MA-FA

Table 3.1 shows that with mean AER 0.0822 and median AER 0.0189, GIZA++ is closer to manual alignments than FA (0.1127 and 0.0691, resp.). These results are in line with earlier findings by Peter et al. (2017) where GIZA++ outperformed `fast_align` in terms of alignment quality. We consider both AER scores to be sufficient for our task and that these automatic word alignment tools can be used as a proxy for manual alignment. Because of its better alignment quality, we will use GIZA++ as our word alignment tool of choice.

3.3.2 Data Set

In our experiments we use the Dutch Parallel Corpus (Macken et al., 2011), which, as the name implies, is a parallel corpus that centres around Dutch as its core language. We make use of both English-to-Dutch and Dutch-to-English parts of the corpus, which in total contain 148,421 sentences after removing duplicates and sentences that are shorter than 3 tokens or longer than 70 tokens. The training set consists of 144,421 sentences, leaving 2,000 sentences for the validation set and 2,000 sentences for the test set. The data was tokenised and lower-cased by using the preprocessing scripts[4] provided by

[4] `https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer`

Moses (Koehn et al., 2007).

3.3.3 Experimental Set-up

The objective of this paper is to predict the cross value of an English source sentence without having access to the Dutch target sentence. As mentioned before, we predict cross values based on word alignments as well as cross values based on sequence alignments. We train two types of systems with their own feature sets. The first type contains traditional ML systems combined with sentence-level features and the second one makes use of recurrent neural networks with word-level features. Additionally, we compare the estimation performance of the two ML approaches to a mean baseline, which we discuss first.

3.3.3.1 Mean baseline

As exemplified above, the cross values for the word alignment and sequence alignment of a sentence can differ. The mean cross value of the whole training set for `WORD` is 1.02 and for `SEQ` it is 0.91. We use these mean values as a baseline. More explicitly: for all 2,000 sentences in the test set, this baseline predicts 1.02 and 0.91 as cross values obtained from word and sequence alignments, respectively. The results for this approach will be referenced as `mean baseline` below.

3.3.3.2 Sentence-level features and RFR/SVR

In our traditional ML systems, we use sentence-level features as input (shown in 11), derived from the sentence level which contrasts with the word features in Section 3.3.3.3. These features have been chosen because — from a linguistic point-of-view — they provide information about the syntactic structure of a sentence. We used the Python package spaCy (Honnibal & Montani, 2017) to extract the required information from the source sentences.

(11)
- parse tree depth
- sentence length
- # coordinating conjunctions
- # subordinating conjunctions
- # punctuation marks
- # content words (adjectives, (proper) nouns, and verbs)
- # subjects
- # objects

We employ Random Forest Regression (`rfr`) and Support Vector Regression (`svr`) from the Python scikit-learn package (Pedregosa et al., 2011).[5]

[5] For more information about the fine-tuned parameters that we use in our experiments, see the package's documentation `https://scikit-learn.org/stable/documentation.html`.

Both `rfr` and `svr` are optimised through grid search with mean squared error (MSE) as the criterion to minimise between predicted and actual values. For `rfr` we tuned the number of trees (`n_estimators`) and found the best results with 500 trees (`WORD`) and 1000 in (`SEQ`). In the case of `svr`, the best parameters were $C = 1$ (C is the penalty of the error term), $\epsilon = 0.01$ (ϵ is the penalty-free distance with respect to training loss), using a radial bias function kernel. This is the case for both `WORD` and `SEQ`.

3.3.3.3 Word-level features and RNN

In the previous section, we introduced the traditional machine learning systems that we use. These systems are relatively fast and rather intuitive in that the features are hand-crafted. This can also be a downside, however: feature extraction is a time-consuming process and the generated features are often incomplete. Neural networks, on the other hand, try to learn high-level features from data and eliminate the need of domain expertise and feature engineering. Combined with the success of word embeddings (Mikolov et al., 2013; Pennington et al., 2014), especially in the last decade, neural networks have been producing superior results compared to traditional machine learning algorithms on various natural language processing tasks, including named entity recognition (Turian et al., 2010), parsing (Socher, Lin, et al., 2011), sentiment analysis (Socher, Pennington, et al., 2011), machine translation (Cho et al., 2014) and quality estimation of MT (Deng et al., 2018). Inspired by previous work on NLP and in addition to the traditional machine learning techniques described above, we use an RNN architecture for the task of predicting syntactic equivalence for a source sentence and an implied translation. RNNs can learn from a sequence of inputs rather than a single data point which makes them ideal for NLP tasks because sentences are sequences of words. Instead of representing a sentence as a single set of sentence-level features, which is the only option in the traditional ML systems, neural networks allow us to use a sentence as a set of words, which all have their own features. In other words, RNNs allow a sentence to be represented as a sequence of words, whereas traditional systems can only process a sentence as one unit. The RNN architecture that we have used will be discussed later in this section.

Word embeddings represent a word as a vector of size `n` based on its context and co-occurrences in a text. Each dimension in such a vector represents a latent feature of a given word, capturing useful syntactic and semantic properties (Turian et al., 2010). Despite its success and popularity on various NLP tasks, Tezcan et al. (2017) suggest that word embeddings, as a word representation technique, should not be considered as a one-size-fits-all approach. On the task of detecting grammatical errors in statistical machine translation (SMT) output, they report a marked improvement in performance over word embeddings by using accurate morpho-syntactic features. Moreover, in a more recent study, Tezcan et al. (2019) showed that the combination of morpho-syntactic features and word embeddings maximised the performance of an RNN system, in comparison to using either type of information alone, on the task of detecting all types of fluency errors in SMT output, consisting of both semantic and grammatical error types. Both studies suggest that such morpho-syntactic features provide complementary information to word embeddings when syntactic properties of texts are important in a given task. Considering the syntactic nature of the task at hand, namely predicting syntactic equivalence in translation, we use morpho-syntactic features, as an alternative word representation technique.

As described in Tezcan et al. (2017), we transform each token of a given sentence into its morpho-syntactic representation, in the form of a multi-hot encoded vector that provides accurate information about its part-of-speech tag, dependency label, and morphology. For every word, the vector values are set to 0 except for the morpho-syntactic features that apply to it, which are set to 1. Figure 3.3 shows the morpho-syntactic features that are extracted for the word *is*, in a given English sentence. In this example, the morpho-syntactic feature vector of the word *is* consists of all zeros except for the fields that represent its PoS tag (*VBZ*), dependency label (*Root*), and morphology information (*finite*, *present tense*, *singular*, and *third person*).

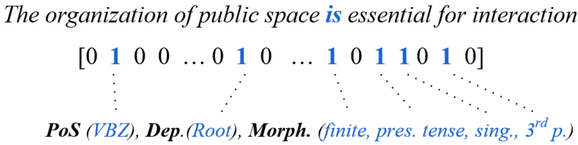

The organization of public space is essential for interaction.
[0 1 0 0 ... 0 1 0 ... 1 0 1 1 0 1 0]
PoS (VBZ), Dep.(Root), Morph. (finite, pres. tense, sing., 3rd p.)

Figure 3.3. A visual representation of how tokens are represented as morpho-syntactic, multi-hot encoded features

We used spaCy (Honnibal & Montani, 2017) to extract the aforementioned morpho-syntactic features on the English part of our dataset, as discussed in Section 3.3.2. We set the length of morpho-syntactic feature vectors to 147, namely the total number of possible features obtained by spaCy. The word embedding model was trained using word2vec (Mikolov et al., 2013) on a merged data set consisting of the English part of our dataset and the English news crawl dataset from the WMT shared task of 2017[6]. To keep the amount of information provided to the RNN system by the two types of input vectors balanced, we trained word embedding models with 147 dimensions.

To go into more detail about the architecture: we built an RNN architecture such that each input vector, word embedding or morpho-syntactic feature vector, is fed into a dedicated Bidirectional Gated Recurrent Unit (BiGRU). GRUs are a specialised variant of RNNs, which are well suited for capturing long range dependencies on multiple time scales (Cho et al., 2014). Bi-directional GRUs consist of two recurrent layers, each processing the given input

[6] Available at `http://data.statmt.org/wmt17/translation-task/`.

sequence in opposite direction, as a means to overcome the generalisation limitations of RNNs in general, which have the tendency to represent recent input nodes better (Bahdanau et al., 2015). As the final output of a GRU, we take the concatenation of the last state of each layer. When both morpho-syntactic features and word embeddings are provided as input, the outputs of both BiGRU layers are concatenated before they are connected to the output layer, which predicts the cross value for a given input sentence using a linear activation function. To help prevent overfitting, we apply dropout in the BiGRU layers (for the input gates and the recurrent connections) and between the BiGRU layers and the output layer (Srivastava et al., 2014).

We test three RNN architectures on this task, with different word representation combinations as input:

- `rnn_ms`: Only morpho-syntactic features
- `rnn_w2v` Only word embeddings
- `rnn_ms+w2v`: Both morpho-syntactic features and word embeddings

All systems were built with Keras (Chollet, 2015) on top of a TensorFlow backend (Abadi et al., 2015), using the Adam optimiser with a learning rate of 1×10^{-3}, hidden layer of size 200 and a batch size of 200. We used *tanh* as activation function between input and hidden layers, and *linear* activation between hidden and output layers. As loss function, we used MSE. All systems were trained for 100 epochs. We trained each system with three different dropout values, namely 0, 0.2 and 0.4, and kept the model that performed best on the development set as the best model. Figure 3.4 illustrates the proposed RNN architecture, which takes both morpho-syntactic features and word embeddings as input.

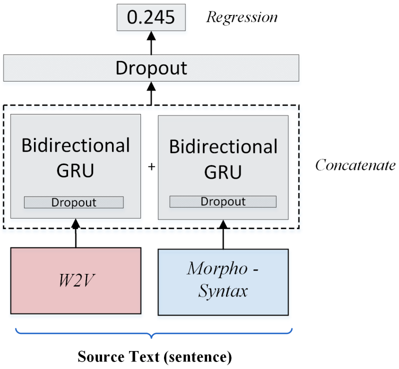


Figure 3.4. A visual representation of `rnn_ms+w2v`

3.4 Results

In this section, we use the Pearson correlation (r) between the predicted values and the actual cross values as our primary evaluation metric. This is done in order to be able to compare the results from `WORD` and `SEQ`: a sentence's cross value based on word alignment is different from the cross value based on sequence alignment. We also provide values for mean absolute error (MAE). MSE and MAE should be minimised whereas Pearson's r has to be maximised. In all results below, it holds that the Pearson correlations are significant ($p < .01$). The figures that are given here show all metrics (MSE, MAE and Pearson r) on the Y-axis, and all tested systems on the X-axis. Pearson r values are given in boldface, MAE in italics.

The predictive performance of the systems that we built using input based on word and sequence alignments are provided in Figure 3.5 and 3.6, respectively.

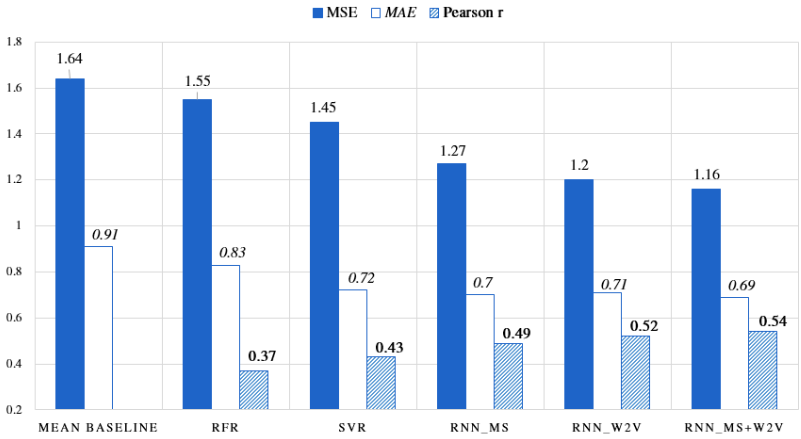


Figure 3.5. Visualisation of results for `WORD` (dataset mean of 1.64)

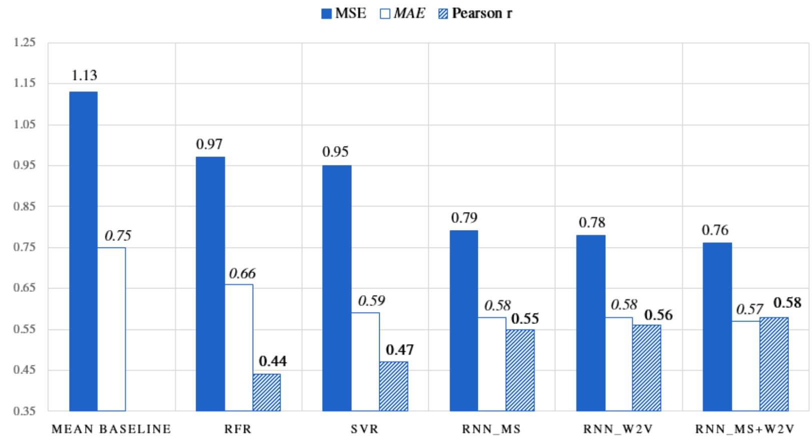


Figure 3.6. Visualisation of the results for SEQ (dataset mean of 0.91)

In a last graph (Fig. 3.7), the performance difference between systems using input based on word alignment and sequence alignment is highlighted.

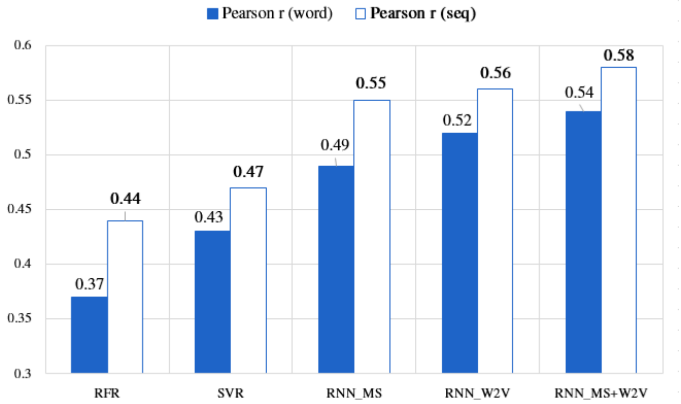


Figure 3.7. Comparison of WORD and SEQ

3.5 Discussion

The goal of this paper was to predict a Dutch source sentence's syntactic equivalence to an implicit English translation. To this end we introduced our version of a *cross* value, which can be based on word alignments (WORD) as well as on sequence alignments (SEQ), as discussed in Section 3.3.1. We found that traditional machine learning systems (rfr and svr) are less performant than recurrent neural networks (rnn_*) across the board. Furthermore, SEQ outperforms the WORD counterpart in all scenarios.

The neural network architectures with word-level features perform better than the traditional machine learning systems using sentence-level features. The best performing traditional ML system, svr, reaches a Pearson correlation of 0.43 (WORD) and 0.47 (SEQ), outperforming rfr (0.37 WORD, 0.44 SEQ). All RNN-architectures achieve better results, though. The reason for recurrent neural networks performing better than traditional ML is two-fold, and already touched upon before in Section 3.3.3. On the one hand, the traditional systems require single data point features as input, meaning that a sentence can only be represented as a number of features (cf. Section 3.3.3.2). These features are thus more coarse grained and not as detailed as word-level features. In contrast, when using neural networks a sentence can be represented as sequences of features. In other words, rather than having a single set of features for a sentence, that sentence can be represented as a sequence of word-level features, which gives much more detailed information to the system. Particularly, *recurrent* neural networks allow for the propagation of information through the sequence. This means that the final output takes into account the order of the words as well as the information of each word. On top of that, the architecture of our tested traditional ML systems are fundamentally different from neural networks. The latter is much more capable of modelling data-specific peculiarities while at the same time generalising sufficiently.

For the neural network systems, we can see that using only morpho-syntactic features (0.49 WORD, 0.55 SEQ) performs slightly worse than using only word embeddings (0.52 WORD, 0.56 SEQ). Because the task at hand is syntactic rather than semantic, it is worth expanding on the performance of word embeddings compared to morpho-syntactic features. Because our task is specifically directed at modelling syntactic changes, we expected a high importance of the morpho-syntactic features. Our morpho-syntactic features are very specific to each word in its context and role in the sentence whereas word embeddings are more general representations of words. Word2vec models do not specifically model syntax or morphology or even semantics; rather, they represent each word-type in relation to each other, implicitly modelling all kinds of language features, including morpho-syntactic and semantic. Because of the different goals of both word representations techniques (one specifically morpho-syntactic, the other more general), we had especially hypothesised them to be complementary, leading to a performance boost when combined. This is indeed the case, with a correlation of 0.54 (WORD), and 0.58 (SEQ).

Finally, comparing the Pearson correlations of both WORD and SEQ, we clearly see that cross values based on sequence alignment can be modelled better than those using word alignment.

In this paper we have presented a number of systems that can predict a source text's syntactic equivalence with an implicit translation, i.e. without needing an actual translation. In our tests, we reached a Pearson correlation of 0.58 but in future work we tend to improve this with more complex neural networks. Furthermore, and in line with Tezcan et al. (2019) who worked on a classification problem, we showed that for this specific task, a regression problem focused on a sentence's syntax, a morpho-syntactic component can be successfully used to improve the quality of predictions.

3.6 Future work

We have presented a way of calculating syntactic equivalence (with cross values) based on computational phrases, that is, phrases that are algorithmically created. Building phrases in this way is often used in automatic systems to create alignments between source and target sentences. However, we would also like to take the linguistic route, and compute the cross values based on linguistically motivated phrases. The phrases can be extracted automatically by using a constituency parser, but this introduces yet another automatic component prone to errors. Additionally, theoretical questions need to be answered concerning how the constituency tree should be segmented, and how to deal with linguistic phenomena such as separable verbs, conjunctions, (in)direct speech, interjected adverbs, and so on. Despite these challenges, we think linguistic phrases can improve performance over algorithmic phrases.

In our experiment we used recurrent neural networks, and even though they are powerful, they have been surpassed by the transformer architecture (Vaswani et al., 2017) in many natural language tasks. In future endeavours we will use transformers and investigate how well they perform for our given task.

The objective of the PreDicT project is to build a system that can predict a source sentence's translation difficulty. The present study discussed one feature (syntactic equivalence) that plays a role in predicting translatability, but we plan to test more features and add those to the final system. These features include semantic information such as word translation entropy, but also source text specific features such as syntactic and semantic complexity. Finally, language (pair) specific difficulties can be added, for instance the translation of the English -ing form to Dutch.

Chapter 4

Metrics of Syntactic Equivalence to Assess Translation Difficulty[1]

Bram Vanroy* · **Orphée De Clercq*** · **Arda Tezcan*** · **Joke Daems*** · **Lieve Macken***
*LT^3, Language and Translation Technology Team, Ghent University

Abstract
We propose three linguistically motivated metrics to quantify syntactic equivalence between a source sentence and its translation. Syntactically Aware Cross (SACr) measures the degree of word group reordering by creating syntactically motivated groups of words that are aligned. Secondly, an intuitive approach is to compare the linguistic labels of the word-aligned source and target tokens. Finally, on a deeper linguistic level, Aligned Syntactic Tree Edit Distance (ASTrED) compares the dependency structure of both sentences. To be able to compare source and target dependency labels we make use of Universal Dependencies (UD). We provide an analysis of our metrics by comparing them with translation process data in mixed models. Even though our examples and analysis focus on English as the source language and Dutch as the target language, the proposed metrics can be applied to any language for which UD models are attainable. An open-source implementation is made available.

Keywords: translation studies · computational linguistics · tree edit distance · syntax

Publication: Vanroy, B., De Clercq, O., Tezcan, A., Daems, J., & Macken, L. (in press). Metrics of syntactic equivalence to assess translation difficulty. In M. Carl (Ed.), *Explorations in translation process research.* Springer International Publishing.

[1] In this chapter, references to “other chapters in this book” refer to chapters in the original book publication

4.1 Introduction

Readability prediction is a well-studied problem. Traditional readability formulas (e.g. Flesch-Kincaid Grade Level (Kincaid et al., 1975), Gunning Fog Index (Gunning, 1952)) typically use shallow source text features such as average word and sentence length and word frequency to assess the reading difficulty level of a given text. Recently, more complex lexical, syntactic, semantic and discourse text features have been used (see for instance De Clercq and Hoste (2016); De Clercq et al. (2014); Francois and Miltsakaki (2012); Schwarm and Ostendorf (2005), and Collins-Thompson (2014) for an overview). The efforts in readability research contrast sharply with research into "translatability": there are no well-established methods yet to assess the difficulty level of a translation task. That is not to say that translation difficulty itself has not been studied, though. In fact, defining translation difficulty has been approached from a number of different directions.

It has been shown that genre, registerial and even cultural factors influence the choices translators have to make (e.g. Borrillo (2000, Section 3) concerning literary translation, and Steiner (2004) on registerial differences), which may introduce difficulties of its own. In addition, there is no doubt that individual translators may face different issues when translating the same text, and they may even choose to translate the same text differently (see for instance Dragsted (2012)). In this paper, however, we will focus on the source and target text itself.

According to Campbell (1999) and Sun (2015), translation difficulty can be attributed to linguistic source text factors and translation-specific factors. For the source text factors, we can refer to the vast literature on readability research (see the survey by Collins-Thompson (2014) for an overview), though a few findings specific to translation should be highlighted. Liu et al. (2019) demonstrated that *source* text complexity plays an important role in perceived translation difficulty, which supports earlier findings by Mishra et al. (2013). Mishra et al. introduced a metric of translation difficulty that is based on source text features alone, namely sentence length, degree of polysemy, and structural complexity. Campbell (1999) looked into translation difficulty from an empirical point of view and identified several source text elements that were difficult to translate across different target languages, such as multi-word units, complex noun phrases, abstract nouns and verbs. Campbell continued their research and developed the Choice Network Analysis (2000) in an attempt to model the mental process that underlies translation, particularly the multitude of choices that translators can choose from given a specific source text. Building on this, Carl and Schaeffer (2017) documented longer translation times when more elaborate choices were at the translators' disposal. This indicates that having more options available can increase the translation difficulty in terms of duration.

However, readability prediction and source text complexity alone do not suffice to adequately assess the *translation* complexity level of a given source text (Daems et al., 2013; Sun & Shreve, 2014). This is not surprising because readability prediction is not designed to take into account co-activation of shared bilingual resources. Specifically, Sun and Shreve (2014) and Sun (2015) state that translation-specific difficulties can be ascribed, in part, to the lack of equivalence due to inherent differences between languages. Hence, this paper will focus on the equivalence between the source and target text, specifically their syntactic similarity.

The notion of syntactic equivalence in a multilingual setting is not easy to define (see the next section) because syntax in itself is such a broad concept, so in this paper we restrict *syntactic equivalence between a source and target segment* to mean three things:

(12) a. differences in word (group) order;
b. differences in dependency labels of aligned words (e.g. a subject (`nsubj`) is translated as an object (`obj`));
c. differences in syntactic structure (dependency tree).

In Section 4.2 we will first discuss background literature concerning the importance of syntactic equivalence with respect to translatability and previous research of equivalence. In Section 4.3 we then introduce three linguistically motivated metrics to quantify syntactic equivalence between a source sentence and its translation. First, we introduce a metric to capture linguistic word group reordering (Syntactically aware cross; SACr). The next metric measures parse tree label changes between source and target sentences. Thirdly, we introduce a method to calculate tree edit distance between aligned dependency trees (Aligned Syntactic Tree Edit Distance; ASTrED). To illustrate the different proposed metrics, we will discuss two example sentence pairs in Section 4.4 to highlight how each metric accounts for different linguistic phenomena. As a proof of concept, we also apply our metrics to an existing dataset and measure the effect syntactic changes may have on the translation process by using mixed models (Sec. 4.5). Finally, we end with a conclusion and thoughts for future work concerning quantifying syntactic equivalence (Sec. 4.6).

4.2 Related Research

4.2.1 Background

In process-based translation studies, literal translation is conceived as the easiest way to translate a text and has been suggested as the default mode of translation, which is only interrupted by a monitor that alerts about imminent problems in the outcome (Tirkkonen-Condit, 2005, and Carl, this volume, Chapter 5). In other words, translators will translate a source text literally into the target text but as soon as an issue is encountered, translators stop working in the literal translation mode and try to find a more appropriate solution. Asadi and Séguinot (2005), for instance, observed that one group of translators processed the source text in short phrase-like segments. They translated while reading the text and followed the source language syntax and lexical items closely, but then rearranged the completed text segments to create a more idiomatic target text. Literal translation, in this sense of translating word-per-word, is identical to the concept of *simple transfer* in transfer-based MT, which can occur when the lexical surface forms are the only required differences between the source and target segment for a successful translation. In other words, when the underlying structure of the segments is the same, a literal translation can happen and only the lexical values need to be changed (Andersen, 1990; Chen & Chen, 1995).

From a cognitive perspective, literal translation is often explained by priming (Hansen-Schirra et al., 2017), i.e. the process in which the production of an output (in the case of translation, the target sentence) is aided or altered by the presentation of a previously presented stimulus (in the case of translation, the source sentence). Priming can occur at different linguistic levels including the morphological, semantic, and syntactic level.

In Carl and Schaeffer (2017, 46), building on earlier work (Schaeffer & Carl, 2014), "literal translation" is defined by three criteria:

(13) a. each ST [source text] word has only one possible translated form in a given context;
b. word order is identical in the ST and TT [target text];
c. ST and TT items correspond one-to-one.

To quantify the first criterion 13a, they use word translation entropy (HTra), which indicates the degree of uncertainty to choose a particular translation from a set of target words based on the number and distribution of different translations that are available for a given word in a given context. To measure the second and third criterion they use word crossings (Cross) calculated on word-aligned source-target sentences.

Criteria 13b and 13c for literal translation relate closely to what we consider syntactic equivalence as described in 12. 12a (differences in word (group) order) relates to criterion 13b (identical word order) above, and 13c is most similar to 12c: if ST and TT items do not correspond one-to-one, this must mean that the syntactic structure of the source and target sentences are different. In that respect, our interpretation for syntactic equivalence is closely linked, in part, to the definition of "literal translation" by Carl and Schaeffer (2017).

The affinity between "literal translation" on the one hand and equivalence on the other can also be seen in other research. Sun and Shreve (2014), repeated in Sun (2015), suggested that translation difficulties can be attributed to the lack of equivalence between the source and target text. Non-equivalence, one-to-several equivalence and one-to-part equivalence situations can be the root cause of translation difficulties. These situations can appear both at the lexical and syntactic level. However, Carl and Schaeffer (2017) note that it is possible that a source text has viable ("equivalent") translation options available, but that a plethora of choices actually implies that there is not one single, obvious translation equivalent. In our current study, we will follow the definitions of *natural* equivalence (Pym, 2014, Chapter 2), applied to syntax:

- equivalence is a relation of "equal value" between a source-text segment and a target-text segment;
- equivalence can be established on any linguistic level, from form to function;
- natural equivalence should not be affected by directionality: it should be the same whether translated from language A into language B or the other way round.

Pym (2014) juxtaposes natural equivalence with directional equivalence, which assumes that the equivalency relationship between a source and target text is asymmetric. For a discussion between the two approaches, see the particularly interesting discussion sections (Pym, 2014, Chapters 2.7, 3.9).

A similar idea to equivalence is that of translation shifts (Catford, 1965), which dates back to an approach to translation that is based on formal linguistics. Catford distinguished two major types of shifts, namely level shifts (e.g. shifts from grammar to lexis in distant languages) and category shifts (e.g. changes in word order or word class). They also contrast obligatory and optional shifts; the former refer to shifts that are imposed as a result of differences in the language systems, whereas the latter term is used to indicate optional choices of the translator.

Bangalore et al. (2015) introduced syntactic entropy and as such expanded translation entropy to the syntactic level. Syntactic entropy measures the extent to which different translators produce the same structure for one source sentence. They analysed a corpus of six English source texts translated into German, Danish and Spanish by a number of translators (24 for German and Danish and 32 for Spanish) and manually coded the following three linguistic features for all translations: clause type (independent or dependent), voice (active or passive), and valency of the verb (transitive, intransitive, ditransitive, impersonal) to quantify the syntactic deviation between translations of the same source text, which is their implementation of syntactic entropy. They obtained lower syntactic entropy values for target sentences that had similar linguistic features as the source segments and obtained higher syntactic entropy values for the cases where they diverged. Moreover, syntactic entropy had a positive effect on behavioural measures such as total reading time on the source text and the duration of coherent typing activity. This study is, to the best of our knowledge, the only study in this field that uses linguistic knowledge to quantify syntactic differences between a source text and its human translation. As an alternative to their three manually annotated linguistic features, we will suggest metrics that can be automatically derived from comparing the syntactic structures of the source and target sentences (Sec. 4.3).

Carl and Schaeffer (2017) used word-order distortion, measured by length of crossing links (called Cross) derived from word-aligned source-target sentences to measure the degree of monotonicity in translations. A bidirectional (symmetric) variant of Cross, which is applicable on either translation direction, was introduced by Vanroy, Tezcan, and Macken (2019) (from now on referred to as `word_cross`). Using word alignment in this way provides a fine-grained (word-based) method to quantifying syntactic equivalence. An alternative, coarse-grained, approach was suggested in Vanroy, Tezcan, and Macken (2019), who calculated cross on aligned word groups, or *sequences*, rather than single words to calculate syntactic equivalence between English source sentences and their Dutch translations (henceforth called sequence cross or `seq_cross`). These sequences, however, were not linguistically motivated but derived automatically adhering to a set of constraints. The lack of linguistic motivation in `seq_cross` prompted the creation of the three different metrics described in this paper. Each metric quantifies a different aspect of syntactic equivalence but all are based on linguistic knowledge, specifically the syntactic structures of the source and target sentences.

There are two main different ways of annotating syntactic structures: by means of a phrase structure or using a dependency representation. The phrase structure representation sees sentences and clauses structured in terms of constituents. The dependency representation, on the other hand, assumes that sentence and clause structures result from dependency relationships between words (Matthews, 1981). While the phrase structure representation is more suitable for analysing languages with fixed word order patterns and clear constituency structures, dependency representations, in contrast, are able to additionally deal with languages that are morphologically rich and have a relatively free word order (Jurafsky & Martin, 2008; Skut et al., 1997). The dependency relation that each dependency label represents is relative to its root (with the exception of the root node itself), and is effectively a *to*-relationship between the word and its root. For instance, in a sentence “He eats the cookies”, “He” is an `nsubj` (subject) *to* its root “eats”, “cookies” is an `obj` (object) to that root, and “the” is a `det` (determiner) to “cookies”. The dependency labels, then, are actually nodes in a directed acyclic graph, starting from the root node of the sentence (in the example “eats”) and recursively going down to its dependants. They can be represented as dependency *trees*. The dependency tree of the example sentence “He eats the cookies” above, can be visualised as in Figure 4.1.

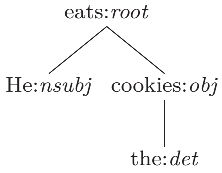


Figure 4.1. Example of a dependency tree of the sentence “He eats the cookies”

In recent years, research on automatic parsing methods has increased due to the availability of linguistically annotated corpora (treebanks) for many different languages (Hajič & Zeman, 2017; Peng et al., 2019; Zeman et al., 2018). However, despite their availability, the annotation schemes in treebanks vary significantly across languages, such as between the Swedish Treebank (Nivre & Megyesi, 2007), the Danish Dependency Treebank (Kromann, 2003), and Stanford Typed Dependencies (de Marneffe & Manning, 2008). Such differences, in turn, restrict multilingual research on and comparability of syntax and parsing (Nivre, 2015; Nivre et al., 2016), as well as research on natural language processing (NLP) that relies on automatic parsers trained on treebanks. Universal Dependencies[2] (UD) is an initiative to mitigate this problem by developing a framework for cross-linguistically consistent morphosyntactic annotation (Nivre et al., 2016), which we will discuss further in Section 4.3.1.

4.2.2 Word Alignment

The metrics suggested in this research aim to compare given source and target sentences to each other. As a starting point, the sentences need to be word aligned to be able to compare the source and target sides on the subsentential level. In word alignment, source words are aligned with target words as a way to find overlapping points of meaning and syntax. Aligned words should either carry meaning that is similar to their aligned counterpart, or should cover syntactic or morphological phenomena that are required to translate the aligned word into the desired language (Kay & Roscheisen, 1993). In that sense word alignment does not only involve semantic, conceptual agreement between a source and target sentence, but also the (morpho-)syntactic connections between them. As shown in Example 15c, alignments are typically written as pairs of indices of the aligned source and target words separated by a dash, e.g. `0-0 1-1 2-3 3-2 4-4`. Such alignments are often visualized with alignment tables (e.g. Och & Ney, 2000, Figure 1), but in this paper we opt for line diagrams such as Figure 4.2.

In the current paper, we manually aligned the source and target sentences in the examples, but in the global scope of our research, we are interested in translatability and we envisage to use large corpora to automatically detect and extract patterns that may be indicative of translation difficulties. Manually

[2] See `http://universaldependencies.org/` for label explanations, guidelines, and so on.

aligning those corpora is not feasible because of their size. Instead, we rely on automatic alignment systems. In previous research (Vanroy, Tezcan, & Macken, 2019), we justified using GIZA++ (Och & Ney, 2003) in favor of another tool, `fast_align` (Dyer et al., 2013), because of its lower Alignment Error Rate (Mihalcea & Pedersen, 2003; Och & Ney, 2000).

Because word alignment occurs on the fine-grained word level, the connections between larger groups of words on each side (source and target) is not taken into account. Take, for example, a simple English noun phrase (Ex. 14) that has been translated into a Dutch noun phrase. The determiners "The" and "De" are aligned, and the nouns "dog" and "hond" are aligned to each other. The alignments are given in Example 14b.

(14) a. The dog
De hond
b. 0-0 1-1

In this example, the linguistic relationship between the determiner and its noun is not present in the word alignments; it is not clear that the determiner and the noun are somehow linguistically connected. Generally speaking, this means that metrics based on word-based representation focus on the position and movement into the target language of single words. As an alternative approach, for one of our metrics (Syntactically Aware Cross (SACr); Section 4.3.2), we want to capture the alignment of word groups. In previous research (Vanroy, Tezcan, & Macken, 2019), we suggested a naive sequence-based approach, but SACr expands on that by including linguistic information to adjust those sequences. The goal is, then, to have a metric that is based on alignment information, but where the alignment is done between linguistically motivated groups instead of words or arbitrary sequences. In the example above, that would mean that "The dog" is aligned, as a group, with "De hond" rather than as single words. We will expand on aligning word groups rather than single words in the following sections.

4.2.3 Existing Word-reordering Metrics

The translation process research database (TPR-DB; Carl et al., 2016) implements a word-based, direction specific metric for reordering, and calculates a cross value based on the movements of words relative to the previously translated word.[3] Vanroy, Tezcan, and Macken (2019) take another approach by introducing a translation-direction agnostic variant that measures the number of times that translated words cross each other (`word_cross`). Example 15 (taken from Vanroy, Tezcan, & Macken, 2019, 104) is visualised in Figure 4.2, where each cross is emphasised with a circle. The total number of these crossing links is normalised by the total number of alignments, which constitutes

[3] We will not go into that version of Cross here but rather focus on our own implementations. See the original work for more details and Carl et al. (2019) for an analysis.

the `word_cross` value. The source and target segments can be aligned as shown in Example 15c. Note that "me" in the source text is not aligned to an equivalent on the target side. If the source sentence had been translated differently as "Soms vraagt ze mij waarom ...", "me" could have been aligned with "mij". However, in this specific translation, the indirect object is not made explicit so the source word is not aligned.

(15) a. Sometimes she asks me why I used to call her father Harold .
0 1 2 3 4 5 6 7 8 9 10 11 12

b. Soms vraagt ze waarom ik haar vader Harold noemde .
Sometimes asks she why I her father Harold called .
0 1 2 3 4 5 6 7 8 9

c. 0-0 1-2 2-1 4-3 5-4 6-8 7-8 8-8 9-5 10-6 11-7 12-9

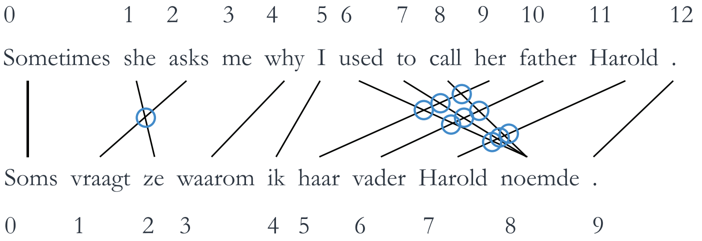

0 1 2 3 4 5 6 7 8 9 10 11 12
Sometimes she asks me why I used to call her father Harold .
Soms vraagt ze waarom ik haar vader Harold noemde .
0 1 2 3 4 5 6 7 8 9

Figure 4.2. Visualisation of `word_cross` in Ex. 15 with a total value of $^{10}/_{12} = 0.83$. (modified from Vanroy, Tezcan, & Macken, 2019)

This approach is word-based, but as discussed in Section 4.2.2, an alternative option is to encode the aligned order of the source and target sentences with aligned word *groups*, or *sequences*. For that reason, Vanroy, Tezcan, and Macken (2019) suggested to group consecutive tokens that are word-aligned to consecutive target tokens together to form a sequential cross metric (`seq_cross`). These sequences should be as large as possible while also adhering to the following constraints (Vanroy, Tezcan, & Macken, 2019, p. 104):

- each word in the source sequence (group) is aligned to at least one word in the target sequence and vice versa;
- each word in the source word sequence is only aligned to word(s) in the aligned target word sequence (and not to words in other target sequences) and vice versa;
- none of the alignments between the source and target word sequences cross each other.

Similar to `word_cross`, normalisation takes place based on the number of alignments, only here it uses the alignments between the sequences rather than the word alignments. Following these requirements, the example in Figure 4.2 can be modified so that instead of word movement, group movement is quantified (Figure 4.3).

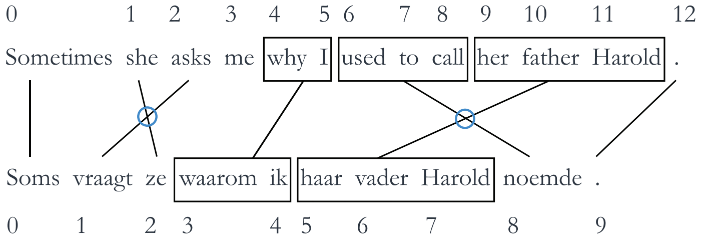

0 1 2 3 4 5 6 7 8 9 10 11 12
Sometimes she asks me why I used to call her father Harold .
Soms vraagt ze waarom ik haar vader Harold noemde .
0 1 2 3 4 5 6 7 8 9

Figure 4.3. Example of `seq_cross` in Ex. 15 with a total value of $^2/_7 = 0.286$ (modified from Vanroy, Tezcan, & Macken, 2019)

The problem with `seq_cross` is that, even though the metric works on the sequence level rather than the word level, its groups are linguistically arbitrary. Words are grouped together based on their relative reordering but irrespective of their linguistic properties (e.g. “why I” and “waarom ik” in the above examples). The need for grouping words founded on linguistic motivation gave rise to the current research. This specific issue involving word reordering is addressed in Section 4.3.2.

Motivated by the findings in previous studies, the main goal of this study is to introduce linguistically motivated, automatic, language-independent metrics to measure syntactic equivalence between source and target sentences in the context of translation.

4.3 Metrics

As discussed in Section 4.1, we restrict ourselves to three sub-components of syntactic equivalence,[4] namely word (group) order differences, changes in the dependency labels, and structural differences with respect to the source and target dependency trees. To address these three individual differences, we introduce three corresponding metrics. First, we build on `seq_cross` and propose an improved version to quantify reordering of syntactic word groups (syntactically aware cross, SACr, Sec. 4.3.2), then we discuss how label changes play a role (Sec. 4.3.3), and finally we introduce a method to calculate aligned syntactic tree edit distance (ASTrED, Sec. 4.3.4). A concise overview table of the metrics is given in Section 4.3.5. As all three metrics are based on comparing the syntactic structures of the source and target sentences using

[4] An open-source implementation of our metrics is available at `https://github.com/BramVanroy/astred`.

dependency representations, we start by explaining the chosen paradigm, Universal Dependencies, in closer detail.

4.3.1 Universal Dependencies

In all the metrics that we propose, we make use of UD annotation schemes (Nivre et al., 2016), which ensures comparable annotations across languages (see Sec. 4.2), such as the dependency labels of an English source text and its Dutch translation. To illustrate: the dependency trees of the source and target sentence of Example 15 are visualised in Figure 4.4[5] and 4.5. In both figures, the nodes' labels are formatted as `word_index:dependency_label:token`. As can be seen, the dependency labels of both trees use the same scheme, which allows for straightforward comparison between the source and target trees without the need to convert one tagset into another. That would not be feasible if the source and target sentences were using different, language-specific annotation schemes.

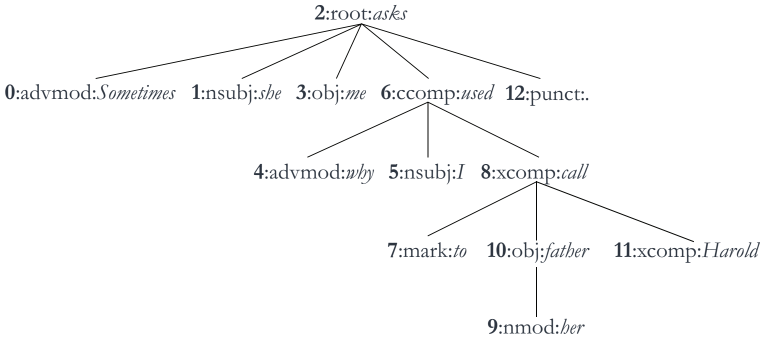


Figure 4.4. Source dependency tree of Ex. 15: "Sometimes she asks me why I used to call her father Harold ."

[5] Note that dependency trees are different from phrase-based trees. For a more theoretical deep-dive into the theory behind UD, we direct the reader to the work on Universal Dependencies (Nivre, 2015; Nivre et al., 2016; Nivre & Megyesi, 2007). Readers who are familiar with different dependency grammars may still disagree with the proposed trees, which may be due to the differences between UD and other grammars. For a critical comparison between UD and its alternatives, see Osborne and Gerdes (2019).

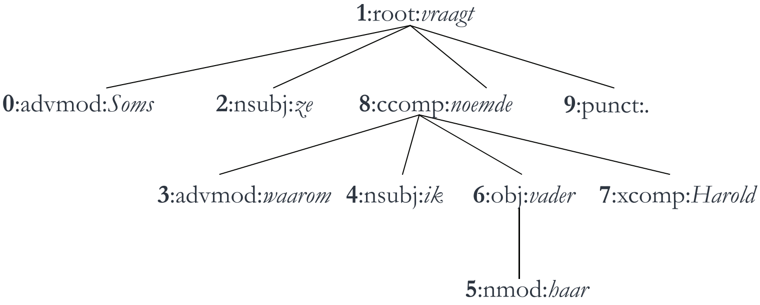


Figure 4.5. Target dependency tree of Ex. 15: "Soms vraagt ze waarom ik haar vader Harold noemde ."

To automate the parsing process, we depend on the recently introduced state-of-the-art `stanza` parser by the Stanford NLP group (Qi et al., 2020). In its annotation scheme, UD allows for language-specific extensions to the dependency relations to capture intricate properties of specific languages that may not generalize well to others languages. These extensions are also called *subtypes* because they always extend an existing UD dependency label. To minimize the effect of small language or model-specific differences, we take a general approach and discard these UD subtypes, so a label such as `obl:tmod` (an oblique, nominal, temporal argument) will be reduced to `obl`.

4.3.2 Syntactically Aware Cross

In Section 4.2, we referred to `seq_cross`, in which reordering is quantified based on word sequences, i.e. consecutive words that are grouped together when they adhere to given constraints, also called *sequences*. Syntactically Aware Cross (SACr) expands on `seq_cross` by verifying that the words in generated `seq_cross` groups are linguistically motivated. Figure 4.6 shows an example of what we are trying to achieve. In this figure, the sequences as defined in `seq_cross` are shown as dotted boxes. In SACr we verify whether these sequences are valid, linguistically motivated groups, and if this is not the case, we split the sequences up in smaller groups. The solid-line boxes in the figure represent those newly created, linguistically motivated groups. These groups (the initial `seq_cross` that were found to be valid SACr groups, and the new SACr groups that were created as a consequence of invalid `seq_cross` groups) are then used to calculate a syntactically aware cross value. Note that in this example, the number of crossing sequences has increased compared to the previous `seq_cross` value, as the sequence "Her father Harold" is now split up into two groups "Her father" and "Harold".[6]

[6] The sentence is ambiguous: "her father Harold" *could* be interpreted as a single phrase ("... her father, who is named Harold"), but here we assume that the correct meaning of the sentence is "... call her father (by the name) Harold".

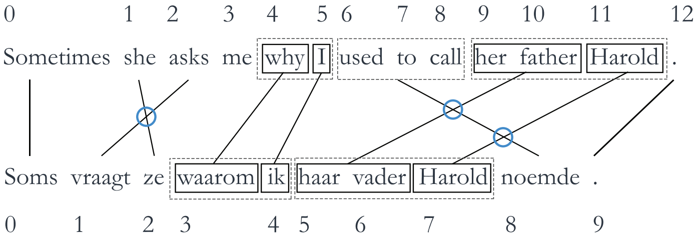

0 1 2 3 4 5 6 7 8 9 10 11 12
Sometimes she asks me why I used to call her father Harold .
Soms vraagt ze waarom ik haar vader Harold noemde .
0 1 2 3 4 5 6 7 8 9

Figure 4.6. Example of SACr with a total value of $3/9 = 0.33$. Dotted boxes indicate the initial groups of `seq_cross`. When required, these groups are split up into linguistically motivated SACr groups (solid boxes)

The criterion for SACr to establish linguistically inspired word groups is that, in addition to the criteria of `seq_cross`, all words in a group need to be "connected" to each other in the dependency tree: all nodes must exhibit one or more child-parent relationships with other nodes in the group. In practice, this means that siblings of a linguistic sub-tree can only be part of the same group if their parent is also in the group. More formally, we verify in a bottom-up, breadth-first fashion for each word that its parent in the dependency tree is also part of the same sequence group. The topmost node is excluded from the search because it cannot have a parent in this group. If all words in the group do not exhibit a child-parent relationship, the initial sequence group is not a valid SACr group. In such an event, in an iterative manner, a smaller sub-group of the initial sequence group is tested until a group is found for which the criterion above holds. We probe the largest sub-groups first and if no satisfying groups are obtained, smaller ones are tested (ultimately to the smallest size of two words) until no more groups can be found. This can mean that, for example, in an initial sequence group of four words only a valid sub-group of two words is found. As a consequence, the other two words will both be singletons (separate SACr groups consisting of only one word each).

Figure 4.7 and 4.8 illustrate which of the proposed sequence groups (cf. dotted boxes in Figure 4.6) are valid SACr groups in the dependency trees: when all items in a `seq_cross` group show a child-parent relation with other nodes in the group, the group is valid, but if not, new SACr subgroups will be created (e.g. "haar vader Harold" is an invalid group, but "haar vader" is a valid subgroup). In the following examples, square-cornered, blue groups are initial `seq_cross` groups that are also valid SACr groups. Round-cornered orange groups are initial `seq_cross` groups that are invalid SACr groups. Round cornered blue and dashed groups are new SACr groups that are subgroups of invalid `seq_cross` groups.

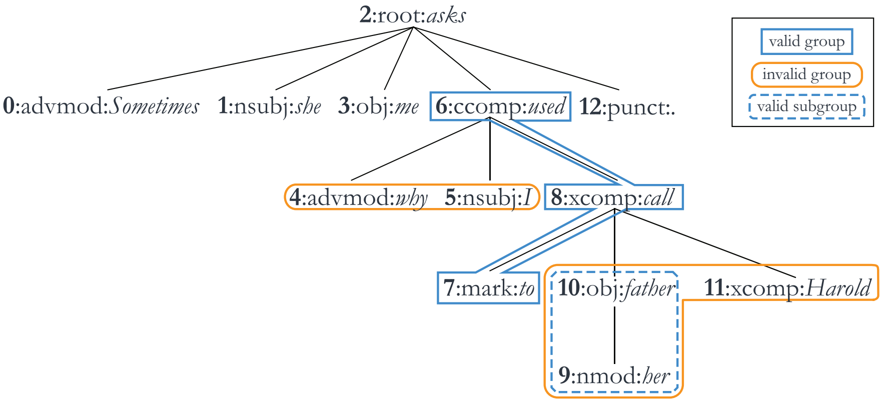


Figure 4.7. Source dependency tree of Ex. 15 with highlighted groups: "Sometimes she asks me why I used to call her father Harold ."

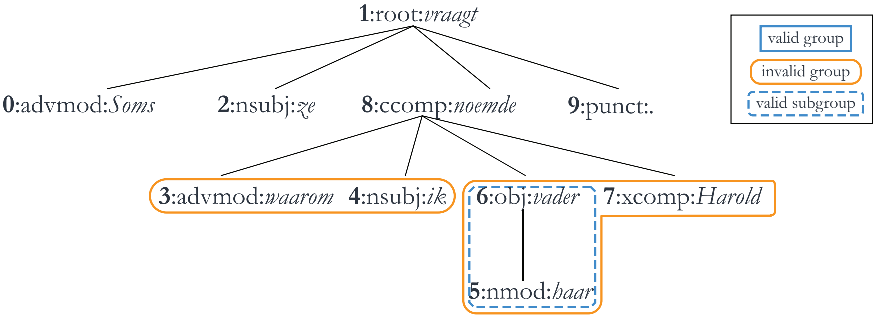


Figure 4.8. Target dependency tree of Ex. 15 with highlighted groups: "Soms vraagt ze waarom ik haar vader Harold noemde ."

Figure 4.6 above shows how the sequences from `seq_cross` have been adjusted according to the linguistic criteria derived from the dependency trees. This process can only increase the number of groups, not decrease them. In this particular case, the group "why I" and "waarom ik" are split into two groups again, namely "why" ("waarom") and "I" ("ik") because these words are not connected to each other in the dependency tree. In both the source and target tree, the adverb and pronoun are siblings but their root is not included in the group, causing them to not form a fully connected group. The group "used to call" remains unchanged because all words are connected in the source dependency tree. The corresponding groups "her father Harold" and "haar vader Harold" are also split up, because in the dependency tree "Harold" is not connected to "her father"/"haar vader". "her father"/"haar vader" are valid subgroups, though.

The final SACr value is the number of crossing alignment links between the source and target SACr groups, normalised by the number of these alignments. The example in Figure 4.6 counts three crossing links and nine total alignment links, leading to a SACr value of $3/9 = 0.33$. This contrasts with the word-based `word_cross` value of the same example, which is $10/12 = 0.83$, and the `seq_cross` value of $2/7 = 0.29$ (cf. Sec. 4.2.3).

The main distinction between our three proposed cross metrics (`word_cross`, `seq_cross` and SACr), is the size of the unit they use to calculate crossing links with. In `word_cross`, the reordering of single words is quantified. Alternatively, reordering can be counted when using sequences of words as alignment points by using `seq_cross`. Here, consecutive words are grouped together following given criteria so that crossing links can be counted on aligned groups of words rather than individual words. However these groups are not linguistically motivated. To ensure that the word groups are linguistically motivated, SACr provides a linguistic correction of the groups of `seq_cross`. An initial group of `seq_cross` is maintained if it is linguistically valid according to our criteria (each item in a group must express a child-parent relationship to another item in the group). If it is not valid, new SACr subgroups are created inside that invalid group. This means that a sentence can have the same number of `seq_cross` and SACr groups, or more SACr groups than `seq_cross` but never less.

Whereas SACr provides a way to quantify the reordering of phrase-like structures of a translation compared to its source text, counting the changes of the dependency labels of a source sentence after translation sheds light on linguistic differences of aligned words on the surface level.

4.3.3 Label Changes

An intuitive solution to syntactic equivalence is to assess how the dependency labels of translated words change from their aligned source text labels. To do so, we can simply count the alignment pairs where the source and target labels of an aligned word pair differ.

Formally, given a collection A of pairs of aligned source and target labels between a source sentence and its translation, the total number of label changes L is calculated as the number of alignment pairs in which the source label src is different from the target label tgt (Eq. 4.1)[7].

[7] Note that if a label, on either the source or target side, is aligned with multiple labels (one-to-many, many-to-one, many-to-many alignment), then all its alignments are counted separately.

$$L = \# \{(src, tgt) \in A : src \neq tgt\} \tag{4.1}$$

where:

A	the collection of pairs of aligned source and target labels
src	the source label of a pair
tgt	the target label of a pair

For an illustrative example, consider the following active source sentence in Ex. 16a, which has been translated into a passive construction (Ex. 16b), and t heir word alignment (Ex. 16c).

(16) a. I saw him
`nsubj root obj`

b. Hij werd door mij gezien
He was by me seen
`nsubj aux case obl root`

c. 0-2 0-3 1-1 1-4 2-0

The word alignments can be visualised as in Figure 4.9.

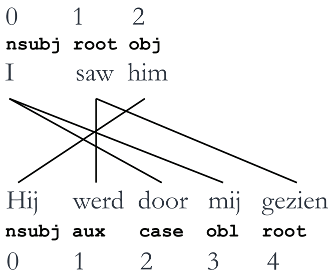


Figure 4.9. Word alignment visualisation of Ex. 16

When counting the label changes, we look at each source word and compare its label to the labels of the words that it is aligned to. To exemplify this, consider the label changes of Ex. 16 in Table 4.1, leading to a total number of four label changes. These label changes are then normalised by the total number of alignments, leading to a value of $^{4}/_{5} = 0.8$.

Source (label)	Target (label)	Change
“I” (`nsubj`)	“door” (`case`)	1
“I” (`nsubj`)	“mij” (`obl`)	1
“saw” (`root`)	“werd” (`aux`)	1
“saw” (`root`)	“gezien” (`root`)	0
“him” (`obj`)	“Hij” (`nsubj`)	1
		Total: 4 (normalised: $^4/_5$ = **0.8**)

Table 4.1. Label changes for Ex. 16

4.3.4 Aligned Syntactic Tree Edit Distance

Whereas SACr calculates a cross value on a shallow level (injected with a tree-based grouping) to quantify word order changes, it is also possible to determine deeper, structural differences between the source and target sentences. To compare the actual source and target dependency *structures*, we propose ASTrED.

As the name implies, aligned syntactic tree edit distance (ASTrED) incorporates a source dependency tree and a target dependency tree with the word alignments between the source and target sentence. The goal is to modify the labels of the source and target dependency tree so that the labels of aligned words are identical. By doing so, we can ensure that the tree edit distance between these modified trees takes word alignment information into account.

Consider the example sentence and its translation in Ex. 17 and its word alignment (visualised in Figure 4.10). This example will be used to explain ASTrED in the following subsections.

(17) a. Does he believe in love ?
`aux nsubj root case obl punct`

b. Gelooft hij in de liefde ?
Believes he in the love ?
`root nsubj case det obl punct`

c. 0-0 1-1 2-0 3-2 4-3 4-4 5-5

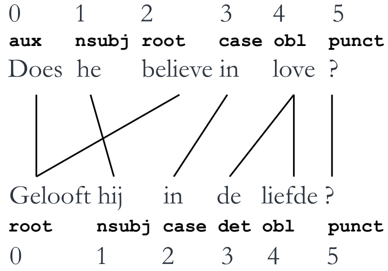

0 1 2 3 4 5
aux nsubj root case obl punct
Does he believe in love ?
Gelooft hij in de liefde ?
root nsubj case det obl punct
0 1 2 3 4 5

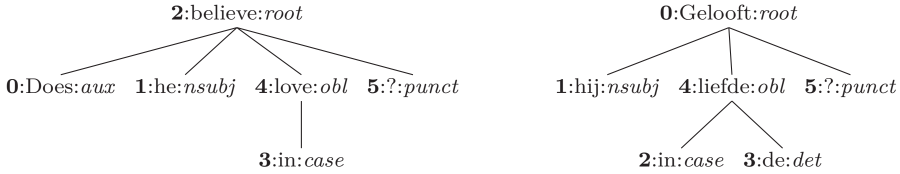
Figure 4.10. Word alignment visualisation of Ex. 17

The metric can be summarised in the following steps, on which we elaborate in the next subsections.

1. Parse the source and target sentences into dependency trees (using UD labels).
2. Find grouped tokens between source and target trees based on word alignment. A group is defined as the minimal group of tokens in the source and target sentences that are exclusively connected to each other through word alignment.
3. Modify the labels of the grouped tokens in their respective trees, so that the labels of tokens belonging to the same group get the same label. Nodes that were not aligned, and thus do not belong to any group, remain unchanged.
4. Calculate tree edit distance between the modified trees, which measures the structural difference between the aligned source and the target sentences. Normalize by the average number of source and target words.

4.3.4.1 Constructing dependency trees

Identical to the previous metrics, we use dependency trees to represent the source and target sentences in a linguistically meaningful way (see Sec. 4.3.1). As an example, let us take the previously mentioned example Ex. 17. The source and target sentence can each be represented as a dependency tree where each node is internally represented as the corresponding dependency label (Figure 4.11a, 4.11b).



(a) Source dependency tree of Ex. 17a: “Does he believe in love ?”

(b) Target dependency tree of Ex. 17b: “Gelooft hij in de liefde ?”

4.3.4.2 Merge grouped tokens and update labels

In order to measure the structural difference between a source and target sentence, we use tree edit distance. The tree edit distance between two trees is the minimal number of operations that are needed to change one tree into the other. The three possible operations are deleting, inserting, or substituting (also called "renaming") a node in the tree.[8] We cannot simply take the edit distance between the source and target dependency trees, however, because that would disregard the word alignment information. Tree edit distance in itself is unaware of which source nodes are supposed to align with which target nodes. To be able to calculate alignment-aware tree edit distance (the distance between the source and target dependency structure while also taking word alignment information into account), we modify the source and target trees by merging their labels with respect to the word alignments. Unaligned words remain untouched. In practice, that means that all tokens that are connected to each other through word alignment are grouped together. Here, they are represented (serialised) as a mapping of source label(s) to target label(s), where source labels are separated by a pipe (`|`) and their corresponding target labels by a comma.

More specifically, if we consider the example in 17, we can distinguish five groups (Example 18) where the corresponding words are given between brackets:

(18)
- `aux:root|root:root` (does:gelooft|believe:gelooft)
- `nsubj:nsubj` (he:hij)
- `case:case` (in:in)
- `obl:det,obl` (love:de,liefde)
- `punct:punct` (?:?)

4.3.4.3 Modify dependency trees

For all items involved in a group, their respective labels in their respective trees are updated to the serialised group. This implies that the nodes in the source and target trees that are aligned, now have the same label. This is important, because the goal is to calculate tree edit distance on the *aligned* source and target trees.

The trees with modified labels are shown in Figures 4.12 and 4.13 with a word's original position (index) placed before the serialised label. Note how the labels are now modified so that aligned nodes share the same label. Also consider that if, for instance, two source nodes are aligned with one target node, then all three will share the same modified label, such as the label `aux:root|root:root` which is the alignment of "does ... believe" to "Gelooft".

[8] To automate the tree edit distance calculation, we use a Python implementation (`https://github.com/JoaoFelipe/apted`) of the APTED algorithm (Pawlik & Augsten, 2015, 2016).

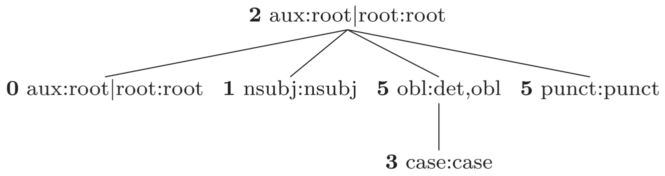


Figure 4.12. Modified source dependency tree of Example 17a: "Does he believe in love ?"

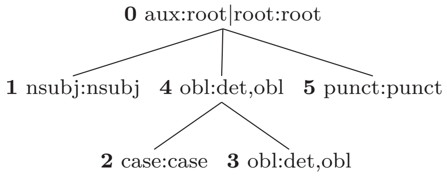


Figure 4.13. Modified target dependency tree of Example 17b: "Gelooft hij in de liefde ?"

4.3.4.4 Calculate tree edit distance

Finally, we calculate the tree edit distance between the modified trees shown above. To change the modified source tree in Figure 4.12 to the modified target tree in Figure 4.13, two operations are needed, as visualised in Figure 4.14:

1. the source node `aux:root|root:root` (orange, solid line) must be deleted;
2. the target node `obl:det,obl` (blue, dashed line) must be inserted.

The ASTrED score is normalised by the average number of source and target words. This is different from the way that SACr and the label changes are normalised: SACr is normalised by the number of alignment links between SACr groups because the crossing links originate from those alignments. Label changes are normalised by the number of word alignment link, because the differences in labels are calculated between aligned labels. ASTrED is calculated between tree representations of the source and target sentence, which means that each word's label in the source or target text is a node in the dependency tree. In other words: ASTrED takes unaligned words (null alignment) into account (see Sec. 4.4.2 for an example), whereas SACr and label changes only consider the alignments themselves. Therefore, ASTrED is normalised by the average number of source and target words. Applying that to this example, with source sentence of six words and a target sentence of six words, we get an ASTrED score of $^{2}/_{6} = 0.33$.

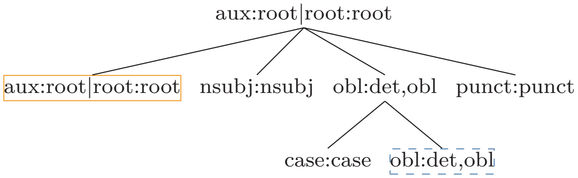


Figure 4.14. A visualisation of the two needed edits to go from modified source tree in Figure 4.12 to the modified target tree in Figure 4.13. The orange solid box indicates the source node that needs to be deleted and the dashed blue box highlights the target node that needs to inserted

To reiterate: we calculate tree edit distance on the modified trees where node labels are replaced by a serialised representation of the aligned source and target nodes. This is done to ensure that tree edit distance takes word alignment information into account.

4.3.5 Metrics Overview

Metric	**Captures**	**Normalisation by**
Label changes	changes in dependency labels in the surface form based on word alignment	# alignments
SACr	reordering of linguistically motivated groups by measuring crossing links	# alignments
ASTrED	structural difference between the source and target dependency tree while also taking word alignment into account	avg. # source and target words

Table 4.2. An overview of the metrics introduced in this paper

4.4 Discussion With Examples

As discussed before, syntactic equivalence is an ill-defined concept because it entails different linguistic aspects: from word reordering at the surface level to deep structural differences. For that reason we proposed three linguistically motivated metrics (that can be used and calculated independently) that all tackle a different part of the problem. In this section we will discuss further what the differences between the metrics are by going over two examples that illustrate other typical linguistic differences between English and Dutch, in addition to the previously given examples (active-passive, indirect speech, English *do*). In the following two examples we discuss subject-verb word order and the future tense, and the translation of the English gerund to Dutch and null alignments.

4.4.1 Subject-verb Word Order and the Future Tense

English is typically classified as a language with subject-verb-object (SVO) word order, but there is no consensus on Dutch. One approach suggests that Dutch uses the subject-object-verb (SOV) with V2, verb-second, word order (Koster, 1975), where in the main clause, the finite verb must be placed second with one constituent preceding it, and where subordinate clauses adhere to the SOV word order. Alternatively, Zwart (1994) suggested that Dutch is SVO, by dissecting the verb phrase (VP) structure of a subordinate clause in detail.

Even though that discussion exceeds the scope of this paper, the practical implication is that in many cases (e.g. topicalisation, left dislocation, subordinate clauses), the word order of English and Dutch differs.

Consider Ex. 19 where the word order of the main verb and the subject differs between Dutch and English because of the dislocated adverb, which leads to inversion in Dutch. The example also shows how the simple future tense can be presented in the present tense in Dutch, which leads to the source auxiliary "will" and its root "go" to be aligned with the present tense root "ga".

(19) a. Tomorrow I will go home .
`advmod nsubj aux root obj punct`

b. Morgen ga ik naar huis .
Tomorrow go I to home .
`advmod root nsubj case obl punct`

c. 0-0 1-2 2-1 3-1 4-3 4-4 5-5

The alignments and word crosses can be visualised as follows in Figure 4.15. The `word_cross` value is $^{2}/_{7} = 0.29$.

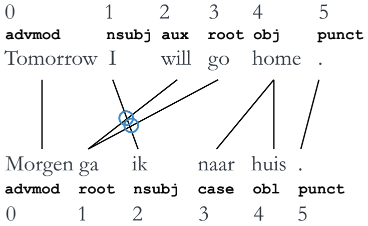

0 1 2 3 4 5
advmod nsubj aux root obj punct
Tomorrow I will go home .
Morgen ga ik naar huis .
advmod root nsubj case obl punct
0 1 2 3 4 5

Figure 4.15. Visualisation of word alignment of Ex. 19. And a `word_cross` value of $^{2}/_{7} = 0.29$

Vanroy, Tezcan, and Macken (2019) suggested a sequential approach to word reordering where consecutive words are grouped together following a given set of criteria (cf. Sec.4.2.3). In the example above, this can be visualised as in Figure 4.16, showing a `seq_cross` value of $^{1}/_{4} = 0.25$.

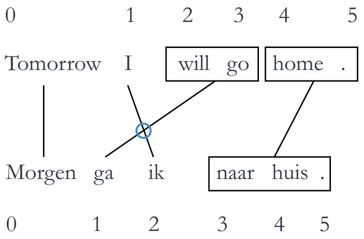


Figure 4.16. `seq_cross` representation of Ex. 19 with a value of 1/4 = 0.25

In this book chapter, we have proposed an improved version of `seq_cross` named SACr. Whereas `seq_cross` is not aware of linguistic information and naively groups word sequences together, SACr ensures that these groups are linguistically motivated: all items in a SACr group must exhibit a child-parent relationship to at least one other word in the group. The valid and invalid groups are shown for both the source and target dependency trees in Figures 4.17 and 4.18.

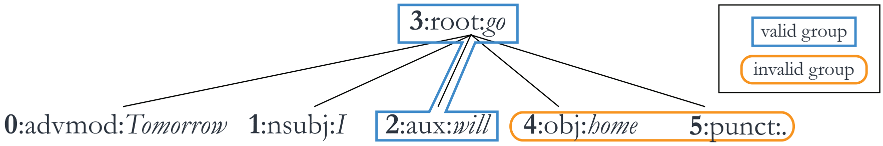


Figure 4.17. Source dependency tree of Ex. 19, highlighting valid and invalid groups

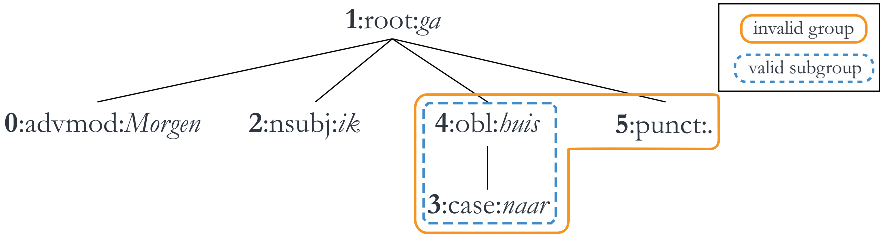


Figure 4.18. Target dependency tree of Ex. 19, highlighting an invalid group and a valid SACr subgroup

The initial groups of `seq_cross` are not linguistically motivated but by means of the dependency trees, we can correct these groups to ensure that all groups are indeed linguistically valid. The alignment between these groups can be used to quantify the reordering of syntactic word groups. In this example, there is one crossing link which is then normalised by the total number of alignments (five). The SACr value, then, is $^1/_5 = 0.2$.

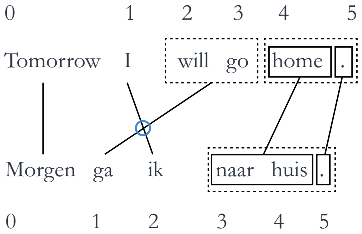


Figure 4.19. SACr representation of Ex. 19 with a value of $^1/_5 = 0.2$. Dotted boxes indicates the groups of `seq_cross`, which, when required, are split up into linguistically motivated SACr groups (solid boxes)

In addition to word reordering, the label changes are indicative of diverging linguistic properties. Looking at the label changes going from the source to the target sentence in Figure 4.15, we find three alignments where the labels of the source word have changed (Table 4.3), which when normalised gives a value of $^3/_6 =$ **0.5**.

Source (label)	**Target (label)**	**Change**
"Tomorrow" (`advmod`)	"Morgen" (`advmod`)	0
"will" (`aux`)	"ga" (`root`)	1
"go" (`root`)	"ga" (`root`)	0
"home" (`obj`)	"naar" (`case`)	1
"home" (`obj`)	"huis" (`obl`)	1
"." (`punct`)	"." (`punct`)	0
	Total: 3 (normalised: $^3/_6 =$ **0.5**)	

Table 4.3. Label changes for Ex. 19

With ASTrED, we also provide a means to compare the underlying structure of aligned dependency trees. This is done by grouping aligned words together in the source and target tree, changing their labels according to this grouping in both trees, and calculating tree edit distance between the modified trees. In Ex. 19, we can distinguish five groups (Ex. 20).

(20)
- `advmod:advmod` (Tomorrow:Morgen)
- `nsubj:nsubj` (I:ik)
- `aux:root|root:root` (will:ga|go:ga)
- `obj:case,obl` (home:naar,huis)
- `punct:punct` (.:.)

We can then modify the original dependency trees (see Figures 4.17 and 4.18) by changing the label of each node to the serialised group that it belongs to. The modified trees are given in:

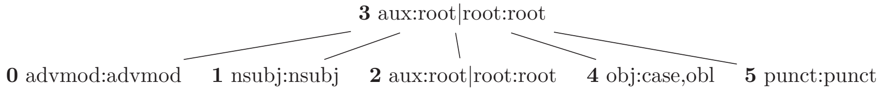


Figure 4.20. Modified source dependency tree of Ex. 19: "Tomorrow I will go home ."

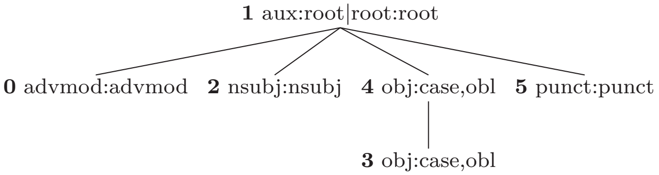


Figure 4.21. Modified target dependency tree of Ex. 19: "Morgen ga ik naar huis ."

These modified trees can then finally be used to calculate tree edit distance. Figure 4.22 shows the two edit operations that are needed to change the modified source tree to the modified target tree. This value is normalised with the average number of source (six) and target words (six), which leads to a ASTrED score of $^2/_6 = 0.33$.

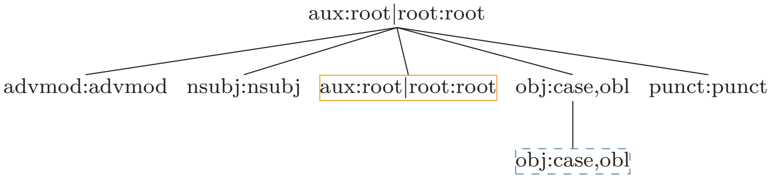


Figure 4.22. A visualisation of the two needed edits to go from the modified source tree in Figure 4.20 to the modified target tree in Figure 4.21. The orange solid box indicates the source node that needs to be deleted and the dashed blue box highlights the target node that needs to inserted

In this example, which involves a different subject-verb order in English and Dutch, SACr clearly models how the word order of the verb with respect to the subject has changed (Figure 4.19). Label changes, on the other hand, do not catch the word group reordering aspect because they solely compares aligned words, disregarding their position relative to each other. In this example, it does catch how the auxiliary verb "will" has a different label than the present tense of its Dutch translation "ga" (root). It also finds that whereas English allows for a “go `obj`” construction, Dutch requires a case marker in such case, in the form of “ga `case obl`”.

The edit operations of ASTrED (e.g. Figure 4.22) highlight that tree edit distance does not account for word reordering in some cases. That is due to the nature of dependency trees: even though our implementation of a dependency tree ensures that the order of *sibling* nodes is identical to their word order, there is no way in the tree to know the word order position of a parent node vis-à-vis its children. So two tree structures may be identical, but the word order of a parent node with respect to its descendants can still differ. In this case, the subtree structure of the subjects (“I” and “ik”) and their main verb (“go” and “ga”) are identical (it is a child-parent relationship), so the tree edit distance for that subtree is 0, even though the word order of the source and target sentence are different: in the English sentence the subject precedes the verb, whereas in the Dutch translation the verb comes first. That order difference is not visible in the trees. As such, it is clear that the reordering metrics capture different information than ASTrED. In this case, ASTrED catches the same differences that the label changes find, concerning the future tense that is translated as a present tense, and the English object following “go” that needs to be case-marked in Dutch. As a consequence, the node of the future auxiliary verb (`aux:root|root:root`) needs to be removed from the English source, and the case marker of the Dutch translation must be added (`obj:case,obl`), to arrive at the same tree structure (see Figure 4.22). The results of all metrics for this example are summarised in Table 4.4.

Metric	Value
`word_cross`	0.29
`seq_cross`	0.25
`SACr`	0.2
Label changes	0.5
`ASTrED`	0.34

Table 4.4. Summary of the results of all metrics for Ex. 19 (rounded to two decimals)

4.4.2 English Gerund, Verb Order, and Null Alignment

In English, gerunds are verb forms that typically end with `-ing` and that most often take a nominal function. In Dutch, however, this construction is frequently translated as an infinitive, but just as often a complete rewrite of the original constituent seems appropriate. In the following example an English gerund (“Shouting”) is translated as an infinitive (“roepen”). Both their dependency relations to their root are `csubj`, meaning that they are clausal subjects, i.e. they are the subject of a clause and they are themselves a clause. Similar to the previous example, the word order of the object (“for help" and "om hulp") with respect to its verb ("Shouting" and "roepen") is a noteworthy difference in the source and target sentence. Finally, in this example, "seemed" is translated by adding a pronoun as an object[9] to the verb "leek" *seemed*, namely "mij" *to me*. Because of this explicitation, "mij" cannot be aligned with a source word.

(21) a. Shouting for help seemed appropriate .
`csubj case obl root xcomp punct`

b. Om hulp roepen leek mij gepast .
For help call seemed me appropriate .
`case obl csubj root obj xcomp punct`

c. 0-2 1-0 2-1 3-3 4-5 5-6

The alignments in Example 21c can be visualised in Figure 4.23, which also shows the crossing links on the word level. In this case, there are two crossing links that indicate the different word order of objects relative to their verb in English compared to Dutch, as discussed before. After normalisation, the `word_cross` value is $^2/_6 = 0.33$.

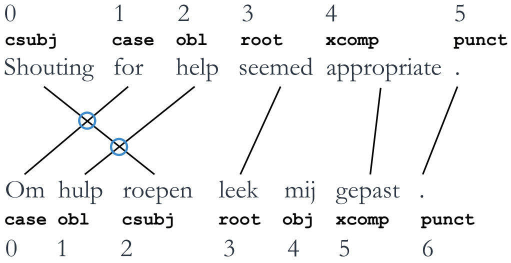


Figure 4.23. Visualisation of word alignment in Ex. 21. And a `word_cross` value of $^2/_6 = 0.33$

When grouping consecutive words, as discussed in Section 4.2.3, we find that "for help" and "Om hulp" each constitute a group, as well as "appropriate ." and "gepast .". This is visualised in Figure 4.24. Grouping "for help" and "Om hulp" leads to a reduction in crossing links: now there is only one crossing. The `seq_cross` value is $^1/_4 = 0.25$.

[9] Following the conventions of UD, we label "mij" as an `obj`. The annotation guidelines suggest that when a verb has only one object, it should be labeled as an `obj` and not an `iobj`, regardless of the morphological case or semantic role of that word. (See `https://universaldependencies.org/u/dep/iobj.html`)

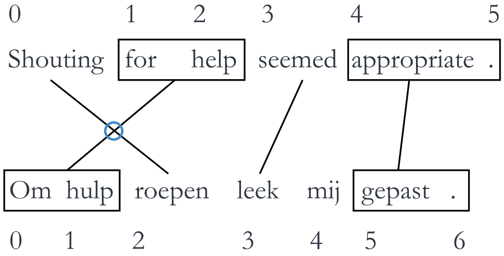

0 1 2 3 4 5
Shouting for help seemed appropriate .
Om hulp roepen leek mij gepast .
0 1 2 3 4 5 6

Figure 4.24. `seq_cross` representation of Ex. 21 with a value of $1/4 = 0.25$

However, as discussed in Section 4.3.2, the groups of `seq_cross` are not linguistically motivated. To create groups that take the linguistic structure into account, we verify that all items in a group share a child-parent relationship with another word in that group. For this example, we can investigate the source and target dependency trees in Figures 4.25 and 4.26 respectively.

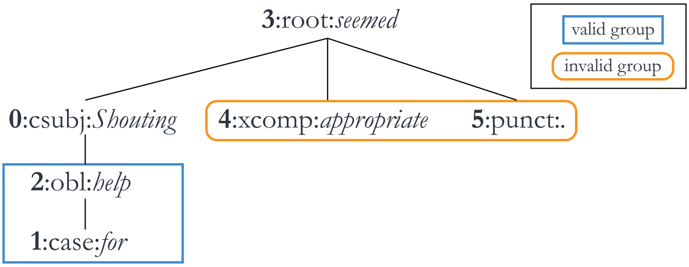


Figure 4.25. Source dependency tree of Ex. 21, highlighting an invalid group and a valid SACr subgroup

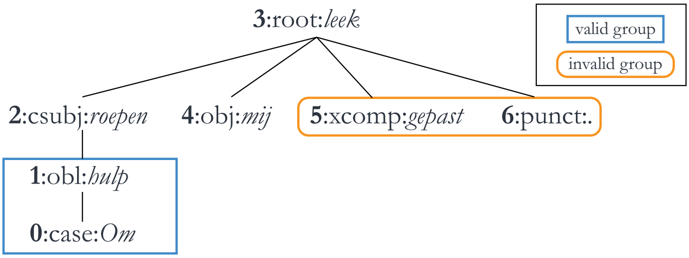


Figure 4.26. Target dependency tree of Ex. 21, highlighting an invalid group and a valid SACr subgroup

The visualisations of the dependency trees make clear that the groups “for help” and “Om hulp” are valid because the prepositions (“for” and “om” respectively) are children of their root (“help” and “hulp”, resp.) and child-parent relationships constitute a valid SACr group. The other groups “appropriate .” and “gepast .” are not valid because the two words in each groups share a sibling relationship rather than a child-parent relationship, which is not sufficient to form a valid SACr group. These linguistically corrected groups have been visualised in Figure 4.27. The number of crossing links is still one, but because the invalid groups are corrected (“appropriate .” and “gepast .”), the normalised value has now changed from `seq_cross` 0.25 to SACr 0.2.

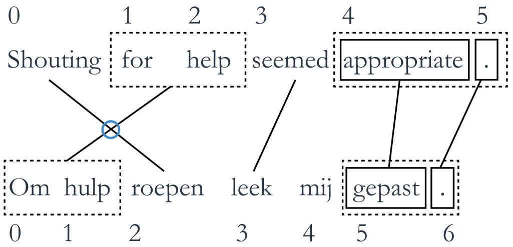


Figure 4.27. SACr representation of Ex. 21 with a value of $^1/_5 = 0.2$. Dotted boxes indicates the groups of `seq_cross`, which, when required, are split up into linguistically motivated SACr groups (solid boxes)

The label changes in this example are quite self-explanatory: looking at the word alignments in Figure 4.23 it is evident that all the labels of aligned words are identical on the source and target side. Therefore there are zero label changes in this example. Nevertheless, that does not mean that are no structural difference, as ASTrED will illustrate.

To calculate ASTrED, first the labels of the source and target trees need to be grouped according to the word alignments. Each group should contain all the labels of words that are connected to each other through word alignment. In Example 22, we can find six groups and also one unaligned word (“mij” *me*).

(22)
- `csubj:csubj` (Shouting:roepen)
- `case:case` (for:Om)
- `obl:obl` (help:hulp)
- `root:root` (seemed:leek)
- `xcomp:xcomp` (appropriate:gepast)
- `punct:punct` (.:.)
- **null alignment** (in target): `obj` (mij)

As a next step, the labels of each node in a group must be updated to the serialised group’s label. In this example, the groups always consist of only one source and one target item. The unaligned `obj` node in the target sentence is still present after changing the labels (Figures 4.28 and 4.29).

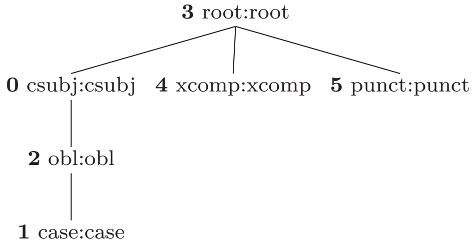


Figure 4.28. Modified source dependency tree of Ex. 21: “Shouting for help seemed appropriate .”

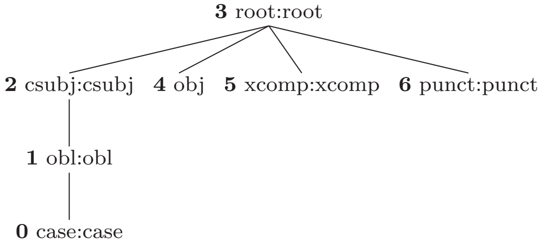


Figure 4.29. Modified target dependency tree of Ex. 21: “Om hulp roepen leek me gepast .” Note the unalgined `obj` node

Now, the tree edit distance between these modified trees can be calculated. The structure of the source sentence is in fact exactly the same as the one in the target sentence, with the exception of one unaligned `obj` node (“mij”). The only operation that is needed to change the source structure to the target structure is inserting the unaligned target node (Figure 4.30). This illustrates that ASTrED is the only one of the tree metrics that is able to take into account null alignments. The edit operations are normalised by the average number of source (6) and target (7) tokens, so the ASTrED value is $^1/_{6.5} = 0.15$.

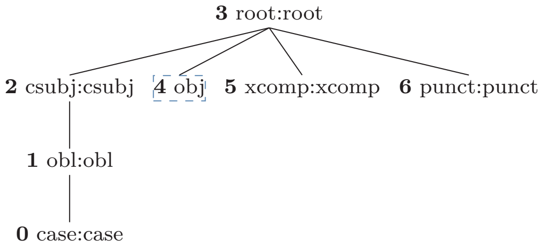


Figure 4.30. A visualisation of the edit (insertion, the dashed blue box) to go from the modified source tree in Figure 4.28 to the modified target tree in Figure 4.29

In this example, it became clear how SACr again accurately quantifies the reordering of linguistically motivated word groups. In particular it showed how the subject-verb order of English and Dutch can be quantified with a single crossing link because of the syntactically aware word grouping of "for help" and "Om hulp". Because the examples were quite closely related in this example, we did not observe any label changes. However, on a deeper structural level we found that the structure of both sentence does differ slightly because of a null alignment on the target side: "mij" *me* was inserted in the translation even though there is no source word to align it with. The results are summarised in Table 4.5.

Metric	**Value**
`word_cross`	0.34
`seq_cross`	0.25
`SACr`	0.2
Label changes	0.0
`ASTrED`	0.15

Table 4.5. Summary of the results of all metrics for Ex. 21 (rounded to two decimals)

Generally speaking, the three metrics model three different things: SACr specifically quantifies the reordering of linguistically inspired word groups. When the surface word order of languages differs in specific structures, SACr catches up on that. This is particularly evident in Example 4.6 where a different word order is found twice in the same sentence ("Sometimes **she asks** me why I **used to call her father Harold** ." vs. "Soms **vraagt ze** waarom ik **haar vader Harold noemde** ."). Also based on the surface forms, label changes compare the labels of the aligned words on the source and target side. By doing so, it can quickly become evident when a source sentence and its translation have been translated completely differently (think, for instance, about the active-passive example in Example 16 where a `nsubj` became an obj). ASTrED serves a similar function but it compares the actual tree structures of the source and target sentence while at the same time also taking the word alignments into account. Whereas SACr and label changes work on the surface forms, ASTrED does a deeper linguistic comparison between a source sentence and its translation, as the last example clearly shows.

4.5 Proof of Concept

To investigate how syntactic differences between a source text and its translation relate to difficulty, we can measure the effect that our syntactic measures have on translation process features that may be indicative of cognitive effort, which in turn points to translation difficulty (also see our previous research for details and a literature overview concerning cognitive effort and translation; Vanroy, De Clercq, & Macken, 2019)[10]. We built mixed-effect models in R (R Core Team, 2019), using the lme4 package (Bates et al., 2015) with lmerTest (Kuznetsova et al., 2017) to obtain p-values and perform automatic backward elimination of effects.

We used part of the ROBOT dataset (Daems, 2016) for this analysis. The full ROBOT dataset contains translation process data of ten student translators and twelve professional translators working from English into Dutch. Each participant translated eight texts, four by means of post-editing (starting from MT output), and four as a human translation task (starting from scratch). Task and text order effects were reduced by using a balanced Latin square design. The texts were newspaper articles of 150-160 words in length, with an average sentence length between 15 and 20 words. As the goal of the original ROBOT study was to compare the differences between post-editing and manual translation, the texts were selected to be as comparable to one another as possible, based on complexity and readability scores, word frequency, number of proper nouns, and MT quality. For the present study, however, only the process data for the human translation task was used. This dataset was manually sentence and word aligned. Dependency labelling was done automatically by using the aforementioned stanza parser (Qi et al., 2020).

We followed exclusion criteria suggested by Bangalore et al. (2015) before analysing our data: exclude cases where two ST (source text) segments were fused into one, exclude the first segment of each text, exclude segments with average normalised total reading time values below 200ms (total reading time; the time (in ms) that participants have their eyes fixated on the source or target side, measured by eye tracking) and exclude data points differing by 2.5 standard deviations or more from the mean. After filtering, the dataset consists of 537 data points, i.e. translated segments. All plots were made

[10] Other chapters in this volume also discuss new advances in cognitive effort research. See for instance the work by Huang and Carl in Chapter 2, and Chapter 3 by Cumbreño and Aranberri regarding cognitive effort during post-editing, and Lacruz et al. on cognitive effort in JA-EN and JA-ES translation (Chapter 11).

using the effects package (Fox & Weisberg, 2019). In parallel with Bangalore et al. (2015), dependent variables from the TPR-DB (Carl et al., 2016) were chosen, specifically total reading time on the target (TrtT) and source (TrtS) side, and duration of coherent typing behavior (total duration of coherent keyboard activity excluding keystroke pauses of more than five seconds; Kdur), normalised by the number of words per segment and centred around the grand mean (hence the negative values in the graphs).[11] The predictor variables were our three proposed metrics: SACr, label changes, and ASTrED. In the full model, all three variables were included with interaction. We performed backward elimination of effects to build the best model for each dependent variable. Participant codes and item codes were included as random effects.

For coherent typing behavior (Kdur), the only predictor variable that was retained in the best performing model was the number of label changes. An increase in label changes had a highly significant ($p < 0.001$) positive effect on Kdur (estimate = 969.1, SE = 232, t = 4.18). This effect can be seen in Figure 4.31. This indicates that translators needed more time to translate those source segments that required more label changes when translating.

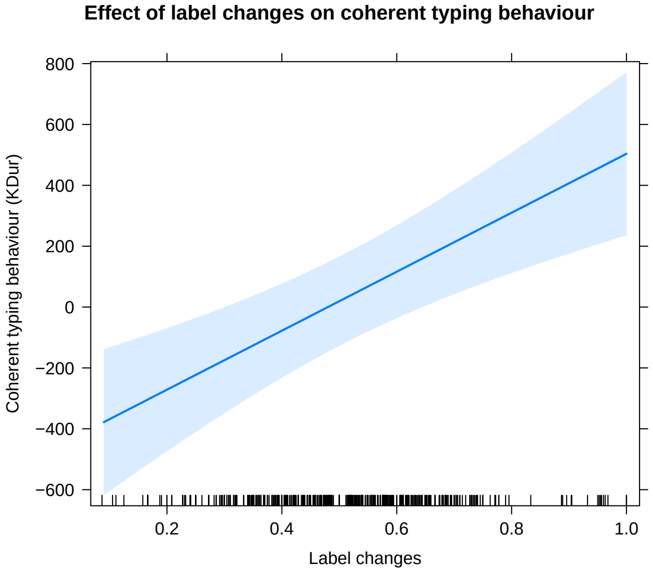


Figure 4.31. Effect plot for the main effect of label changes on coherent typing behaviour

Source reading time (TrtS) was best predicted by SACr only, although the model which included both participants and items as random effects gave

[11] Even though our experimental set-up is similar, our results cannot be compared to those of Bangalore et al. (2015) because we use a different data set, and do not use entropy but absolute values per-segment.

rise to convergence warnings. The main effect of SACr on TrtS was positive (estimate = 69.82, SE = 28.39, t = 2.46) and significant (p = 0.01). The effect can be seen in Figure 4.32. The model without participants as random effect did converge and showed a similar main effect (estimate = 95.11, SE = 33.85, t = 2.81, p = 0.005). This means that those segments that were translated by moving more word groups or move word groups further away required more reading time on the source side.

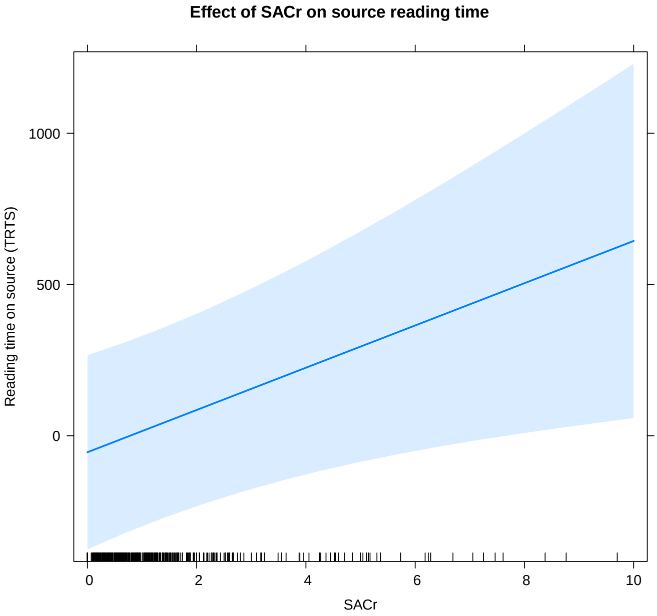


Figure 4.32. Effect plot for the main effect of SACr on source text reading time

Target reading time (TrtT), on the other hand, was best predicted by a combination of all three predictor variables with interaction. The three-way interaction effect was significant (estimate = 3383.2, SE = 1173.6, t = 2.88, p = 0.004). All effects included in the model are summarised in Table 4.6. The interaction effect is visualised in Figure 4.33. The figure shows the effect of ASTrED values on target reading time, given a certain SACr value and number of label changes. Only the minimum and maximum values of SACr and label changes are included as reference points (0 and 9.7 for SACr, and 0.09 and 1 for label changes, respectively). What this indicates, is that, if SACr is low, an increase of ASTrED or an increase in the number of label changes does not really have that much of an impact on target reading time. However, if SACr values are high and there is a low number of label changes, target reading time goes down for higher ASTrED values; whereas target reading time goes up for higher ASTrED values when SACr values are high and there is a high number of label changes. Looking at the graph on the right (high SACr value), it would seem that when a lot of word group reordering is required without many label changes (blue line with negative slope), structurally similar source and target sentences (low ASTrED) lead to a higher TrtT. Conversely, when a lot of word group reordering is needed alongside many label changes (orange line with positive slope), dissimilar syntactic structures (high ASTrED) positively affect the time that translators read the target text. This conclusion should be taken with a grain of salt, though, and additional experiments with other data sets are required to draw more certain conclusions.

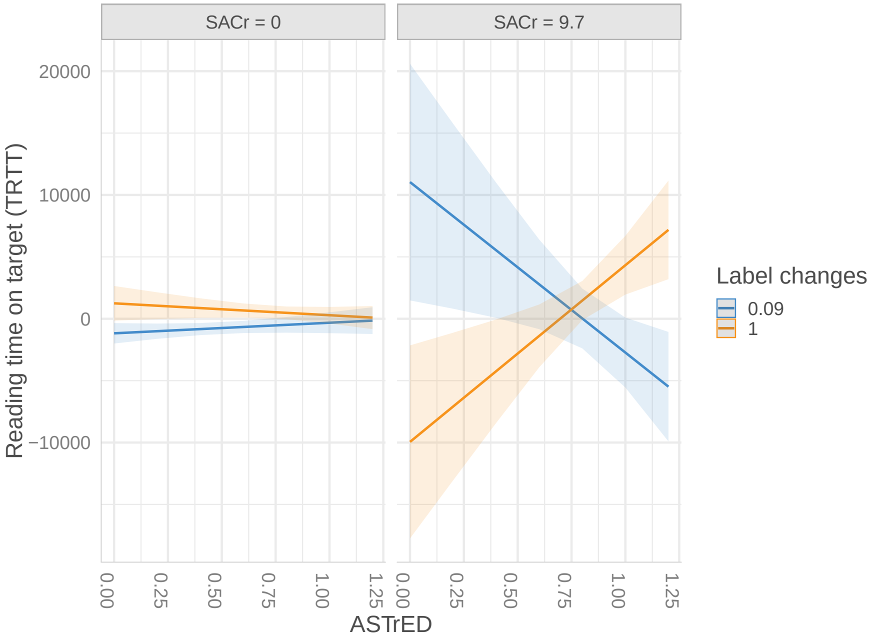


Figure 4.33. Effect plot for the three-way interaction effect of ASTrED, label changes, and SACr on target reading time

Fixed effect	Estimate	SE	t	p
ASTrED	1034.4	819.1	1.26	.207
label changes	2662.5	1103	2.41	.016*
SACr	1498.3	602.3	2.49	.013*
ASTrED : label changes	−1994.7	1514.1	−1.32	.188
ASTrED : SACr	−1812.6	692.3	−2.62	.009**
label changes : SACr	−2652.4	989.5	−2.68	.008**
ASTrED : label changes : SACr	3383.2	1173.6	2.88	.004**

$^{*}p < .05$
$^{**}p < .01$

Table 4.6. Effect summary of three-way interaction effect between ASTrED, label changes, and SACr on target reading time

Unsurprisingly, the metrics are only weakly to moderately correlated, as seen in Table 4.7. This is likely due to a single common factor of all metrics: they are, at their core, all based on the same dependency labels. Different dependency trees lead to different SACr groups, a change in the merged ASTrED trees, as well as the label changes themselves. However, because each metric uses the dependency labels in its own way, a change in dependency structures affect specific metrics differently. The metrics are therefore mildly correlated but they have a different effect on the translation process, as shown above.

	ASTrED	Label changes
ASTrED		
Label changes	.41*	
SACr	.40*	.35*

$^{*}p < .01$

Table 4.7. Kendall correlation between normalised metrics: ASTrED, label changes, and SACr

In this section we have calculated the effect of our proposed syntactic metrics on translation process features to show that our interpretation of syntactic equivalence has an effect on the translation process. Even though our dataset was rather small, and more elaborate experiments are needed, these findings already confirm that, as the literature indicates (cf. Section 4.2), (syntactic) equivalence does affect some translation process features such as reading time and typing duration, which serve as a proxy for the translation difficulty. Generally speaking, this experiment arrives to the same conclusion as Bangalore et al. (2015), namely that syntactically diverging source and target segments impose difficulty on the translator. In addition, this experiment also confirms that all three metrics seem to affect the translation process differently, which motivates further research into this topic.

4.6 Conclusion and Future Work

In this work, we have introduced three new metrics to measure syntactic equivalence between a sentence and its translation. The three metrics serve different purposes, which is also revealed in Section 4.5. Keeping track of dependency label changes is an intuitive approach to see how the relation of each word to its root has changed in the translation. Syntactically aware cross (SACr) offers a linguistically motivated method to calculate word group reordering. Finally, aligned syntactic tree edit distance (ASTrED) compares the deep linguistic structure of the source and target sentence while taking word alignment into account. We open-source the implementation of the metrics as a Python package.

Broadly speaking, we are interested in ways to quantify translation difficulty. Syntactic equivalence is one part of that, as we have discussed in previous research (Vanroy, De Clercq, & Macken, 2019; Vanroy, Tezcan, & Macken, 2019). In future work we want to investigate whether we can distil typical word group reordering patterns, label changes, or structural divergence and categorize them into Catford's obligatory and optional shifts (Catford, 1965). The hypothesis is that in language pair specific contexts, some word group orders, labels, and structures are simply incompatible between two languages, in which case the translator is forced to make an obligatory shift and cannot rely on a literal translation. In addition, we want to perform more analyses using our metrics and compare them to translation process data. As a proof-of-concept, we presented one such analysis in Section 4.5, but since the used dataset is relatively small, similar experiments should be done to confirm, and expand on, these results. Moreover, we intend to run equivalent experiments on different language pairs to investigate (the difficulties between) syntactically divergent languages.

Finally, rather than calculating syntactic entropy based on the features Valency, Voice, and Clause type (Bangalore et al., 2015), we are interested in investigating the feasibility of calculating syntactic entropy based on our metrics. Syntactic entropy can be simplified as the agreement between the translators of the same source text with respect to the syntax of their translations. Put differently, how similar or divergent in syntax are the different translations of the translators? Because our proposed metrics aim to quantify syntactic equivalence between a source sentence and its translation, they are good candidates to be used in an entropy setting to see how well translators agree on structural or syntactic changes when translating. This information, in turn, can be used in modelling the translatability of specific linguistic phenomena.

Chapter 5

Comparing the Effect of Product-based Metrics on the Translation Process[1]

Bram Vanroy* · **Moritz Schaeffer+** · **Lieve Macken***
*LT^3, Language and Translation Technology Team, Ghent University
+TRA&CO, Center for Translation and Cognition, Johannes Gutenberg University Mainz

Abstract
Characteristics of the translation product are often used in translation process research as predictors for cognitive load, and by extension translation difficulty. In the last decade, user-activity information such as eye-tracking data has been increasingly employed as an experimental tool for that purpose. In this paper, we take a similar approach. We look for significant effects that different predictors may have on three different eye-tracking measures: First Fixation Duration, Eye-Key Span, and Total Reading Time on source words. As predictors we make use of a set of established metrics involving (lexico)semantics and word order, while also investigating the effect of more recent ones concerning syntax, semantics or both. Our results show a, particularly late, positive effect of many of the proposed predictors, suggesting that both fine-grained metrics of syntactic phenomena (such as word reordering) as well as coarse-grained ones (encapsulating both syntactic and semantic information) contribute to translation difficulties.

Keywords: translation studies · translation difficulty · eye tracking · syntax · entropy · translation process

Publication: Vanroy, B., Schaeffer, M., & Macken, L. (2021). *Comparing the effect of product-based metrics on the translation process.* [Manuscript submitted for publication].

[1] This article is currently under review and is subject to change following comments from the reviewers.

5.1 Introduction

Translation difficulty prediction, which aims to assess the difficulty of a translation task, is a topic of interest within translation studies that can benefit both pedagogical and research settings. Advances in translatability could for instance ensure that appropriate text material is used in translation classes, and to create general-purpose machine translation (MT) systems that are trained on a balanced mix of simple and hard texts. On the other hand, it could also help the research fields of translation studies and psycholinguistics to select source material of suitable translation difficulty for experiments. Even though a well-established methodology to quantify a source text's translatability does not exist (yet), the problem of translation difficulty has gained some attention over the years.

The PreDicT project[2] (Predicting Difficulty in Translation) aims to contribute to the field of translatability by investigating source text language features that add to a text's translation difficulty. As described above, the application of advances in this field could be to predict the translation difficulty of a source text, or parts of it, without having access to a translation. That would allow users to automatically rate a text or highlight its difficulties without the need of translating it beforehand. The PreDicT project has particularly focused on syntactic equivalence and divergence between a source text and its translation. In previous work (Vanroy, Tezcan, & Macken, 2019), two metrics were introduced to calculate the word and word group movement on the sentence level. In addition, a machine learning system was built that could predict these word and word group reordering values by only using source text information with a moderate Pearson r correlation. Additional sentence-level metrics were introduced in Vanroy et al. (in press). In the current paper, however, we take a more fine-grained approach and make these metrics available on the word level so that meaningful translation process analyses can be done to investigate their impact on the translation task.

We examine the effect of a number of predictor variables (Sec. 5.3.3) on translation process data as a proxy for cognitive effort and, hence, difficulty, as is usual in translation process research (Muñoz Martín, 2012). We include metrics that are intended to measure syntactic or (lexico)semantic (dis)similarities between a source text (ST) and its target text (TT), or both. Some metrics require multiple translations (and are entropy-based), and others can be calculated on single translations. The unit of interest is the word, but some of the metrics are calculated with word group information in mind. The current research can thus serve as a peek into the effects that such different metrics have on process data. We test their effect on three different eye-tracking measures on the source tokens (Sec. 5.3.2), both early (first fixation duration) and late (eye-key span, total reading time).

This paper is structured as follows. First an overview of related research

[2] `https://research.flw.ugent.be/nl/projects/predict`

regarding literal translation and the relationship between ST and TT is discussed. Then, the experimental set-up is described in Section 5.3, with specific attention for the data and model description. The predictor variables are discussed extensively as well. Section 5.4 reports the results, which are discussed in detail in the discussion (Sec. 5.5). Finally, we end with broad conclusions and suggestions for future research (Sec. 5.6).

5.2 Related Research

A lot of work has been done on the relationship between ST and TT, particularly on the concept of literal translation and the (formal) transfer of the source text to the target. We will discuss one specific way how literal translation can be operationalised (Schaeffer & Carl, 2014), which leads us to different ways of how the relationship between a source and target text can be measured. Finally, research concerning the unit of translation is described, as it relates to our decision to include predictors that are calculated based on word as well as on word group information.

5.2.1 Literal Translation

“Literal translation” is often contrasted with its “rival” free translation and yet a single definition is not available (Shuttleworth & Cowie, 2014, p. 95-97). The concept has been used in different ways to mean different things (see Halverson, 2015, for an extensive overview of varying interpretations). For instance, some consider literal translation ungrammatical and outside the acceptable norm depending on the genre. In such a view, literal translation is considered as nothing more than what Seleskovitch (1976) calls code switching, the technical conversion of one symbol to another. Others restrict literal translation to mean word-for-word translation that leads to a necessarily “grammatically and idiomatically correct TL text” (Vinay & Darbelnet, 1995, p. 33), or go even so far that the only requirement for literal translation is that the translation is “structurally and semantically *modelled upon* the ST fragment while respecting TL grammatical constraints” (Englund Dimitrova, 2005, p. 53; our emphasis).

Abstracting away from the discussion above, and without defining literal translation itself, Chesterman (2011, p. 26) refers to the *literal translation hypothesis* that states that “during the translation process, translators tend to proceed from more literal versions to less literal ones”. He does not make any claims about what the starting point is nor about what a “most” and “least” literal translation would look like. The literal translation hypothesis simply states that initially formal features of the source text have a large effect on the (perhaps mental or “interim”) translation that is being produced and that this effect decreases over the duration of the translation process. The literal translation hypothesis has received supporting evidence from translation process studies that measure the effects of literality metrics (see below) on process data (e.g. Bangalore et al., 2015, 2016; Schaeffer, Dragsted, et al., 2016). Such experiments show that the translation procedure starts from a more literal translation, but when this is not possible due to the constraints of TL or other contextual or extralinguistic factors, non-literality must inevitably increase, which - the experiments show - goes hand in hand with a higher requirement of cognitive effort. These findings also (implicitly) support the (revised) Monitor Model (Tirkkonen-Condit, 2005) that suggests that literal translation is the "default rendering procedure" (p. 407-408). The translation process is monitored by an internal monitor function and when it encounters an issue in the rendered translation (e.g. contextual or grammatical), it intervenes and other, less literal, approaches are considered.

Schaeffer and Carl (2013) introduce a revised, recursive, version of the Monitor Model. It suggests that default (literal) translations are produced based on the shared representations of source language (SL) and target language (TL) items that are active in the mind of the translator. If the monitor recognises that the influence of the source text leads to unacceptable (literal) target text, then the automatic process is interrupted. Similarly, Carl and Dragsted (2012) propose that understanding the source text and producing a translation occur in parallel. The production process is monitored and when issues arise, alternative translation options are considered. Such parallel processing is especially straightforward in a copy task but also in literal translation empirical evidence is found to support this view.

In an effort to define literal translation in terms of the equivalence between the source and target text, Schaeffer and Carl (2014, p. 29-30) propose that three criteria need to be met:

1. the word order is identical in the ST and TT;
2. ST and TT items are one-to-one translation equivalents;
3. each ST word has only one possible translated form in a given context.

These criteria for literality have served as the starting incentive for the creation of equivalence metrics that compare the syntactic and (lexico)semantic properties of a source sentence with its translation (cf. Sec. 5.3.3). As such, these metrics operationalise literality and can be used to measure the impact of literality, but also of divergent structures in general, on the translation process.

5.2.2 Measuring the Relationship Between ST and TT

The literal translation hypothesis and the way that is has been operationalised, is often used in translation process research as predictors for cognitive load during translation (Muñoz Martín, 2012). A high cognitive load is indicative of difficulties that a participant is experiencing. Reichle et al. (2009) show that during reading, a participant processes previous information while absorbing new text and during this stage of postlexical processing lexical, semantic or syntactic difficulties may arise that involve previously encountered words.

These difficulties require pauses on the word that triggered the problem or regressions to previous information. Hence, measures involving eye tracking, keyboard logging, and duration data can provide hints towards the cause of translation difficulties. Some of the metrics discussed here will be elaborately repeated in Section 5.3.3 because we use them as predictors in our experiments.

The first point of the definition of literal translation of Schaeffer and Carl (2014) is important in terms of syntactic similarity between ST and TT: "the word order is identical in the ST and TT". Word order can be modelled in different ways but the metrics that are discussed here all rely on word alignment information. Word alignment is the linking of a source word with its translated word(s). This alignment information can then serve as the building block for a variety of metrics. Carl and Schaeffer (2017); Carl et al. (2016); Schaeffer and Carl (2014) propose a metric, Cross, that for each translated word measures the distance from the previously translated word. For each source word, the Cross value is therefore relative to the previous word. Schaeffer, Dragsted, et al. (2016) showed a positive effect of Cross on the translation process, indicating that word reordering needs problem solving effort. Vanroy, Tezcan, and Macken (2019), on the other hand, suggest an absolute, bidirectional metric based on crossing word alignment links. The authors additionally put forward a measure to quantify word *group* reordering. The idea of word group reordering is further elaborated on in Vanroy et al. (in press).

In addition to word (group) order, other syntactic properties have also been suggested. Vanroy et al. (in press) propose a range of syntactic equivalence measures. One measure compares the dependency labels (e.g. subject, object) of source words with their aligned target word, another calculates word group movement in the same fashion as Vanroy, Tezcan, and Macken (2019), but uses syntactically motivated word groups where each group contains words from the same subtree. And a final measure compares the abstract linguistic representations of the source text with that of the target text while also taking word alignment information into account. In Vanroy et al. (in press), these measures were only available on the sentence level. Positive effects of the changes in average dependency label changes on production duration and of syntactic word group reordering on the total reading time on the source sentence were reported. Since then, all those metrics have been re-implemented so that they can be calculated on the word level. It allows us to measure the effect of such word-level, linguistic metrics on the translation process. Similar to the dependency label changes mentioned above, Nikolaev et al. (2020) compares part-of-speech tags (such as noun and adjective) between source words and target words. They also quantify diverging syntactic structures based on different dependency paths of aligned content words. Their research is computational in nature and applied to corpus linguistics, but theoretically it could apply just as well to the comparison of ST and TT in TPR studies.

In the work by Bangalore et al. (2015, 2016) sentences were manually annotated with three syntactic values (valency of the verb, voice type, clause type) and the variation between different participants' translations concerning these features (entropy, as will be discussed in Section 5.3.3) was then used as a predictor for total reading time. A significant positive effect was observed, meaning that sentence-level measure of variance in specific syntactic properties lead to a higher cognitive load.

Using multiple translations to model variance amongst translators can also be applied to semantics. Word translation entropy (Carl & Schaeffer, 2014), for example, focuses on the third point mentioned above: “each ST word has only one possible translated form in a given context”. In a similar, entropy-based, fashion as Bangalore et al. (2015) above, Carl and Schaeffer (2014) uses multiple translations to quantify the agreement among translators concerning lexicosemantic decisions, i.e. it models the certainty of lexical choice. This metric has been shown to have a positive effect on different translation process measures (Carl & Schaeffer, 2017; Schaeffer & Carl, 2017; Schaeffer, Dragsted, et al., 2016), indicating that having many appropriate word forms to choose from leads to a slower or more difficult translation process. Finally, a recent metric was introduced (HSTC) that aims to incorporate a measure of variance among translators in both word group reordering and semantic alignment (Carl, in press). It therefore includes syntactic and semantic information. Similar to word translation entropy, HSTC shows a positive effect on production duration (Carl, in press).

5.2.3 Unit of Translation

In translation studies, (the size of) the unit of translation remains a much discussed topic. On the one hand, the distinction can be made based on the focus of the research, i.e. the translation process or its product. In product-emphasised studies, it is generally accepted that the translation unit (TU) is a *pair* of (a) source item(s) and its corresponding target item(s). In process-based studies, the focus lies on the source text. The translation unit here is considered to be the source item(s) that a translator processes one at a time (Malmkjær, 2006). An overview of this dichotomy is given in Alves and Vale (2009).

In this paper we are particularly interested in the translation unit in the first interpretation because we compare the source text with its translation (the product). However, a lot of work has been done on the unit of translation during the translation process. For instance, Dragsted (2005) found that the size of translation units (or “segments”) differs depending on the difficulty level of a text (smaller units for difficult text) and between novice and professional translators. Professionals tend to work on larger chunks of text at a time. Translation units, in the work of Dragsted but also in related research, are frequently defined as the productive part of the process in between two pauses of a specified length where keyboard activity can be observed. In the experiment of Dragsted, this pause length was chosen by using a formula that takes idiosyncrasies of translators into account.

Rather than investigating a single type of translation unit in process data, Carl and Kay (2011) proposes the usage of different kinds of units as proxies for the TU itself. Source and target pairs of items can be segmented into alignment units (AU; aligned source and target words), the eye-tracking data in fixation units (FUs; consecutive fixations segmented by a pause of a given threshold), and the keystroke data in text production units (PUs; coherent typing behaviour segmented by a pause). By separating the concept of a unit across different parts of the translation process, the authors intend to approximate the "properties and shapes of Translation Units" (p. 972). When the boundaries that constitute these units are chosen correctly, PUs are shown to be a rough approximation of the translation unit, i.e. a unit of cognitive activity. The size of these units in terms of time, as segmented by pauses, differs between novices and professional translators. The PUs of professionals are larger, which indicates the processing of larger chunks at a time, which lends to support to the findings by Dragsted (2005). By extension, Carl et al. (2016) suggest activity units (CUs). Activity units can be categorised according to the activity type at hand such as "translation typing while reading the source text" or "target text reading". There are eight types in total (Carl et al., 2016, p. 38-39).

Alves and Vale (2009), and continued in later work (Alves et al., 2010), make the distinction between micro and macro translation units. A macro TU encapsulates a series of micro TUs. A micro TU is therefore more similar to the TU as it was discussed up to now (a unit of activity segmented by a pause of a given length). Macro TUs, on the other hand, are collections of micro units that are all related to the same source segment. In other words, when different micro TUs all contribute to the production of the translation of a specific word (by inserting or deleting characters or by revising previously produced text), then all of those together are considered the macro TU.

Immonen and Mäkisalo (2010) aim to find overlaps and correlations between syntactic units and the pause boundaries that are typically used to segment translation units. Among other things, their results show that in translation the processing of small units require more processing time compared to a monolingual task, and larger linguistic units are relatively speaking less time demanding. Their explanation for this is that during translation a translator spends a lot of time on getting the translation of small units right in terms of its equivalence to the source text. But for larger linguistic structure this integration requires less time because they are easier to copy from the source text (e.g. the internal structure of a text or paragraph). These findings are confirmed in a later study as well (Immonen, 2011).

It is clear that research is actively involved in the translation unit, but clear-cut definitions do not exist. A translation unit is a variable concept: it differs between participants and tasks, and may or may not necessarily correspond to syntactic units. In this paper, however, we rely on the minimal product-based view that a translation unit is a pair of (aligned) source and target items. We investigate both small, word-based units and larger (word group) units. Section 5.3.3 describes how we define what a word group is.

5.3 Materials and Methods

In this section we describe our dataset and the processing that was applied to it, followed by a description of the regression models that were built and the involved dependent variables. Finally, the predictors are introduced.

5.3.1 Data and Processing

For our experiments, we use a subset of ROBOT (Daems, 2016), an English-to-Dutch translation process data set. In terms of complexity and readability, all texts were chosen to be comparable. Each participant was asked to post-edit machine translations of four texts and translate the other four from-scratch. The translation process was recorded using an EyeLink 1000 eye tracker in combination with Inputlog (Leijten & Van Waes, 2013) and CASMACAT (Alabau et al., 2013). After the translation process was completed, the final translations were manually sentence and word aligned with the source texts with YAWAT (Germann, 2008).

The full dataset consists of post-edited and from-scratch translations of eight news articles by ten student translators (P1-P10) and twelve professionals (P21-P32;P34[3]). Because the translations of P10 were not aligned and because our metrics require word alignments, we could not include that participant's data. P32's eye-tracking data was not included because it was of poor quality, probably due to lenses. The product information of P32 was taken into account for the calculation of entropy values, however (Sec. 5.3.3). In total, that leaves us with 21 translators who each translated three or four texts. That means that the eight texts each have between nine and eleven translations. Segments that were not translated as exactly one target sentence were not included because one of our metrics requires a linguistic parse tree, which is generated on a per-sentence basis.

The translation process research database (Carl et al., 2016, TPR-DB)[4] was used to ingest the aligned source and target texts on the one hand (product data) and the process data on the other to create useful overview tables of the source segments as well as the source and target tokens. Relevant process features were automatically calculated by the TPR-DB, including fixation durations and keystroke information. Product features, such as the (H)Cross feature (Schaeffer & Carl, 2014, 2017), are derived from the final translation and its relation to the source text and are added automatically as well (Sec. 5.3.3). All this information can then be exported into so-called TPR-DB tables where each word is supplied with all of the aforementioned measures and more.

The metrics proposed by Vanroy and colleagues (Vanroy et al., in press; Vanroy, Tezcan, & Macken, 2019) were added at a later stage. A Python script

[3] P33 was not included in the original ROBOT dataset. The reason for this is not known to us

[4] `https://sites.google.com/site/centretranslationinnovation/tpr-db`

that we provide in our library[5] can calculate and add the metrics automatically to the TPR-DB tables. To create the linguistic structures that are needed for one of our metrics, we rely on `stanza` (version 1.2) (Qi et al., 2020) to parse both source and target sentences into the Universal Dependency schema (Nivre et al., 2016) (version 2.7). Both the predictors and the use of Universal Dependencies are discussed in more detail in Section 5.3.3.

5.3.2 Regression Models

We built regression models with dependent variables First Fixation Duration (FFDur), eye-key span (EKS), and total reading time on source tokens (TrtS). FFDur, a very early measure, is the time in milliseconds of the first fixation when a source word is first encountered. Eye-key span is the time between the first fixation on a source word and the first keystroke that contributes to the translation of that word (EKS; Dragsted, 2010; Dragsted & Hansen, 2008). It is therefore a relatively late measure because when a translator starts typing the target word, it is assumed that they have at least processed the source word and perhaps some of its context sufficiently to start producing a translation for it. TrtS, finally, is the total time (sum of fixations) that a translator has spent looking at a source word. It is therefore a very late measure. Initial models for First Pass Duration (FPD) and Regression Path Duration (RPD) were created but those did not yield promising results and were not included in the final paper. FPD is the sum of the first consecutive fixations on a word before moving to any other word (before or after the current word). RPD is a late measure that is the sum of all fixations on a word including regressions to previous words.

For our analyses, we used R (R Core Team, 2020) and the package lme4 (Bates et al., 2015) for linear mixed regressions. To test for statistical significance of the effects, we used the R package `lmerTest` (Kuznetsova et al., 2017). Kurtosis and skewness were calculated with the package `moments` (Komsta & Novomestky, 2015) and we used the `MuMIn` package (Bartoń, 2009) for calculating R^2 for fitted models. Model comparison was carried out with the anova function from the base stats package. Multicollinearity was assessed by using the `vif.mer()` function (Frank, 2011). In order to assess whether the normality assumption of model residuals was met we used the package `moments` (Komsta & Novomestky, 2015) to compute kurtosis and skewness of model residuals. A skewness of $> |2|$ and kurtosis of $> |7|$ are considered as severe deviations from the normality assumption regarding model residuals (Kim, 2013). We use the `effects` package (Fox, 2003) to visualise model results.

Prior to model building, for each dependent variable, we excluded data points from the raw data which differed by more than 2.5 standard deviations from the mean for each participant. This resulted in no case in a loss of more than 3%. All models had, as random variables, participant and item (this

[5] `https://github.com/BramVanroy/astred/blob/master/examples/add_features_tprdb.py`

was the source word for all models). The first model we built always included HCross – whether it was significant or not. We then included word form frequency (from the English Lexicon Project (Balota et al., 2007) for the reading times on the source text. We also included the sequential numbering of tokens in the source texts (STid) and in the sentence (word_id) as predictors. However, if inclusion of both these variables meant that the model did not converge, only one of these – whichever was more significant – was included. In subsequent models, we substituted HCross for the new metrics (Sec. 5.3.3) one by one, allowing for a comparison between the models with HCross as a predictor and otherwise identical models via the anova function (we report results from the χ^2 test). We use HCross as the base model because it is a syntactic, entropy-based measure. The metrics by Vanroy and colleagues are also syntactic, but not entropy-based, which can lead to an interesting comparison. More on that in Section 5.3.3. If convergence was not possible in subsequent models with the new predictors, we excluded predictors one by one until convergence was possible and compared these to a base model with the same predictors – apart from HCross. After comparing models with the new predictors to the base model with HCross, we excluded residual outliers ($>$ 2.5 SD from the mean) and report model results with and without removal of residual outliers. Finally, we compared models in which the critical predictors were significant with each other, again via the anova function. We report results from the χ^2 test, and Akaike's Information Criteria (AIC; Akaike, 1973) and Bayesian Information Criteria (BIC; Schwarz, 1978) are used as indicators of goodness-of-it of individual models without outliers. We also report marginal R^2 for both versions of each model (with and without residual outliers), which reports the variance of the fixed effects only. In all models, skewness was below $|1|$ and kurtosis below $|3|$ after exclusion of residual outliers. Variance inflation factors in all models were below 2.

5.3.3 Predictors

In this study, we measure the effect of different predictors on process data. These predictors each focus on different relations between the source text and its translation(s). A couple of distinctions can be made, such as the one between (lexico)semantic measures and syntactic ones, metrics that are calculated on the word versus the word group level, and predictors that rely on the availability of multiple candidate translations to calculate probability values in contrast to those that are calculated between a source sentence and a single translation. We will start the description of predictors with Cross, and continue to discuss other metrics that are also available in the TPR-DB. After that, the recently introduced metrics by Vanroy and colleagues will be discussed alongside some improvements that were made to their initial implementation.

Cross (Carl & Schaeffer, 2017; Carl et al., 2016; Schaeffer & Carl, 2014) quantifies the reordering of a word's translation relative to the position of the previous word's translation. That means that Cross values can be positive (when the translation is placed after the previous one) or negative (when it is placed before the previous translation). In an absolute literal translation where a one-to-one relation exists between every source word and a corresponding target word (Schaeffer & Carl, 2014), and where the word order is maintained, every word has a Cross value of 1 (because each translation is one step further than the previously translated word). In this study we focus on the source side so we are only interested in CrossS; Cross calculated on source tokens as opposed to CrossT (calculated on target tokens). An example showing the difference between Cross and other reordering metrics is given in Figure 5.1. Similar to previous studies, we use the absolute value of Cross in our experiments (Carl & Schaeffer, 2017; Schaeffer & Carl, 2017; Schaeffer, Dragsted, et al., 2016). In previous research, (absolute) Cross values were found to have a significant positive effect on first fixation duration and total reading time (Schaeffer, Dragsted, et al., 2016).

As an extension to the relative word reordering of a single translation, Schaeffer and Carl (2017) introduce the concept of HCross, which is an entropy-based variant of Cross. Entropy (Formula 5.1) is a measure from information theory to quantify the added value of new information (Shannon, 1948). Applied to our use cases, entropy can be interpreted as the amount of agreement between translators or the amount of uncertainty with respect to a given phenomenon. Low entropy values mean high agreement (or low uncertainty), and high entropy would indicate low agreement (high uncertainty). As such, multiple translations of the same text need to be available to have meaningful entropy results. By taking as many shadow translations into account as possible ("possible alternative translations defined by the systemic potential of the target language"; Matthiessen, 2001, p. 83), the hope is to approximate all translation possibilities and by extension model the entropy; the uncertainty for choosing between all those options.

$$H(X) = -\sum_{event \in X} P(event) log_2 P(event) \tag{5.1}$$

where:

X	a set of possible unique events
$P(event)$	the probability of a given event

The general entropy formula is applied to Cross by the authors as in Formula 5.2. Instead of only considering a single translation, entropy is calculated on all available translations of the same source text. In other words, by taking into account the translations of the same source text by different translators, HCross can quantify how pre-determined the reordering of a source word must be. If there is little variation in the Cross values for a source word among different translators, then the entropy will be low. For high statistical variance, the entropy value will be high. Put differently, if translators reorder a source word in the same way (and agree about the repositioning of the translation), then HCross will be low, and otherwise it will be high.

$$HCross(w, C) = -\sum_{c \in C} P(c|w) log_2 P(c|w) \tag{5.2}$$

where:

C	a set of unique Cross values associated with w in this context
$P(c\|w)$	the probability that w has a Cross value of c in this context

HCross has been shown to correlate with word translation entropy (see below), a measure for lexical entropy, both within and across languages (Carl, in press; Carl et al., 2019). That is unsurprising: different words in the target language may require different word orders, which in turn may be an indicator of different syntactic structures. Schaeffer and Carl (2017) further present that HCross has an effect on the duration of the eye-key span.

Where HCross is a way to quantify the uncertainty of word reordering, word translation entropy (HTra; Carl & Schaeffer, 2014; Carl et al., 2016) does the same for the lexical choice for a translation. For a given source word, HTra takes all translations of that word in the specific context into account. Depending on how much agreement or disagreement there is between translators to choose the same target word, HTra will be low or high, respectively. Applying Formula 5.1 to word translation entropy, HTra can be defined as Formula 5.3.

$$HTra(w, T) = -\sum_{t \in T} P(t|w) log_2 P(t|w) \tag{5.3}$$

where:

T	a set of unique translations of w in this context
$P(t\|w)$	the probability that w is translated as t in this context

This measure is thus a way to see how many translations (lexical entries) are suitable translations. It gives us a (limited) insight in the different options that translators can choose from (contextual lexicon). A high HTra value means that many options are available and that a single, straightforward choice is not necessarily available. As a consequence, a high word translation entropy is expected to have an impact on process data as well: more choices to choose from for a given word in a specific context, is likely to require more time to make a decision. This has been confirmed in a number of studies. Effects of HTra were reported on production duration (Dur) (Carl & Schaeffer, 2017), first fixation duration (FFDur) and total reading time (TrtS) (Schaeffer, Dragsted, et al., 2016), and eye-key span (Schaeffer & Carl, 2017). This would mean that the effect of word translation entropy is present in both early and late processing stages during translation.

Recently, a new entropy-based metric has been introduced that incorporates different types of information into a single metric (Carl, in press). It is called “joint source-target alignment / translation distortion entropy”, or HSTC for short, and takes into account translation and reordering probabilities. Specifically, a given source word w is part of a group of source words s, which is aligned to a group of target words t. An alignment group is defined as a number of source and target words that are aligned with each other. These groups represent meaning-equivalent expressions in the context of the sentence. All words in a source group s have the same Cross value c. As such, the probability of a specific translation can be calculated in terms of its self-information of the source group, the target group and its relative re-ordering. Worded differently, the joint alignment/distortion probability for a given source word w is based on its associated source group s, the alignment with target group t, and the corresponding Cross value c. These probabilities can then be used to calculate the entropy (Formula 5.4). In a way, HSTC encompasses both HTra and HCross discussed above. It is intended as a single metric to measure the (non-)literality of a translation, both (lexico)semantically and syntactically.

$$HSTC(w, A) = -\sum_{(s,t,c)\in A} P(s,t,c|w) log_2 P(s,t,c|w) \tag{5.4}$$

where:

w a given source word

A a set of unique triplets of associated values of w in this context

s the source word group that w belongs to in this set, aligned with its respective t

t the target word group that w belongs to in this set, aligned with its respective s

c the Cross value of all words in group s

$P(s,t,c|w)$ the probability that w is associated with this source group s, target group t, and Cross value c in this context

Carl (in press) shows that, perhaps unsurprisingly, HSTC correlates strongly with both HTra and HCross, which implies that uncertainty in choice of lexical translation goes hand in hand with similar uncertainty about the reordering. Similar to the aforementioned measures, Carl shows significant effects of HSTC on production duration during translation.

With the exception of Cross, the above measures are all meant to be calculated involving a (relatively high; Carl, in press) number of translations. The main idea is that a sufficient number of translations approximate all the possible choices that translators are faced with, and that more choices (or less-straightforward ones) lead to a more difficult translation process. Vanroy and colleagues introduced different syntactic metrics that are not reliant on multiple translations and each focus on different aspects of syntactic differences between a source text and its translation (Vanroy et al., in press; Vanroy, Tezcan, & Macken, 2019). Instead of trying to comprise "one metric to rule them all" such as HSTC, where a lot of information is included in a single measure, they split up syntactic (dis)similarities between a source and target text into individual measures. In the current section we will discuss three that are used as predictors for our experiments.

Cross, as discussed, above is a metric to measure the reordering of a word's translation relative to the translation of the previous word. It is directional, in the sense that a word and its translation can have different values. In Vanroy, Tezcan, and Macken (2019), we suggest a different approach to word reordering that is bidirectional and absolute. We will call this metric `word_cross` in the current paper to distinguish it from the aforementioned Cross value (Carl & Schaeffer, 2017; Carl et al., 2016). First, `word_cross` is calculated as the number of times an alignment link of a specific word crosses the alignment link of *any* other word in the sentence. Formally, two alignment links cross each other if the order of the source words is inversed on the target side. An example is given in Figure 5.1.

In other words, whereas a word's Cross value is determined by the reordering of its translation relative to the previous word's translation, its `word_cross` value is impacted by the reordering of *all* words in the sentence, including its own. The implication of this is that the cross value of a target word is the same as the cross value of its aligned source, at least in one-to-one alignments. If a word is aligned with multiple target words, we can choose to take the average cross value of its alignments, or sum them up (in this paper we sum them), which means that for some aligned structures the cross value of a source word could differ from its aligned target word, because that target word is aligned with other source words as well. In Vanroy, Tezcan, and Macken (2019) and later in Vanroy et al. (in press), this metric was only available as an aggregated value on the sentence level and could therefore not be used for word-level predictions or correlations. The reason for this is that we initially wanted to make word (group) order distortion predictions for a given sentence, i.e. we were answering the question whether we can predict the difference in word (group) order between a source sentence and its translation. In the current paper, we use the word-level values as predictors.

Similar to Gile (1995, pp. 101-102), we consider that the translation unit can vary and is not necessarily restricted to only words nor to only word groups. The unit of translation may differ between translators, between tasks and even specific texts and difficulties (Sec. 5.2). Therefore, we also investigate the effect of word *group* (or *sequence*) reordering on process data. Similar to `word_cross` above, `seq_cross` was introduced in Vanroy, Tezcan, and Macken (2019) and further discussed in Vanroy et al. (in press), but in both cases the metric was calculated on the sentence level. Here, we make two improvements: first and foremost, each *word* is now assigned a `seq_cross` value, which is the cross value of its word group. Word groups can be created based on the alignments of the involved words and restrictions apply as per the requirements in 23, taken from Vanroy, Tezcan, and Macken (2019). If a word does not belong to a group that follows these requirements, then that word's original word alignment will be used as a "singleton" sequence alignment as well.

(23) a. Each word in the source sequence is aligned to at least one word in the target sequence and vice versa
b. Each word in the source word sequence is only aligned to word(s) in the target word sequence and vice versa
c. None of the alignments between the source and target word sequences cross each other

So looking at this from a technical perspective, aligned word groups are created as described above, and for these word groups and their alignment a cross value is calculated in the same fashion as for `word_cross`. It is ensured that these groups are as large as possible according to the requirements. The sequence cross value of a group is passed on to all the words belonging to that group. Each word thus has a `word_cross` value, based on word alignment and its own reordering, and a `seq_cross` value that is based on the alignment of the word group that it belongs to. These sequence alignments (alignment between two word groups) can greatly reduce the number of alignments and, consequently, the cross values calculated on these groups (`seq_cross`) can be much smaller than their `word_cross` equivalent because there are less (group) alignments present in the sentence to cross compared to word alignments.

A second improvement compared to when we first introduced this metric, is that we consider m-to-n alignments of consecutive items as valid aligned word groups, too. In other words, requirement 23c does not apply to these so-called multi-word groups (MWGs), but as an alternative requirement all source words need to be aligned with all target words of the construction. The assumption here is that m-to-n alignments are used for groups of words or phrases that cannot be easily compositionally aligned, such as idioms or free translations of specific concepts. Semantically, however, the source and target side should constitute the same concept or phrase. Note that this does not necessarily mean that from a monolingual perspective these constructions are multi-word expressions or idiomatic expressions: MWGs are purely based on the alignments between the source and target words belonging to the construction. As an example of a MWG, consider the following translation, where "marine sentinels" - "wachters van de zee" constitutes a MWG according to our specification and as such only one alignment link will be needed between the two groups rather than the m-to-n word alignments (which would lead to a lot of crosses because all word alignments in m-to-n alignment cross each other).

(24) a. Whales are often called **marine sentinels**
b. Vinvissen worden ook wel **wachters van de zee** genoemd
Whales are also guardians of the sea called

c. Word alignments: 0-0 1-1 2-2 2-3 3-8 4-4 4-5 4-6 4-7 5-4 5-5 5-6 5-7

Note that allowing m-to-n alignments to be groups, also greatly reduces the sequence cross value of other words: because "called" is aligned with "genoemd" it crosses the m-to-n alignment, leading to a large word_cross value of 8. However, its sequence alignment (which is the same as its word alignment), has a seq_cross value of 1 because the m-to-n construction that it crosses is considered a valid sequence and only has one alignment link connecting "marine sentinels" to "wachters van de zee" instead of eight. Example 24 can be visualised as in Figure 5.1. It shows the differences between Cross, word_cross, and sequence cross. The groups of words that adhere to the requirements above are boxed in and aligned (solid black lines). Their original word alignments are given in grey dotted lines. If a word does not belong to a multi-word group, it is its own singleton group (like "called" in the example). Cross and word_cross are calculated on the alignments of the single words, whereas sequence cross uses the alignments between word groups. On the word-level (based on word alignments), "called" crosses eight alignment links. On the word-group level, however, this is reduced to only one.

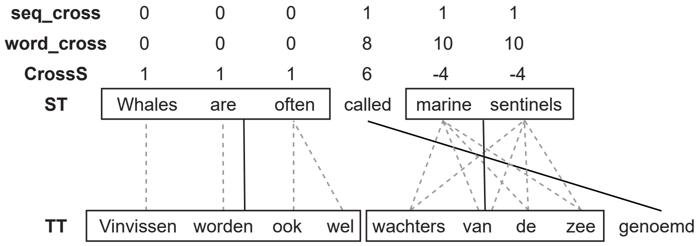


Figure 5.1. A visualisation of Cross, word_cross and sequence cross in Example 24

Finally, in this paper we will also investigate the effect of a linguistic measure that is a word-based by-product of the metric that we called Aligned Syntactic Tree Edit Distance (ASTrED; see Vanroy et al., in press, for an in-depth explanation and examples). The syntactic structure of a sentence can be represented as a hierarchical tree where each child presents a lower item in the tree to its parent. Specifically, we make use of dependency trees where each word has a to-relationship with its parent in the tree. That means that each node in a tree is the dependency role label of that word (for instance, a word can have the role of subject *to* the root verb; see Fig. 5.2 for an example). The structure of the source tree can then be compared with a target tree representation to find structural differences between the two. To do so, however, the label set and way of structuring a sentence needs to be comparable between languages in the first place. Therefore, we make use of the

Universal Dependencies annotation scheme [6] (UD), which is an initiative to facilitate and accelerate multilingual, comparable research (Nivre et al., 2016). It is specifically designed to do away with the prior difficulty of comparing two languages syntactically. As an example, Figure 5.2 shows dependency tree of the sentence "This morning I saw the baker preparing cookies" where the nodes are represented as `word:dependency-label`. In reality, however, only the dependency label is used in comparing the structures.

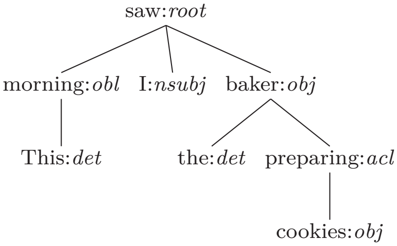


Figure 5.2. Example dependency tree of the sentence "This morning I saw the baker preparing cookies"

Because we are certain that the structures of a source text and its translation use the same annotation scheme, we can compare the tree representation of a source sentence and its translation. One could naively measure the tree edit distance (TED) between the two, a common metric to measure differences between trees. TED looks for the most optimal way to transform the source tree into the target tree by making use of different operations: match (when a source node has the same label as a node on the target side in the same position), insertion (when a node is not present in the source tree but needs to be inserted in the target tree), deletion (when a source label is not present in the target tree and needs to be deleted) and substitution (also called rename; when a source label is structurally correct but its label needs to be changed to be identical to a target node). Every operation has a cost attached to it, and the TED algorithm needs to look for the sequence of operations that has the lowest total cost. In our case, match has no cost to it (and is thus the preferred operation if possible), and the others have a cost of 1. TED as-is is a naive approach, however, as it will not take word alignments into account. It will simply find the most optimal solution to change the source sentence *structure* into the target structure, irrespective of word alignments and effectively ignoring any semantic or structural correspondence between the source and target sentences. ASTrED, on the other hand, can be seen as a preprocessing procedure for syntactic trees that ensures that only aligned words can match in the source and target tree by merging the node labels in both the source and target tree to include information about the aligned words. This procedure is described in much detail in Vanroy et al. (in press) and will not be duplicated

[6] See `http://universaldependencies.org/` for label descriptions

here for brevity's sake. Important to know is that ASTrED changes the node labels in such a manner that the nodes of aligned source and target words will end up having the same label in their respective trees so that only words that are semantically aligned can match each other. Because match is a preferred operation (cost 0), this ensures that TED will try to match aligned words (rather than words that coincidentally have the same label) in the tree and fill out the rest of the tree with substitution, insertion, and deletion operations.

In this paper, we do not use the calculated ASTrED value directly, but instead for each source word we see if it was matched (and not changed) or whether an edit operation was necessary to transform this specific node to create the target tree (changed). These operations can be deletion or substitution, as insertion can only happen for target words. Each word, then, has an `astred_change` value of "FALSE" (match) or "TRUE" (no match), indicating whether a specific operation needs to occur on this word.

To summarise, the predictors that we use can be divided in a number of ways. Some of them are semantic in nature (HTra), others are syntactic (Cross, `word_cross`, sequence cross, HCross, `astred_change`) and HSTC is both. Most measures consider the word as unit, but some use word groups in their calculation as well (sequence cross, HSTC). Finally, HSTC and HTra require multiple translations, whereas the other are calculated between a given source text and its (one) translation.

5.4 Results

In this section we present the effects of the predictors `word_cross`, sequence cross, `astred_change`, absolute Cross, HCross, HTra, HSTC on three eye-tracking measures: First Fixation Duration, Eye-Key Span, and Total Reading Time. In the overview tables, the "ANOVA (HCross)" column compares each model individually with HCross (χ^2). "ANOVA" compares for each model whether it significantly improved over the previous model (models are ordered based on BIC/AIC values with the best fitting model at the bottom). "base" indicates when a model has been used as the first reference model in an ANOVA. When the models are compared, all residual outliers are included. The variance that they account for is given in "R^2 (outliers)". Separate models are also built that exclude for each model its respective residual outliers. These results are reported in "R^2 (no outliers)". In each table only those predictors are included that had a significant effect (with or without outliers) on the dependent variable. Significance of the specific predictor under scrutiny are given in the p columns. The individual significance levels of secondary fixed effects (ID in source text, ID in source sentence, frequency) were not reported but in all cases they were significant ($p < 0.05$). The BIC and AIC columns are given for transparency to indicate the absolute goodness-of-fit of the models (lower is better), as discussed in Section 5.3.2.

5.4.1 First Fixation Duration

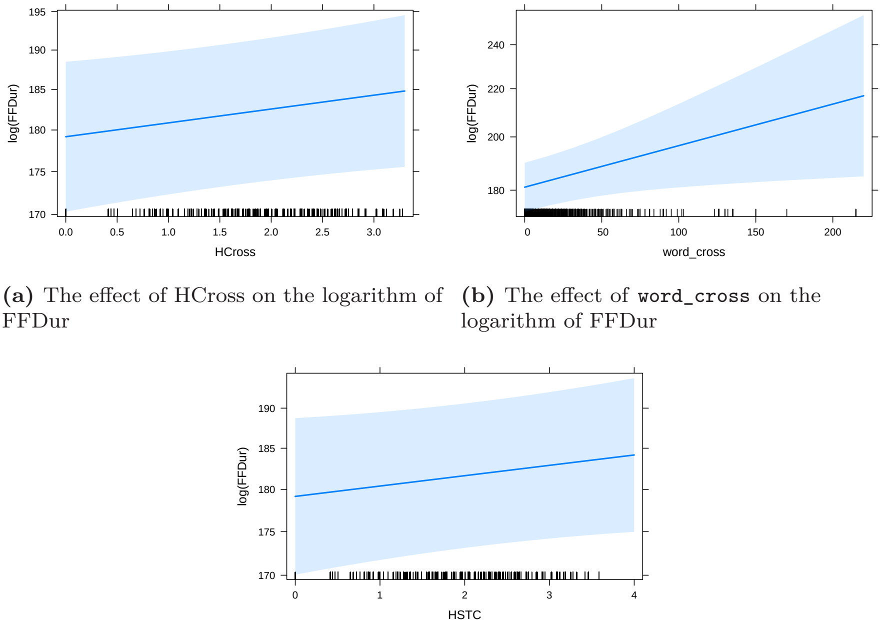
Table 5.1 shows the summary of significant effects on First Fixation Duration (the earliest measure) of which there are few. HCross, word_cross and HSTC have a significant effect. HCross performs best in terms of BIC/AIC as well as R^2 when outliers are included. Neither word_cross nor HSTC perform better according to the ANOVA. However, when outliers are removed, only word_cross has still a significant effect suggesting that outliers were driving the effects in HCross and HSTC in the first place. Only very little variance is explained in these settings.

	w. residual outliers						w.o. residual outliers	
	ANOVA (HCross)	**ANOVA**	**BIC**	**AIC**	p	R^2	p	R^2
HCross	base	base	9154.9	9106.1	0.018*	0.0023	0.077	0.0023
word_cross	ns	ns	9156.1	9107.2	0.034*	0.0021	0.023*	0.0025
HSTC	ns	ns	9156.6	9107.7	0.046*	0.0021	0.153	0.0022

*$p < .05$; ns = not significant
See the introductory paragraph in Section 5.4 for an explanation of the column names

Table 5.1. Summary of effects on First Fixation Duration (FFDur)

The effect plots for the base model HCross, word_cross and HSTC are given in Figures 5.3a, 5.3b, and 5.3c respectively. Important to note is the difference in scale of the y-axis.



(a) The effect of HCross on the logarithm of FFDur

(b) The effect of `word_cross` on the logarithm of FFDur

(c) The effect of HSTC on the logarithm of FFDur

Figure 5.3. Effects on First Fixation Duration

5.4.2 Eye-key Span

Eye-Key Span is considered a late measure, assuming that the translator has fixated a word long enough to at least start producing a translation for it. It does imply, however, that initial problems have been resolved when the production of a word starts (but revision may still happen at a later stage). Many predictors show a significant effect. However, sequence cross only converged when the word ID (the index of the word in the sentence) was excluded as a predictor (the corresponding model is called `seq_cross`$^{+}$). Therefore, a separate HCross model was built (HCross^{+}) that similarly contains the source text ID (the index of the word in the text) and word frequency, but not the word ID. With these fixed effects, seq_cross performs significantly better than HCross according to the ANOVA but it is also evident from their respective BIC/AIC values. On top of that, HCross does not have a significant effect in this context. For that reason, the HCross^{+} model was not included in the second ANOVA.

	w. residual outliers						w.o. residual outliers	
	ANOVA (HCross)	**ANOVA**	**BIC**	**AIC**	p	R^2	p	R^2
HCross$^+$	base				0.146	0.0088	0.515	0.0073
`seq_cross`$^+$	***	base	20 259.6	20 213.3	0.047*	0.0090	0.019*	0.0079
HCross	base	***	20 200.4	20 147.5	0.037*	0.0192	0.129	0.0231
abs(Cross)	***	***	20 200.3	20 147.4	0.034*	0.0190	0.069	0.0230
`astred_change`	***	***	20 200.2	20 147.3	0.032*	0.0192	0.040*	0.0229
HTra	***	***	20 196.0	20 143.1	0.003**	0.0204	0.008**	0.0240
HSTC	***	***	20 192.4	20 139.5	***	0.0207	0.002**	0.0243

$^+$ without `word_id` as a predictor

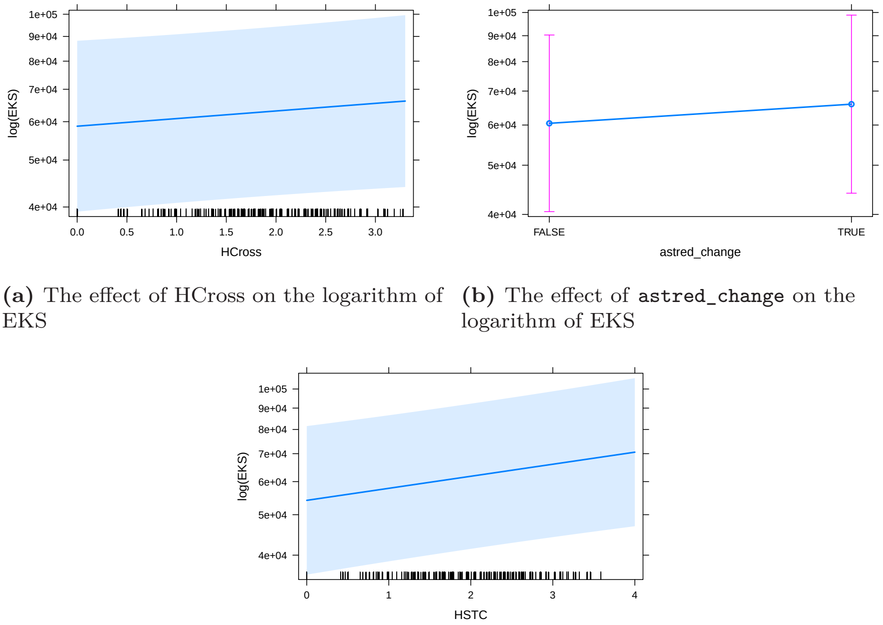
$^{*}p < .05$; $^{**}p < .01$; $^{***}p < .001$

`seq_cross` only converged without `word_id` (ID in the sentence)

See the introductory paragraph in Section 5.4 for an explanation of the column names

Table 5.2. Summary of effects on Eye-Key Span (EKS)

The models that did converge with all secondary predictors and that are significant, are HCross, absolute Cross, `astred_change`, HTra and HSTC. The base model HCross (Fig. 5.4a) is significantly outperformed by other predictors and its variant without residual outliers is not significant. The same is true for absolute Cross. `astred_change` has a significant effect both with and without outliers (Fig. 5.4b). Word translation entropy (HTra) and especially HSTC (Fig. 5.4c) provide the best fitting models to the data.



(a) The effect of HCross on the logarithm of EKS

(b) The effect of `astred_change` on the logarithm of EKS

(c) The effect of HSTC on the logarithm of EKS

Figure 5.4. Effects on Eye-Key Span

5.4.3 Total Reading Time

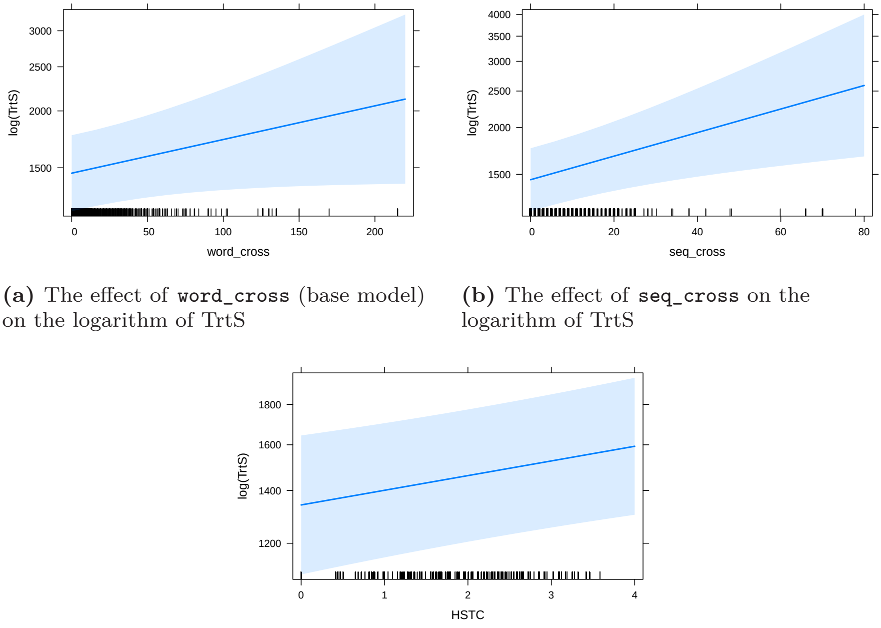
Similar to Eye-Key Span, Total Reading Time (the latest measure which includes all fixations on a word), is affected by many predictors. The base model, HCross, does not have a significant effect so it is no surprise that all other predictors that have a significant effect also perform significantly better than HCross ("ANOVA (HCross)"). Most predictors have a significant effect with and without residual outliers with the exception of `word_cross`, which is not significant without. With outliers included in the model it is only marginally significant ($p = 0.058$; in all others cases $^*p < 0.05$). Sequence cross and HSTC, word group based metrics, are the best performing models according to their BIC/AIC, with HSTC coming out on top. Their effect is highly significant ($p < 0.01$). Absolute Cross is the third best fitting model followed by HTra and finally `word_cross`. The fixed effects in the HTra model explains the most variance in Total Reading Time, however. Note that HCross did not have a significant effect. Therefore, it was not part of the second ANOVA comparison. In that case, the `word_cross` model was the reference model (because it has the highest BIC/AIC), although it was just marginally significant in the first place.

	w. residual outliers						w.o. residual outliers	
	ANOVA (HCross)	**ANOVA**	**BIC**	**AIC**	p	R^2	p	R^2
HCross	base				0.235	0.0345	0.362	0.0395
`word_cross`	***	base	22 593.4	22 537.6	0.058*	0.0346	0.062	0.0346
HTra	***	***	22 592.5	22 536.6	0.033*	0.0358	0.016*	0.0417
abs(Cross)	***	***	22 591.1	22 535.3	0.015*	0.0348	0.004**	0.0400
`seq_cross`	***	***	22 590.1	22 534.3	0.009**	0.0349	0.005**	0.0401
HSTC	***	***	22 589.7	22 533.9	0.007**	0.0359	0.004**	0.0411

$^{*}p < .06$; $^{**}p < .01$; $^{***}p < .001$
See the introductory paragraph in Section 5.4 for an explanation of the column names
BIC/AIC columns have been rounded for conciseness sake but they are in descending order

Table 5.3. Summary of effects on Total Reading Time of source tokens (TrtS)

The effects of `word_cross` (the base model for the ANOVA comparison), sequence cross and HSTC are visualised in Figures 5.5a, 5.5b, and 5.5c respectively.



(a) The effect of `word_cross` (base model) on the logarithm of TrtS

(b) The effect of `seq_cross` on the logarithm of TrtS

(c) The effect of HSTC on the logarithm of TrtS

Figure 5.5. Effects on Total Reading Time on source tokens

5.5 Discussion

In our experiments, we see very little effect of our predictors on the early measure of First Fixation Duration (FFDur) and those that are significant only explain variance by a minimal amount. Furthermore, both HCross and HSTC (both entropy measures) lose their significance when their residual outliers are removed. The effect of HTra and absolute Cross on FFDur as reported in Schaeffer, Dragsted, et al. (2016) could not be reproduced (but this can likely be attributed to the smaller size of our dataset), although HSTC was significant without outliers, which is interesting because it contains both reordering and translation entropy (of the word group). `word_cross` was significant both with and without outliers but again, the variance explained was very small.

The effects in later measures are much more prominent. In EKS, a positive effect of sequence cross can be observed but the explained variance is low as is the significance of the predictor. This effect is only present when the word ID predictor is dropped. Because of that, a fair comparison cannot be made with the other predictors by themselves for this dependent variable. Except for `word_cross`, which is not significant, all other predictors show a positive significant effect. Especially the measures involving semantic information perform well (HSTC, HTra), closely followed by structural changes between ST and TT (`astred_change`). Absolute Cross is further behind, with a considerable gap in BIC/AIC between `astred_change`. It is also not significant without the outliers. The same is true for HCross. Therefore, we can cautiously confirm the results reported in Schaeffer and Carl (2017) where HCross was shown to affect EKS positively, although the effect disappears when the residual outliers are removed. With more certainty, we report results in line with Schaeffer and Carl (2017) concerning the significant positive effect of HTra on EKS.

In Total Reading Time, similar effects can be observed with respect to the semantic measures (HTra, HSTC). Interesting, however, is that both absolute Cross and sequence cross perform slightly better than HTra in terms of BIC/AIC although HTra still explains more variance. We can therefore also confirm similar findings by Schaeffer, Dragsted, et al. (2016) concerning the effect of HTra on TrtS. `word_cross` is only marginally significant and only with its residual outliers included, but sequence cross, on the other hand, is highly significant ($p < 0.001$) and performs significantly better than absolute Cross, although the difference in R^2 is minimal. All predictors explain more variance in TrtS than any predictor could in EKS. The reason for this may lie in late, conscious processes. Even after a translation is being generated (EKS is the time from the first fixation on a word until the first keystroke that contributes to its translation), additional fixations on a word may indicate control and revision processes that are active. The implication could be that more divergent source and target structure (in terms of the significant predictors) require longer control and/or revision processes but this needs further investigation. Surprisingly, the significant positive effect of `astred_change` did not continue in TrtS. This could be related to the aforementioned control processes: syntactic divergent structures may have a significant impact on the problem-solving process right before a translation can be produced (right before the first keystroke of the translation of a word; EKS), but as soon as that problem is resolved, such structural issues are not likely to cause issues during later fixations on the word (i.e. during production or revision).

Because both sequence cross and HSTC involve word groups, it is tempting to attribute their significant effects on late processes, especially TrtS, to a gradual increase of the cognitive unit of translation (from individual words to larger groups in later stages of the translation process). However, because absolute Cross is word-based, the suggestion would be that the unit of translation increases in a compounding manner. In other words: in later stages of the translation process, *both* individual words and (surrounding or involved) word groups are important to the translator. During later processes, a translator may be trying to incorporate or resolve larger units while still taking into account the properties associated with the single word. As mentioned before, a lot of research exists on translation units (e.g. Alves et al., 2010; Carl & Kay, 2011; Immonen & Mäkisalo, 2010; Schaeffer, Carl, et al., 2016), and we do not make any conclusive interpretations that confirm or refute any of the suggestions, but we observe that the (possibly changing) unit of translation and its corresponding features may play distinct roles during the time course of the translation process. This is similar in thought to Alves et al. (2010, p. 121): "translators navigate between different linguistic units and levels during translation". Further research in this direction would be useful. Particularly, interaction effects of word-based and group-based metrics on process data can shed a light on the importance of the properties of the involved translation units during different stages of the translation process. In addition, interaction effects between (lexico)semantic and syntactic properties should also prove interesting, and has already been investigated in some detail by Ruiz and colleagues (Ruiz et al., 2008; Ruíz & Macizo, 2019).

Why we found more effects in late measures (EKS, TrtS) compared to early eye-tracking measures is not easy to explain. One possibility is that our metrics especially model language properties that need conscious decisions. Whereas early measures are often indicative of automatic processes, later measures hint towards conscious decision-making and problem solving, which cannot be resolved automatically (Bell, 1998; Kiraly, 1995). This explanation works for the syntactic measures, where it is conceivable that reordering (Cross, `word_cross`, sequence cross, HCross, partly HSTC) and insertions and deletions (partly what ASTrED models) need more specific attention from the translator. But it does not explain why semantic measures such as HTra and HSTC only have a late effect; the variance in FFDur that is explained by the fixed effects (with HSTC) is very small and HSTC does not have significant effect when residual outliers are excluded. It may be the case that TL features *are* activated during first contact but that they simply do not pose a problem yet. Another likely explanation is that more data (in terms of the number of data points) is needed to show consistent, early effects.

Conclusions concerning entropy are hard to make because a variety of factors are involved. HTra and HSTC both have a semantic component, whereas HCross and HSTC contain syntactic information. HSTC involves word groups, whereas HTra and HCross are metrics on the word level. A single statement on the effect of entropy cannot be made. What we can indefinitely say, though, is that more translations could change the picture. Carl (in press) shows that HTra scores only approximate a real population with a Pearson correlation of more than $r = 0.8$ when approximately ten translations are available for a given text (we have between nine and eleven). It is hard to tell then whether entropy-based metrics based on more translations would have a greater effect on the process data. The goal of entropy-based metrics is to approximate the real population, i.e. all shadow translations (Matthiessen, 2001) that a translator needs to choose from. Entropy, then, serves as a way to quantify how straight-forward the choice out of all those options is. It should be clear that this intention needs many translations in order to approximate “all” translation options.

Although hard conclusions are hard to draw because of the size of our dataset, our results indicate that particularly late process measures are affected by the predictors. The reason for this may lie in the conscious processes that occur in such late stages, like problem-solving and revision. In addition, we find that HSTC, an entropy-based metric that incorporates both word group translation and reordering probabilities, is the best-fit predictor across the board. This is perhaps unsurprising, exactly because it entails both syntax and lexicosemantic information while also being based on all available translations. In terms of metrics that are not based on probabilities, absolute Cross has a consistent significant effect in the late measures. Sequence cross, which is based on word-group reordering, has a particularly strong significant late effect which poses interesting questions about the cognitive unit of translation and how that unit might change during the translation process.

5.6 Conclusion

In this paper we investigated the effect of a number of predictors that each model different parts of the relationship between a source text and its translation(s). Although our results are promising, “it is dangerous to make sweeping generalizations about translation processes” (Tirkkonen-Condit, 2005, p. 406), particularly because our dataset is limited in size. We encourage other research to confirm or refute our findings with experiments involving different tasks (e.g. sight translation) and datasets (different language pairs, more data points). Furthermore, we wish to emphasise that controlled experiments are necessary if fine-grained linguistic concepts are involved whose effects may not be as clear-cut in empirical corpus-based translation studies. In future research we want to particularly focus on more language pairs and see how well the effect of syntactic and semantic divergence generalises to other languages. In addition, we would like investigate additional measures, such as those of Nikolaev et al. (2020) discussed above.

Specifically for the PreDicT project, it is very promising to see that metrics that do not rely on multiple translations also show an effect. Ultimately we wish to predict the difficulty of a given source text, and these results indicate that such singular metrics have predictive power as well. Technically speaking, that is very important: it is much easier to find parallel corpora with one translation than with multiple translation. Such large parallel corpora can be used to train a machine learning model to predict these relevant features (e.g. `astred_change`) for a given source word, which in turn can be used in a translatability measuring system which predicts difficulties for a given source text without access to a translation.

Our main contributions lie in introducing new metrics to the existing arsenal of product-based features that can be calculated on a source word and its translation. We also confirmed pre-existing findings by fellow researchers in the field and made our own observations by measuring the effect of a set of predictors on translation process data. And finally, with our results we believe to have added interest to a number of existing research questions that are keen to be investigated, especially involving the (size of) the translation unit, the distinction between (lexico)semantic and syntactic predictors (and their relevance in the time course of the translation process), and whether or not entropy-based measures are a necessity in predicting cognitive effort.

Chapter 6

Discussion

The previous chapters show the road that I took with the much appreciated help of my coauthors and supervisors. At every step, I gained new insights and expertise. Both in terms of methodological approaches and knowledge of existing literature, I had the chance to improve along the way. Even though the initial few months inspired me with tantalising ideas of how to continue my research – translation difficulty and its related *linguistic* problems are an incredibly broad topic after all – it quickly became clear that I could not let go of my linguistic interest rooted in syntax, and from Chapter 3 onwards, the course of research was set to translation difficulty that is caused by syntactically diverging source and target structures. That is not to say that syntax is more important than other properties of language but it indicates on the one hand my own interests and on the other the many different aspects of translatability that can be investigated.

Because my experience with the topic of translatability, and with research in general, evolved over time, I have had many opportunities to look back on previous work. Things that could have been done differently or for which an alternative approach would have been interesting as well. In what follows, I will provide a critical reflection on each chapter as well as a return to the questions posed in the introduction. Such a discussion is not intended to discredit the publications themselves. I stand by the work that I and my colleagues published, and I take the thorough peer-review process that they were subject to as an indicator of their quality (Chapter 5 is submitted but not reviewed yet, however). This consideration serves only as an expansion on what my thoughts were at the time, how the research process proceeded, and which choices, methodological or otherwise, were made and why. As an aid to the reader, an additional section is provided that summarises the experimental findings that resulted from the publications. After that I will discuss how this PhD project, and its conceived metrics, can be linked to related fields and previously discussed research. I will end this thesis with some final remarks.

6.1 Chapter Overview

Chapter 2: Correlating Process and Product Data. As a first step onto the aforementioned road, a literature study was most required to understand the topic of translation difficulty. This paper was highly influenced by previous work. In fact, it was a pilot study to see if we could replicate correlations that were found in other studies, and answer the question How does translation process data correlate with the translation product?. There are two main differences with those previous studies, though. First, my interest at the time was the sentence level rather than the word level, even though most research involves the word as the unit of interest. I was likely biased by the Natural Language Processing (NLP) task of Sequence Classification: it is generally speaking easier to predict a value for a given sentence than for each individual word. In my mind, I figured that I would rather work towards a *good* sentence-level difficulty prediction system than a *worse* word-level one. In retrospect, it may have been worthwhile to start my investigation on the word level rather than working towards it in a top-down fashion. A second difference with the literature that I read, was that we decided to use correlations even though most related work relies on experimental methodology involving mixed-effects models. My experience with statistics was limited and I did not have the background to confidently use such models. In fact, I have been fortunate enough to have had coauthors that specifically contributed by means of building mixed models, for which I am incredibly grateful (dr. Joke Daems in Ch. 4 and dr. Moritz Schaeffer in Ch. 5). Using correlations between two variables rather than a feature-complete mixed model is a simplification, which disregards a number of factors such as inter-participant variability and the impact of individual fixed effects on the dependent variable. Still, a simplification or not, a correlation between two variables does tell us something about the relation between the two, and our question was answered: Significant correlations between process and product data exist (though small), particularly between syntactic equivalence and translation process features, which I took as a point of focus for the subsequent work.

Chapter 3: Predicting Syntactic Equivalence. The goal of this paper was to build a machine learning system that given a source sentence could predict the average word (group) reordering that needs to occur without having access to its translation. In theory that means that you infuse a machine learning system with abstract target language information. In a sense, the model learns which target word order corresponds with a given source word order to ultimately quantify the difference. Methodologically, this paper was of much interest to me as it increased my existing knowledge of machine learning. The neural systems were built by coauthor dr. Arda Tezcan, and were mostly based on his own previous work. At a later stage, after the experiments had already been concluded, I spent a lot of time delving into deep learning and I re-implemented the whole system from-scratch in a different deep learning library (PyTorch; Paszke et al., 2019). Around the same time, transfer learning with BERT (Devlin et al., 2019) and other Sesame-street characters (large Transformer-based neural models), became a popular choice in NLP research, and I integrated that in our system as well (BERT performed slightly better than our previously best system). Those results were never published as a paper, but they were presented on the MEMENTO workshop of 2019, collocated with the MT Summit conference. The research question How well can a machine learning system predict word (group) reordering by only making use of source text information? was optimistically answered: Word reordering and word group reordering can be modelled by a machine learning system with a Pearson r correlation of at least 0.54 and 0.58 respectively. At the end of this work, we started to think about different ways of capturing other syntactic differences between a source and target text in addition to word (group) reordering.

Chapter 4: Metrics of Syntactic Equivalence. In terms of the cognitive effort required by my coauthors and myself, this book chapter and the development that led up to it was probably the most (in)tense of all the publications. The first step was choosing Which fine-grained metrics can quantify syntactic divergences between a source and target text on the sentence level? We decided on changes in dependency labels to capture phenomena such as passivisation (a subject turns into an object), linguistically motivated word group reordering, and – most intricate of all – aligned syntactic tree edit distance, which compares the abstract, syntactic source and target structures while also taking word alignment information into account. From a development perspective, I made the decision to stick with *only* a sentence-level aggregation of the metrics, as a continuation of Chapter 3. The experimental results of this book chapter answer the question What is the effect of sentence-level, syntactic metrics on process data? We found that changes in dependency label have a significant, positive effect on coherent typing behaviour and that the amount of linguistic word group reordering positively and significantly affects the total reading time on the source sentence.

Chapter 5: The Effect of Product-based Metrics. The most recent and final work, which has been submitted to a journal and awaits review, is a logical follow-up study to the previous one: instead of only looking for sentence-level effects, smaller linguistic units, i.e. words, are investigated. Compared to Chapter 4, the balance of focus has shifted from a focus on a detailed explanation of the metrics to the experimental design and results, and rightly so. But although that may seem logical – they are the same metrics by name as in the previous study, so why bother explaining them again? – a lot of work needed to happen behind the scenes. As I wrote before, the metrics were initially implemented in such a way that only aggregated sentence-level values could be retrieved. For this paper, the measures needed to be available for each word in the sentence. The underlying idea is that a machine learning system would benefit from word-level features to automatically annotate sub-sentential translation difficulties. Because of that, the whole library was built again from the ground up. It is now easily extensible by advanced users while at the same time straightforward to use by non-technical researchers. In this paper we also answer the experimental question What is the effect of word-level, syntactic product-based metrics on process data? Among other things, we found that word (group) reordering plays a significant part in the difficulty of the translation process, and so do abstract structural differences between the source and target text. See the summary of experimental results in Section 6.2 for more. Ideally, these findings are confirmed by follow-up research with larger, more diverse data sets. What we intended to show, starting from Chapter 4, is that syntax cannot be constrained to a single metric and that different measures can be developed to measure separate aspects of syntax. Rather than having a single all-encompassing metric that can explain as much statistical variance as possible, we strove to investigate the effect of individual syntactic properties.

6.2 Summary of Experimental Results

This section synthesises the research results involving the proposed metrics. These metrics were elaborately discussed in the initial chapter (Sec. 1.4) and will not be repeated here in full. If the reader desires a refresher, the glossary describes the metrics succinctly as well, and also provides page references to important discussions concerning the specific concepts.

6.2.1 Label Changes

We suggested "label changes" as a straight-forward way to quantify whether or not a linguistic label (in our case a dependency label) differs between a source word and its translation (Vanroy et al., in press). We showed that, on the averaged sentence level, label changes had a highly significant effect on coherent typing behaviour (Kdur), indicating that many changes in dependency labels have a positive effect on how long translators type their translation (note that Kdur includes deletions, revisions, and any other keyboard activity). The effect of label changes on the word level was not further looked into in the last paper because it was economically infeasible to investigate different models (including contrastive ANOVAs) for *all* proposed metrics. The clear effect on the sentence level does invite further analysis, however.

6.2.2 `word_cross`

`word_cross` was initially suggested in Vanroy, Tezcan, and Macken (2019), where the aim was to predict the average amount of word reordering of a given translation solely based on source text information. The best-performing machine learning model used both semantic and morphosyntactic features as input for a recurrent neural network and achieved a Pearson r correlation of 0.54 (see Sec. 3.4). As discussed in the previous section, I later improved this model (and the model for `seq_cross`) further by using transfer learning and Transformer-based models. The conclusion is that average word reordering can be predicted reasonably well without access to the translation.

In Vanroy et al. (2021) word-level `word_cross` was used as a predictor for mixed-effect models. It showed a significant and positive effect on first fixation duration but the explained variance was very low. No significant effect was found for eye-key span, the time between the first fixation on a word and the first keystroke that contributes to the translation of that word. But for total reading time on the source text, `word_cross` showed a marginally significant, positive effect when including residual outliers. In short, the word reordering metric does seem to affect the required cognitive effort, i.e. if the word order in a text is distorted/reordered more, then this would have an effect on the translation process. Especially because of the marginal significance, more research is needed.

6.2.3 `seq_cross`

In conjunction with `word_cross`, `seq_cross` was introduced in Vanroy, Tezcan, and Macken (2019) with the goal of predicting the average word group reordering based on source text information only. The machine learning model could predict the reordering values with a Pearson r correlation of 0.58 (Sec. 3.4). That means that it was slightly better at predicting sequence reordering than word reordering.

In later work we showed that word group reordering also affects the translation process on the word level (Vanroy et al., 2021). The metric was slightly adapted to allow for multi-word groups (m-to-n alignments) to form single groups (Sec. 5.3.3). Particularly in late processes an effect was observed. Eye-key span was positively affected but only if the word_id predictor (a word's position in the sentence) was not included. More prominent, both in terms of significance and R^2 value, was the effect on the total number of fixations on a source word (total reading time). This effect is more clearly significant than the effect of `word_cross` – although indubitably there is a correlation or overlap. As I suggest in Section 5.5, this late effect may imply that translators work on units of increasing size and resolve issues per unit size. At first word-related issues are scrutinised and increasingly larger units, such as sequences/phrases, are resolved at a later stage. That would explain why the effect of `seq_cross` only becomes apparent in the later stages of the translation process. Similarly, Gile (1995) suggests that translators do not necessarily always work on the same type of unit.

6.2.4 SACr

SACr (syntactically aware cross) is a continuation of `seq_cross`. SACr refines the sequence groups so that the source and target groups of aligned groups are valid subtrees in their respective sentences rather than solely based on the happenstance of the word alignments (Vanroy et al., in press). On the sentence level (the number of SACr group reorderings averaged by the number of SACr alignments), SACr has a clear positive, significant effect on the total reading time on the source text (Sec. 4.5). This means that the more word group reordering (of linguistically motivated groups) are done, the more cognitive effort is required. Similar to `seq_cross` above, this is an effect on a *late* processing measure (total reading time), further lending support to the suggestion that larger units are processed at a later stage. SACr on the word level was not further investigated in the last paper but a comparison between `seq_cross` and SACr would be interesting to examine how important the *linguistic* aspect of word group movement is.

6.2.5 ASTrED

In an effort to quantify the linguistic structure of a source text and its translation, ASTrED compares their dependency trees and also considers word alignment information. We found an effect on the eye-key span that was positive and significant, and in fact was the most significant syntactic metric – only outperformed by entropy-based metrics with a semantic component (HTra and HSTC). In this case, ASTrED was used as a boolean value where it indicated whether a source word was matched in the target tree structure or whether structural changes needed to happen. If such changes were needed, and a tree transformation was required, the time between first encountering a word and producing its translation was significantly longer. This, again, suggests that structural changes do require more processing effort during translation, assuming that longer processing serves as a proxy for cognitive effort.

6.3 Relation to Related Fields

Near the end of Section 1.2.3 I explained that the metrics suggested in this thesis are intended to contribute to the field of translatability. Similar to how other metrics are used to look into specific text or translation-specific properties and problems, my proposed measures serve the same purpose. They allow researchers to investigate and control for fine-grained syntactic features. The impact of these translation-specific properties on the translation process (and so translation difficulty) has been discussed at length before. In this section the metrics are placed in a broader context, and similarities with related fields are discussed.

6.3.1 Machine Translation and Formal Language Theory

Section 2.2.3.1 refers to translation difficulty research in MT, but in this section I want to draw parallels between the introduced metrics themselves and specific parts of MT. I will solely focus on statistical machine translation. Not because neural machine translation (NMT) does not have plenty of research on the topic of syntax – which it does – but because it was already discussed in Section 3.2, particularly with a focus on word order and syntactically augmented neural networks. Here it should suffice to say that NMT is better at learning reordering procedures than SMT (Toral & Sánchez-Cartagena, 2017).

In phrase-based statistical MT (PBSMT; e.g. Koehn et al., 2003; Och & Ney, 2003), the goal of the system can be very broadly summarised as follows: find the most probable translation using a log-linear model consisting of phrase translation probabilities, phrase reordering probabilities or costs, and n-gram probabilities of the target language model. Using phrases (groups of words that are not necessarily linguistically consistent) rather than single words has some advantages, particularly related to context, word order and compound nouns (Och et al., 1999). Of particular interest to `seq_cross` is that phrases in PBSMT are not linguistically motivated and based on word alignment. PBSMT phrases can only exist if "the words within the source phrase are only aligned to words within the target phrase" (Och et al., 1999, p. 24). This is in fact identical to the Definition of Consecutiveness (Def. 1) to create sequence groups for `seq_cross`.

Because three of the proposed metrics deal with changes in word (group) order, a reference to lexical and phrasal reordering in MT is also fitting. Initially, PBSMT models were not great at displacing source text items in the target text because they did not consider the likelihood of the reordering of items and instead use an added "cost" (or penalty) for reordering, which is the same for all phrases and linear to the reordering distance. This is unnatural, of course, as some constructions or lexical items are more likely to need reordering depending on the context and language pair. The translation toolkit Moses (Koehn et al., 2007), for instance, by default has a relatively weak reordering model where the cost of reordering a phrase is calculated in a similar way as the aforementioned Cross metric in TPR research (Schaeffer & Carl, 2014), but on the phrase level.[1] For each phrase, the cost of reordering is calculated relative to the position of the translation of the previous phrase. This approach is therefore different than our cross-based metrics which rely on absolute reordering in relation to all surrounding items rather than only relative to the previous one. A number of suggestions have been proposed to increase the capabilities for long range displacement of phrases, which are often called conditional or lexicalized reordering models (Galley & Manning, 2008; Koehn et al., 2005) that incorporate more phrase-specific reordering probabilities in the log-linear model so that a more natural order can be achieved.

Hierarchical models (such as Chiang, 2007) make use of grammar rules (typically Synchronous Context Free Grammars; SCFG) rather than a direct mapping of source to target phrase. This allows for the "hierarchical" aspect where phrases contain other phrases, i.e. recursion. Such SCFGs are of the type "make use of $X_1 \rightarrow X_1$ gebruiken [X_1 TO-use]", where the phrase structure of the source language can be transposed into a target structure by

[1] See `http://www.statmt.org/moses/?n=Moses.Tutorial#ntoc9` and `http://www.statmt.org/moses/?n=FactoredTraining.BuildReorderingModel` for more information about reordering in Moses

means of a transfer rule.[2] In syntax-based SMT models, a type of hierarchical models, the non-terminals (like X_1 above), correspond to linguistic categories (e.g. NP). Some variants of syntactic models have been proposed depending on where the linguistic information is available. In tree-to-string models, the input is a parse tree, which can be manipulated (e.g. reordering, insertions) but the output is regular text, namely the translation of the leaf nodes of the final source tree (Collins et al., 2005; Yamada & Knight, 2001). Conversely, string-to-tree models start from an input source string and produce a CFG tree derivation and the translated text (Yamada & Knight, 2002). The observant reader may notice that this is related to ASTrED, which also tries to find ways to convert the source syntactic structure into the target structure *and quantify the difference*. Below follow a few research topics within SMT/FLT that are conceptually similar to ASTrED, and that are viable, alternative approaches to modelling syntactic divergences between two sentences, although perhaps not as easy to implement or to quantify.

Similar to Och et al. (2004), one could use the probability of a tree-to-tree alignment model, where both the input and the output contain structural, linguistic information, as a measure of how syntactically likely a translation is given its source text, which may shed a light on its difficulty – following the same thought process for the elaborately discussed entropy-based metrics of Carl and colleagues (Sec. 5.3.3). Alternatively, rather than sticking with SCFGs, one could use fully tree-based grammars such as (synchronous) tree-adjoining grammars (TAG; Joshi et al., 1975) or in the least tree substitution grammars. Instead of having grammar rules that only consist of symbols (on the left side), tree grammars allow the rewriting of (nodes of) a tree as another tree. TAG allows two notable tree operations, namely substitution and adjunction. These concepts are hard to illustrate without providing elaborate examples, so instead I refer the curious reader to the aforementioned article on TAG.[3] In brief, substitution allows the replacement of a node in a given tree by another tree, provided that the non-terminal node that is being substituted is the same as the root node of the substituting tree. Adjunction is, theoretically, more complex. It can insert a full tree in another tree, irrespective of its place in the tree. (Synchronous) tree substitution grammar (STSG), then, is similar but does not have the adjoining operation. Of particular relevance is the work of Hajič et al. (2004) who incorporate (an extension to) TSGs in MT to try to improve the syntax of the target text. This brief overview should make clear that the work in FLT is relevant to syntactic translatability, particularly the work on synchronous grammars, because it exactly involves comparing two structures with each other.

The paragraph above is quite dense as it is and yet only touches on the surface of research in this direction. For the reader it is most important to

[2] See the Moses documentation for more examples `http://www.statmt.org/moses/?n=Moses.SyntaxTutorial`

[3] `http://www.let.rug.nl/~vannoord/papers/diss/diss/node59.html` provides an approachable overview as well

understand that in the field of MT and Formal Language Theory, a lot of work has been done to compare structures. Some of that work served as an inspiration for my research, and others may be useful in their own right when applied to translation studies. ASTrED makes use of a preprocessing step before calculating tree edit distance to make sure that word-aligned tokens match in the calculated optimal path. It therefore does not rely on grammar rules to derive the target tree from the source tree.

6.3.2 Equivalence and (Shared) Syntax in the Mind

Although this thesis relies on underlying computational approaches, "[u]sing computational methods to explore translational phenomena does not commit one to a computational theory of mind" (Alves & Jakobsen, 2020, p. 8). It is therefore worthwhile to briefly reconnect to Section 1.2.2. Most of those models discuss bilingual models in term of the lexicon, though, which makes it difficult to relate my results and syntactic metrics to them. Still, one theory to reflect back on here would be the shared-syntax account of Hartsuiker et al. (2004) and cross-linguistic priming. First, a short detour needs to be made to equivalence and specifically the shifts of Catford (1965) should be reiterated.

The work accompanying this thesis provides new ways to quantify the syntactic equivalence between a source sentence and its translation. The operationalisation of equivalence in that manner can be linked to the categorical shifts of Catford (1965), discussed before in Section 1.2.1. Structure-shifts can occur on different linguistic levels and include shifts in surface and deep structure. Therefore, they can be measured with the word (group) reordering metrics `word_cross`, `seq_cross` and SACr and with the structural comparison metric ASTrED. Class-shifts are those shifts where the function of a word has changed in a phrase, clause or sentence. Those can be measured with the metric called "label changes".

The shared-syntax account posits that "[language] rules that are the same in the two languages are represented once" (Hartsuiker et al., 2004, p. 409), and supports earlier findings by Loebell and Bock (2003). Such rules are for instance the creation of grammatical constructions. In this research in bilingualism, syntax was understood as sentence type (passive, active, intransitive, or object-verb-subject order) and the experiment involved Spanish-to-English transfer. In follow-up research Hartsuiker et al. (2016) confirmed these results with more languages (English, Dutch, French, German) and different syntactic structures (relative clause attachment with Dutch, French and English target, dative constructions with English target).

Interestingly, Bernolet et al. (2007) found that such cross-linguistic priming effects were reliant on characteristics of the languages themselves. Specifically, they investigated priming constructions in Dutch, English and German of the type "the red shark" (AN; which is grammatical in all languages involved), and similar constructions with a relative clause (RC). In the latter case, there is no one-on-one correspondence between the languages: Dutch and German require that the adjective is placed before the verb in the relative clause, and in English it must be placed after ("the shark that is red"). Priming effects were observed for those cases where the word order of the construction was the same (i.e. AN for all languages and RC for Dutch and German). So, as an example, a Dutch RC construction does not prime an English RC construction (which has the adjective after the verb) but it does prime the German RC. The conclusion here is that syntactic structures that do not share the same word order do not share a representation in the mind.

Our suggested metrics can be used in a similar fashion as word order was used in Bernolet et al. (2007), namely as a means to verify conditions in which priming may or may not hold. A natural question in light of this thesis would be whether the size of the difference in word order plays a role rather than only a binary distinction "different or identical word order". Formally: are constructions less likely to prime in correspondence to a scale of word order differences (as measured, for instance by `word_cross`)? In addition, it would be interesting to investigate whether the same is true for deeper syntactic differences, but where the word order is maintained. Specifically, syntactic trees may be the same even if their word order differs (due to unordered parent-child relations), and similarly sentences with a different word order can still have identical underlying structures. An interesting topic to investigate in this respect would be verb clusters, whose order is relatively free in Dutch (and yet the underlying syntactic structures are the same) but not so much in English. Formally: do sentences with identical trees (those with an ASTrED value of 0) but different word orders impact priming (and vice versa)? It is relevant here to again refer to the priming study by Jacob et al. (2017) who suggest that (differences in) hierarchical syntactic trees can be used to explain priming effects, although in their study they use phrase trees rather than dependency trees and do not quantify the differences in trees.

In addition to priming studies in bilinguals, work linked to the translation process finds that ambiguity in target realisations may disrupt the translation process (Tokowicz & Kroll, 2007). Ambiguity, here, is restricted to semantics. Alterations to existing lexical models of the bilingual mind (cf. Sec. 1.2.2) have been suggested to account for such ambiguity, particularly the Revised Hierarchical Model of Translation Ambiguity (Eddington & Tokowicz, 2013), which predicts that translators work more slowly when having access to multiple translations of a source word. Such ambiguities could be important to take into account when setting up experiments and may impact results when left unchecked. As Prior et al. (2007) write "ambiguity must be carefully considered and controlled in the construction of experimental materials" (p. 1035). This ties in with the effects of word translation entropy that have been discussed frequently before (HTra) where a source word is translated differently by translators. On a syntactic level, variance in word reordering (HCross) has already been shown to affect eye-key span (Schaeffer & Carl, 2017). Another operationalisation of syntactic entropy (HSyn; Bangalore et al., 2015) uses manually annotated properties of the target text (clause type, verb valency, voice) as the subject of entropy and report a positive effect on translation process measures, also indicating that statistical variance in different syntactic options causes difficulties. It would be worthwhile to inspect whether the same is true for our fine-grained metrics as well. Does a multitude of structural target realisations for instance slow down the translator? Do multiple possibilities in terms of label changes affect the translation process? This was not done in this thesis because our intent was to create translation-specific features that can be used in a machine learning system by leveraging large parallel corpora. Because entropy-based approaches require many translation alternatives and such large parallel corpora rarely contain more than one translation per sentence, such methods were not our priority. Nevertheless, entropy-wrapped versions of our metrics can be used to investigate the effect of translation-specific syntactic variation in more detail.

Of course the research questions above require more thought and an experimental approach would need careful curation and further literature study. The important take-away is that many models of the mind have focused on the lexicon, and that those that consider syntax often restrict themselves to language (group) specific constructions (e.g. relative clauses, dative realisations). *Complementary* to that, the metrics introduced here can allow for a more general approach to involve syntax, both in psycholinguistics and translation studies, and both on shallow and deep linguistic levels.

6.4 Concluding Remarks

In this research project, I have contributed to the topic of translation difficulty prediction together with my coauthors in four ways. First, we have delivered experimental studies that investigate difficulties in English-to-Dutch from-scratch translation and showed that syntactic properties of a source text compared to its translation have an effect on translation process features that can be used as proxies for cognitive effort, both on the word and sentence level. Thereby confirming that diverging syntactic properties between a source and target unit cause increased translation difficulty. In addition, we have also created a high quality parallel corpus that supplements the existing multiLing study with English-to-Dutch translations. These translations were manually tokenised and aligned. Unfortunately, due to the global pandemic, we were not able to record eye-tracking information for the study but keylogging data has been included. Even though we did not use the data ourselves, it is publicly available and can be used by other researchers, too. We also built a machine learning system to model the word (group) reordering that is required to transform the source order to the target order. It is capable of predicting syntactic properties of a source text with respect to its translation without having access to the translation itself. These syntactic measures of word (group) reordering have an effect on the translation process (as we have shown), so being able to predict them is a step in the right direction to create a full-blown translatability prediction system. Lastly, we have developed a new set of syntactic metrics and made them available to the public. These metrics make use of Universal Dependencies, which means that at least 66 languages can easily be used with the tool by using `stanza` (Qi et al., 2020) behind the scenes for automatic parsing. My hope is that this open-source implementation will have a positive impact on the field, and encourages fellow researchers to use a more divergent set of syntactic metrics in their experiments – and perhaps even develop new ones.

As discussed in the conclusion of Chapter 5, our findings and developments have paved the way for a cross-roads of prospective research questions. We raised the question whether individual syntactic metrics have different effects on the translation process and showed that they do. These results should be confirmed and extended by additional research, however. Controlled experiments involving manually curated stimuli should be presented to translators to investigate our preliminary findings. After all, our approach was, what you may call, corpus-based: we made use of a set of translations of given texts, but without any controls or specific syntactic phenomena to investigate. Although such an approach yields relevant results from approximately "real-world" translation, it is likely to contain more noise than in a controlled setting due to competing syntactic phenomena and difficulties. In addition, and conversely perhaps, more corpus-based studies are recommended to confirm our results with larger, more varied data sets in different languages. Another direction can be taken as well, into the realm of computational translation studies to continue working on a translatability prediction system. It is clear that a variety of syntactic phenomena contribute to translation difficulty, so a system can be built that makes use of those features. Finally, one could create a multitude of different linguistic metrics that model other aspects of language than we suggested. We by no means intend to claim that our metrics are exhaustive in terms of modelling syntactic differences between a source text and its translation, and in syntax but also semantics, a lot more work can be done. What is more, instead of limiting ourselves to the word or sentence level, investigating discourse-focused properties of the whole text would be a challenging but fascinating endeavour, too.

Even though we are arriving at the end of this thesis, and its related PhD project, it is by no means the conclusion of our efforts in translatability. The fruitful collaboration that bloomed between my own research group LT^3 and TRA&CO in our last paper in Chapter 5 will be maintained and hopefully expanded in the future. In addition, many related research ideas, such as the ones above, have been discussed between me and my colleagues that could lead to exciting, future projects. The field of translatability prediction is relatively young and still quite small, but that only means that it has many secrets left for us to uncover.

Appendix A

Proof for cross value of MWGs

The equation for cross calculated on multi-word groups was given in Equation 1.3 and reproduced below.

$$cross_{MWG} = \frac{1}{4} \cdot mn(m-1)(n-1)$$

where:

m	number of words on the source side of the MWG
n	number of words on the target side of the MWG

This can be proven as follows. There are only two scenarios in which an alignment link (k, l) crosses any another alignment link (i, j), as was given in Equation 1.2.

$$cross((k,l),(i,j)) = \begin{cases} 1, & \text{if } i < k \ \ \& \ \ j > l \\ & \text{or } k < i \ \ \& \ \ l > j \\ 0, & \text{otherwise} \end{cases}$$

where:

k	source index of the first alignment link
l	target index of the first alignment link
i	source index of the second alignment link
j	target index of the second alignment link

Given m source and n target tokens, these conditions can be generalised for an alignment link (k, l).

$i < k \ \ \& \ \ j > l :$	$(k-1)(n-l)$
$k < i \ \ \& \ \ l > j :$	$(m-k)(l-1)$

I.e., in a multi-word group, the (k, l) alignment link crosses any other (i, j) link if the conditions are met. The crosses of (k, l) with alignments (i, j) where $i < k \ \ \& \ \ j > l$ can be written in function of n target words: $(k-1)(n-l)$. The crosses of (k, l) with alignments (i, j) where $k < i \ \ \& \ \ l > j$ can be written in function of m source words: $(m-k)(l-1)$.

The total number of crosses in the MWG is then a sum of these conditions for all possible alignment links (all combinations of l and k). However, as should be evident, this will count each cross twice (once for each alignment link involved), which has to be taken into account.

$$\begin{aligned}
&\frac{1}{2}\cdot\sum_{l=1}^{n}\sum_{k=1}^{m}\Big((k-1)(n-l)+(m-k)(l-1)\Big)\\
=&\frac{1}{2}\cdot\sum_{l=1}^{n}\sum_{k=1}^{m}\Big(k(n-l)-k(l-1)+m(l-1)-(n-l)\Big)\\
=&\frac{1}{2}\cdot\sum_{l=1}^{n}\left(\sum_{k=1}^{m}\Big(k(n-2l+1)\Big)+\sum_{k=1}^{m}\Big(ml-m-n+l\Big)\right)
\end{aligned}$$

The internal sums can be expanded separately. Assume the first part is B and the second A.

$$\begin{aligned}
A=&\sum_{k=1}^{m}\big(ml-m-n+l\big)\\
=&m(ml-m-n+l)\\
=&m\big(l(m+1)-(m+n)\big)\\
=&ml(m+1)-m(m+n)
\end{aligned}$$

$$\begin{aligned}
B=&\sum_{k=1}^{m}\Big(k(n-2l+1)\Big)\\
=&\frac{m}{2}\cdot\Big((n-2l+1)+m(n-2l+1)\Big)\\
=&\frac{m}{2}\cdot(n-2l+1+mn-2lm+m)\\
=&\frac{m}{2}\cdot\Big(-2l(m+1)+m(n+1)+(n+1)\Big)\\
=&\frac{m}{2}\cdot\Big(-2l(m+1)+(m+1)(n+1)\Big)\\
=&-ml(m+1)+\frac{m}{2}\cdot(m+1)(n+1)
\end{aligned}$$

A and B can then be reintegrated in the original formula.

$$\begin{aligned}
&\frac{1}{2}\cdot\sum_{l=1}^{n}\big(B+A\big)\\
=&\frac{1}{2}\cdot\sum_{l=1}^{n}\Big(-ml(m+1)+\frac{m}{2}(m+1)(n+1)+ml(m+1)-m(m+n)\Big)
\end{aligned}$$

At this point l can be discarded ($-ml(m+1)$ is nullified by $ml(m+1)$). Because there is no l in the equation, the summation happens on n. The equation can then be further simplified to ultimately be formulated as Equation 1.3.

$$
\begin{aligned}
& \frac{1}{2} \cdot \sum_{l=1}^{n} \left(m \Big(\frac{(m+1)(n+1)}{2} - (m+n) \Big) \right) \\
&= \frac{1}{2} \cdot nm \Big(\frac{mn+m+n+1-2m-2n}{2} \Big) \\
&= \frac{1}{4} \cdot nm(mn-m-n+1) \\
&= \frac{1}{4} \cdot nm\big(m(n-1)-(n-1)\big) \\
= cross_{MWG} &= \frac{1}{4} \cdot nm(m-1)(n-1)
\end{aligned}
$$

List of figures

List of tables

Glossary

`seq_cross` An extension to `word_cross` that creates phrases (non-linguistic word groups) based on the initial word alignments. The alignments between the generated phrases are then used to calculate word group reordering. Suggested in Vanroy, Tezcan, and Macken (2019) and further extended in Vanroy et al. (2021) where m-to-n alignments are considered as valid group alignments. 68, 84, 130, 149, 151, 153

`word_cross` An alternative version to Cross to calculate word reordering. It is absolute, that is, the reordering of a word is calculated with respect to its own reordering as well as the reordering of all the other words in the sentence. Suggested in Vanroy, Tezcan, and Macken (2019). 24, 67, 84, 130, 148, 153

ASTrED Aligned syntactic tree edit distance. A means to calculate tree edit distance between two aligned structures. The goal is to make sure that only aligned items can match each other in the tree and that other structural differences are then quantified. Proposed in Vanroy et al. (in press). 35, 95, 132, 150, 152, 153

Cross A word reordering metric as defined by Schaeffer and Carl (2014). For each word, the reordering of its translation *relative to the translation of the previous word* is calculated. 17, 24, 54, 67, 82, 121

eye-key span The time between the first fixation on a source word and the first keystroke that contributes to the translation of that word as suggested by Dragsted (2010); Dragsted and Hansen (2008). 17, 125, 149, 154

HSTC "joint source-target alignment / translation distortion entropy", an entropy-based metric that incorporates both lexicosemantic and syntactic information using word-group information. Proposed by Carl (in press). 17, 129, 150

HTra Word translation entropy (HTra) is a metric to quantify the agreement or uncertainty of lexical choice. Given a set of translations for a specific word, one can calculate the probabilities of each translation option and hence the entropy for that word. The lower the entropy, the more certain the choice or the higher the agreement between translators. First suggested by Carl and Schaeffer (2014). 16, 54, 82, 128, 150, 154

Kdur Duration of coherent typing behaviour, i.e. the total duration of coherent keyboard activity excluding keystroke pauses of more than five seconds. 111, 148

label changes An intuitive metric to see whether a linguistic label of a word (e.g. dependency label or POS tag) is different from the word that it is aligned with. Proposed in Vanroy et al. (in press) where it involves dependency labels. 22, 93, 148, 153

SACr Syntactically aware cross. An extension to `seq_cross` that refines the word groups of `seq_cross` to ensure that they are linguistically motivated, i.e. that each group constitutes a valid subtree in its respective sentence. Proposed in Vanroy et al. (in press). 29, 90, 149, 153

total reading time Total reading time on the source or target, i.e. the sum of all fixations. This can be calculated on the segment level as well as on the word level, on the source side and the target side. 110, 125, 149

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous distributed systems.*

Aharoni, R., & Goldberg, Y. (2017, July). Towards string-to-tree neural machine translation. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 132–140). Vancouver, Canada: Association for Computational Linguistics. doi: 10.18653/v1/P17-2021

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. Petrov & F. Caski (Eds.), *Proceeding of the Second International Symposium on Information Theory* (pp. 267–281). Budapest: Akademiai Kiado.

Akbari, A., & Segers, W. (2017). Translation difficulty: How to measure and what to measure. *Lebende Sprachen*, *62*(1). doi: 10.1515/les-2017-0002

Alabau, V., Bonk, R., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., … Tsoukala, C. (2013). CASMACAT: An open source workbench for advanced computer aided translation. *The Prague Bulletin of Mathematical Linguistics*, *100*(1). doi: 10.2478/pralin-2013-0016

Alves, F., & Jakobsen, A. L. (Eds.). (2020). *The Routledge handbook of translation and cognition* (First ed.). New York: Taylor and Francis.

Alves, F., Pagano, A., Neumann, S., Steiner, E., & Hansen-Schirra, S. (2010). Translation units and grammatical shifts: Towards an integration of product- and process-based translation research. In G. M. Shreve & E. Angelone (Eds.), *American Translators Association Scholarly Monograph Series* (Vol. XV, pp. 109–142). Amsterdam: John Benjamins Publishing Company. doi: 10.1075/ata.xv.07alv

Alves, F., & Vale, D. (2009, December). Probing the unit of translation in time: Aspects of the design and development of a web application for storing, annotating, and querying translation process data. *Across Languages and Cultures*, *10*(2), 251–273. doi: 10.1556/Acr.10.2009.2.5

Andersen, P. (1990). How close can we get to the ideal of simple transfer in multilingual machine translation (MT). In *Proceedings of the 7th Nordic Conference of Computational Linguistics (NODALIDA 1989)* (pp. 103–113). Reykjavík, Iceland: Institute of Lexicography, Institute of Linguistics, University of Iceland, Iceland.

Asadi, P., & Séguinot, C. (2005). Shortcuts, strategies and general patterns in a process study of nine professionals. *Meta: Journal des Traducteurs/Meta: Translators' Journal*, *50*(2), 522–547.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR 2015.*

Baker, M. (2011). *In other words: A coursebook on translation* (Second ed.). Abingdon, UK: Routledge.

Balling, L. W., Hvelplund, K. T., & Sjørup, A. C. (2014, November). Evidence of parallel processing during translation. *Meta: Journal des Traducteurs/Meta: Translators' Journal*, *59*(2), 234–259. doi: 10.7202/1027474ar

Balota, D. A., Yap, M. J., Hutchison, K. A., Cortese, M. J., Kessler, B., Loftis, B., ... Treiman, R. (2007, August). The English Lexicon Project. *Behavior Research Methods*, *39*(3), 445–459. doi: 10.3758/BF03193014

Bangalore, S., Behrens, B., Carl, M., Gankhot, M., Heilmann, A., Nitzke, J., ... Sturm, A. (2015). The role of syntactic variation in translation and post-editing. *Translation Spaces*, *4*(1), 119–144. doi: 10.1075/ts.4.1.06sch

Bangalore, S., Behrens, B., Carl, M., Ghankot, M., Heilmann, A., Nitzke, J., ... Sturm, A. (2016). Syntactic variance and priming effects in translation. In M. Carl, S. Bangalore, & M. Schaeffer (Eds.), *New directions in empirical translation process research* (pp. 211–238). Cham: Springer International Publishing. doi: 10.1007/978-3-319-20358-4_10

Barone, A. V. M., & Attardi, G. (2013). Pre-reordering for machine translation using transition-based walks on dependency parse trees. In *Proceedings of the eighth workshop on statistical machine translation* (pp. 164–169). Sofia, Bulgaria: Association for computational linguistics.

Bartoń, K. (2009). *MuMIn: Multi-modal inference.* Retrieved from `https://cran.r-project.org/package=MuMIn`

Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., & Sima'an, K. (2017, September). Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1957–1967). Copenhagen, Denmark: Association for Computational Linguistics. doi: 10.18653/v1/D17-1209

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. doi: 10.18637/jss.v067.i01

Bell, R. T. (1998). Psycholinguistic/cognitive approaches. In M. Baker (Ed.), *Routledge Encyclopedia of Translation Studies* (First ed., pp. 185–190). London: Routledge.

Benjamin, R. G. (2012). Reconstructing readability: Recent developments and recommendations in the analysis of text difficulty. *Educational Psychology Review*, *24*(1), 63–88. doi: 10.1007/s10648-011-9181-8

Bentivogli, L., Bisazza, A., Cettolo, M., & Federico, M. (2016, nov). Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 257–267). Austin, Texas: Association for Computational Linguistics. doi: 10.18653/v1/D16-1025

Bernolet, S., Hartsuiker, R. J., & Pickering, M. J. (2007). Shared syntactic representations in bilinguals: Evidence for the role of word-order repetition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *33*(5), 931–949. doi: 10.1037/0278-7393.33.5.931

Bernth, A., & Gdaniec, C. (2001). MTranslatability. *Machine Translation*, *16*, 175–218.

Birch, A., Osborne, M., & Koehn, P. (2008). Predicting success in machine translation. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP 2008)* (pp. 745–754). Honolulu, Hawaii: Association for computational linguistics. doi: 10.3115/1613715.1613809

Bird, S., Loper, E., & Klein, E. (2009). *Natural language processing with Python.* O'Reily Media Inc.

Bock, J. K. (1986, July). Syntactic persistence in language production. *Cognitive*

Psychology, *18*(3), 355–387. doi: 10.1016/0010-0285(86)90004-6

Bod, R. (1998). *Beyond grammar: an experience-based theory of language.* Stanford, Calif: Center for the Study of Language and Information, [Stanford University].

Borrillo, J. M. (2000). Register Analysis in Literary Translation: A Functional Approach. *Babel*, *46*(1), 1–19. doi: 10.1075/babel.46.1.02bor

Campbell, S. (1998). *Translation into the second language.* Boston, USA: Addison Wesley Longman Limited.

Campbell, S. (1999). A cognitive approach to source text difficulty in translation. *Target*, *11*(1), 33–63. doi: 10.1075/target.11.1.03cam

Campbell, S. (2000). Choice network analysis in translation research. In M. Olohan (Ed.), *Intercultural faultlines: Research models in translation studies* (pp. 29–42). Manchester, UK: St. Jerome.

Campbell, S., & Hale, S. (1999). What makes a text difficult to translate? In *Proceedings of the 1998 ALAA congress.* Nathan, Queensland, Australia.

Carl, M. (2012). Translog-II: A program for recording user activity data for empirical reading and writing research. In *Proceedings of the eighth international conference on language resources and evaluation* (pp. 4108–4113). Istanbul, Turkey.

Carl, M. (in press). Information and entropy measures of rendered literal translation. In M. Carl (Ed.), *Explorations in empirical translation process research.* Springer International Publishing.

Carl, M., & Dragsted, B. (2012). Inside the Monitor Model: Processes of Default and Challenged Translation Production. *Translation: Computation, Corpora, Cognition*, *2*(1), 127–145.

Carl, M., Jakobsen, A. L., & Jensen, K. T. H. (2008). Studying human translation behavior with user-activity data. In *Proceedings of the 5th international workshop on natural language processing and cognitive science* (pp. 114–123). Barcelona, Spain. doi: 10.5220/0001744601140123

Carl, M., & Kay, M. (2011). Gazing and typing activities during translation: A comparative study of translation units of professional and student translators. *Meta: Journal des Traducteurs/Meta: Translators' Journal*, *56*(4), 952–975. doi: 10.7202/1011262ar

Carl, M., Kay, M., & Jensen, K. T. H. (2010). Long distance revisions in drafting and post-editing. In *Proceedings of CICLing 2010* (pp. 193–204). Iaşi, Romania.

Carl, M., & Schaeffer, M. (2014). Word transition entropy as an indicator for expected machine translation quality. In K. J. Miller, L. Specia, K. Harris, & S. Bailey (Eds.), *Proceedings of the Workshop on Automatic and Manual Metrics for Operational Translation Evaluation* (pp. 45–50). Reykjavik, Iceland: European Language Resources Association.

Carl, M., & Schaeffer, M. J. (2017). Why translation is difficult: A corpus-based study of non-literality in post-editing and from-scratch translation. *HERMES - Journal of Language and Communication in Business*(56), 43–57. doi: 10.7146/hjlcb.v0i56.97201

Carl, M., Schaeffer, M. J., & Bangalore, S. (2016). The CRITT translation process research database. In M. Carl, S. Bangalore, & M. J. Schaeffer (Eds.), *New directions in empirical translation process research* (pp. 13–54). Cham, Switzerland: Springer.

Carl, M., Tonge, A., & Lacruz, I. (2019). A systems theory perspective on the translation process. *Translation, Cognition & Behavior*, *2*(2), 211–232. doi: 10.1075/tcb.00026.car

Castilho, S., & O'Brien, S. (2017). Acceptability of machine-translated content: A multi-language evaluation by translators and end-users. *Linguistica Antverpiensia*, *16*, 120–136.

Catford, J. C. (1965). *A linguistic theory of translation: An essay in applied linguistics.* Oxford University Press.

Chen, K.-h., & Chen, H.-H. (1995). Machine translation: An integrated approach. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation* (pp. 287–294). Leuven, Belgium.

Chesterman, A. (2011). Reflections on the literal translation hypothesis. In C. Alvstad, A. Hild, & E. Tiselius (Eds.), *Methods and Strategies of Process Research Integrative Approaches to Translation Studies* (Vol. 94, pp. 23–35). Amsterdam ; Philadelphia: John Benjamins Publishing Company.

Chiang, D. (2007, June). Hierarchical phrase-based translation. *Computational Linguistics*, *33*(2), 201–228. doi: 10.1162/coli.2007.33.2.201

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014, oct). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics. doi: 10.3115/v1/D14-1179

Chollet, F. (2015). *Keras.*

Chomsky, N. (1963). Formal properties of grammars. In R. D. Luce, R. R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology* (Vol. 2, pp. 323–418). New York, USA: John Wiley And Sons, Inc.

Christoffels, I. K., & de Groot, A. M. B. (2005). Simultaneous interpreting: A cognitive perspective. In J. F. Kroll & A. M. B. de Groot (Eds.), *Handbook of Bilingualism: Psycholinguistic Approaches* (pp. 454–479). New York, USA: Oxford University Press.

Collins, M., Koehn, P., & Kučerová, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics (ACL 2005)* (pp. 531–540). Ann Arbor, Michigan: Association for computational linguistics. doi: 10.3115/1219840.1219906

Collins-Thompson, K. (2014). Computational Assessment of Text Readability: A Survey of Current and Future Research. *International Journal of Applied Linguistics*, *165*(2), 97–135. doi: 10.1075/itl.165.2.01col

Collins-Thompson, K., & Callan, J. (2005). Predicting Reading Difficulty with Statistical Language Models. *Journal of the American Society for Information Science and Technology*, *56*(13), 1448–1462. doi: 10.1002/asi.20243

Currey, A., & Heafield, K. (2018). Multi-source syntactic neural machine translation. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2961–2966).

Daems, J. (2016). *A translation robot for each translator* (PhD thesis). Ghent University, Ghent, Belgium.

Daems, J., Macken, L., & Vandepitte, S. (2013). Quality as the Sum of its Parts: A Two-Step Approach for the Identification of Translation Problems and Translation Quality Assessment for HT and MT+PE. In S. O'Brien, M. Simard, & S. Lucia (Eds.), *Proceedings of MTS 2013 Workshop on Post-Editing Technology and Practice* (pp. 63–71). Nice, France.

Daems, J., Vandepitte, S., Hartsuiker, R. J., & Macken, L. (2015). The Impact of Machine Translation Error Types on Post-Editing Effort Indicators. In *Proceedings of MTS 2015 Workshop on Post-Editing Technology and Practice* (pp. 31–45). Miami, Florida, USA.

Daems, J., Vandepitte, S., Hartsuiker, R. J., & Macken, L. (2017). Translation Methods and Experience: A Comparative Analysis of Human Translation and Post-editing with Students and Professional Translators. *Meta: Journal des Traducteurs/Meta: Translators' Journal*, *62*(2), 245–270. doi: 10.7202/1041023ar

de Bot, K., & Schreuder, R. (1993). Word Production and the Bilingual Lexicon. In R. Schreuder & B. Weltens (Eds.), *The Bilingual Lexicon* (pp. 191–214). Amsterdam: John Benjamins Publishing Company.

de Groot, A. M. B. (1997). The cognitive study of translation and interpretation: Three approaches. In J. H. Danks, G. M. Shreve, S. B. Fountain, & M. K. McBeath (Eds.), *Cognitive processes in translation and interpretation* (pp. 25–56). Thousand Oaks CA: Sage Publications.

de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., & Manning, C. D. (2014, May). Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (pp. 4585–4592). Reykjavik, Iceland: European Language Resources Association (ELRA).

de Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation* (pp. 449–454). Genoa, Italy: European Language Resources Association (ELRA).

de Marneffe, M.-C., & Manning, C. D. (2008, aug). The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on cross-framework and cross-domain parser evaluation* (pp. 1–8). Manchester, UK: Coling 2008 Organizing Committee.

de Pedro, R. (1999). The translatability of texts: A historical overview. *Meta: Journal des Traducteurs/Meta: Translators' Journal*, *44*(4), 546–559. doi: 10.7202/003808ar

De Clercq, O., & Hoste, V. (2016). All Mixed Up? Finding the Optimal Feature Set for General Readability Prediction and Its Application to English and Dutch. *Computational Linguistics*, *42*(3), 457–490. doi: 10.1162/COLI_a_00255

De Clercq, O., Hoste, V., Desmet, B., Van Oosten, P., De Cock, M., & Macken, L. (2014). Using the Crowd for Readability Prediction. *Natural Language Engineering*, *20*(3), 293–325. doi: 10.1017/S1351324912000344

Deng, Y., Cheng, S., Lu, J., Song, K., Wang, J., Wu, S., ... Chen, B. (2018, oct). Alibaba's neural machine translation systems for WMT18. In *Proceedings of the third conference on machine translation: Shared task papers* (pp. 368–376). Belgium, Brussels: Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, may). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*.

Dou, Z.-Y., & Neubig, G. (2021). Word alignment by fine-tuning embeddings on parallel corpora. In *Conference of the european chapter of the association for computational linguistics (eacl)*.

Dragsted, B. (2005). Segmentation in translation: Differences across levels of expertise and difficulty. *Target*, *17*(1), 49–70. doi: 10.1075/target.17.1.04dra

Dragsted, B. (2010). Coordination of Reading and Writing Processes in Translation:

An Eye on Uncharted Territory. In G. M. Shreve & E. Angelone (Eds.), *Translation and Cognition* (Vol. 15, pp. 41–62). Amsterdam, The Netherlands: John Benjamins Publishing Company.

Dragsted, B. (2012). Indicators of Difficulty in Translation — Correlating Product and Process Data. *Across Languages and Cultures*, *13*(1), 81–98. doi: 10.1556/Acr.13.2012.1.5

Dragsted, B., & Hansen, I. (2009). Exploring translation and interpreting hybrids. The case of sight translation. *Meta: Journal des Traducteurs/Meta: Translators' Journal*, *54*(3), 588–604. doi: 10.7202/038317ar

Dragsted, B., & Hansen, I. G. (2008). Comprehension and production in translation: A pilot study on segmentation and the coordination of reading and writing processes. In S. Göpferich, A. L. Jakobsen, & I. Mees (Eds.), *Looking at eyes: Eye-tracking studies of reading and translation processing* (Vol. 36, pp. 9–29). Frederiksberg: Samfundslitteratur.

Du, J., & Way, A. (2017). Pre-reordering for neural machine translation: Helpful or harmful? *The Prague bulletin of mathematical linguistics*, *108*(1), 171–182. doi: 10.1515/pralin-2017-0018

DuBay, W. (2004). *The Principles of Readability.* Costa Mesa, CA: Impact Information.

Dyer, C., Chahuneau, V., & Smith, N. A. (2013). A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of NAACL-HLT 2013* (pp. 644–648). Atlanta, Georgia, USA: Association for Computational Linguistics.

Eddington, C. M., & Tokowicz, N. (2013, April). Examining English–German translation ambiguity using primed translation recognition. *Bilingualism: Language and Cognition*, *16*(2), 442–457. doi: 10.1017/S1366728912000387

Englund Dimitrova, B. (2005). *Expertise and explicitation in the translation process* (Vol. 64). Amsterdam ; Philadelphia: John Benjamins Publishing Company.

Eriguchi, A., Hashimoto, K., & Tsuruoka, Y. (2016, August). Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 823–833). Berlin, Germany: Association for Computational Linguistics. doi: 10.18653/v1/P16-1078

Ervin, S., & Bower, R. T. (1952). Translation problems in international surveys. *The Public Opinion Quarterly*, *16*(4), 595–604.

Fox, J. (2003, July). Effect Displays in R for Generalised Linear Models. *Journal of Statistical Software*, *8*(1), 1–27. doi: 10.18637/jss.v008.i15

Fox, J., & Weisberg, S. (2019). *An R companion to applied regression* (3rd ed.). Thousand Oaks CA: Sage.

Francois, T., & Miltsakaki, E. (2012). Do NLP and Machine Learning Improve Traditional Readability Formulas? In *Proceedings of the Workshop on Predicting and Improving Text Readability (PITR 2012)* (pp. 49–57). Montréal, Québec, Canada.

Frank, A. (2011). *Diagnosing Collinearity in Mixed Models from Lme4, Vif.mer Function.* Retrieved from `https://github.com/aufrank/R-hacks/blob/b0ddb70c7204e48c29139b6a3400da068b03076a/mer-utils.R`

Galley, M., & Manning, C. D. (2008, October). A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (pp. 848–856). Honolulu, Hawaii: Association for Computational Linguistics.

Gathercole, S., & Baddeley, A. (1993). *Working Memory and Language.* Hove: Lawrence Erlbaum.

Germann, U. (2008, June). Yawat: Yet Another Word Alignment Tool. In *Proceedings of the ACL-08: HLT Demo Session* (pp. 20–23). Columbus, Ohio: Association for Computational Linguistics.

Gile, D. (1995). *Basic concepts and models for interpreter and translator training* (Vol. 8). Amsterdam; Philadelphia: John Benjamins Publishing Company.

Grosjean, F. (1985). The bilingual as a competent but specific speaker-hearer. *Journal of Multilingual and Multicultural Development*, *6*(6), 467–477. doi: 10.1080/01434632.1985.9994221

Grosjean, F. (1989). Neurolinguists, beware! The bilingual is not two monolinguals in one person. *Brain and Language*, *36*(1), 3–15. doi: 10.1016/0093-934X(89)90048-5

Grosjean, F. (1997). The bilingual individual. *Interpreting*, *2*(1-2), 163–187. doi: 10.1075/intp.2.1-2.07gro

Grosjean, F. (2001). The bilingual's language modes. In J. L. Nicol (Ed.), *One Mind, Two Languages: Bilingual Language Processing* (pp. 1–22). Oxford, UK: Blackwell Publishers.

Gunning, R. (1952). *The technique of clear writing.* New York: McGraw-Hill.

Hajič, J., Cmejrek, M., Dorr, B., Ding, Y., Eisner, J., Gildea, D., . . . Rambow, O. (2004). *Natural Language Generation in the Context of Machine Translation* (Tech. Rep.). Baltimore: Center for Language and Speech Processing, Johns Hopkins University.

Hajič, J., & Zeman, D. (Eds.). (2017). *Proceedings of the CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies.* Vancouver, Canada: Association for Computational Linguistics. doi: 10.18653/v1/K17-3

Hale, S. B., & Campbell, S. (2002). The Interaction Between Text Difficulty and Translation Accuracy. *Babel*, *48*(1), 14–33. doi: 10.1075/babel.48.1.02hal

Halverson, S. L. (2015, January). Cognitive Translation Studies and the merging of empirical paradigms: The case of 'literal translation'. *Translation Spaces*, *4*(2), 310–340. doi: 10.1075/ts.4.2.07hal

Hancke, J., Vajjala, S., & Meurers, D. (2012, dec). Readability Classification for German using Lexical, Syntactic, and Morphological Features. In *Proceedings of COLING 2012* (pp. 1063–1080). Mumbai, India: The COLING 2012 Organizing Committee.

Hansen-Schirra, S., Nitzke, J., & Oster, K. (2017). Predicting cognate translation. *Empirical Modelling of Translation and Interpreting*, *7*, 3.

Hart, S. G., & Staveland, L. E. (1988, January). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Advances in Psychology* (Vol. 52, pp. 139–183). North-Holland. doi: 10.1016/S0166-4115(08)62386-9

Hartsuiker, R. J., Beerts, S., Loncke, M., Desmet, T., & Bernolet, S. (2016, October). Cross-linguistic structural priming in multilinguals: Further evidence for shared syntax. *Journal of Memory and Language*, *90*, 14–30. doi: 10.1016/j.jml.2016.03.003

Hartsuiker, R. J., & Bernolet, S. (2017, March). The development of shared syntax in second language learning*. *Bilingualism: Language and Cognition*, *20*(2), 219–234. doi: 10.1017/S1366728915000164

Hartsuiker, R. J., Pickering, M. J., & Veltkamp, E. (2004, June). Is syntax separate or shared between languages?: Cross-linguistic syntactic priming in Spanish-English bilinguals. *Psychological Science*, *15*(6), 409–414. doi: 10.1111/j.0956-7976.2004.00693.x

Heilmann, A. (2020). *Profiling effects of syntactic compexity in translation* (PhD thesis). Rheinisch-Westfaelische Technische Hochschule, Aachen.

Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.*

Huang, P.-S., Wang, C., Huang, S., Zhou, D., & Deng, L. (2018). Towards neural phrase-based machine translation. In *Proceedings of International Conference on Learning Representations.* Vancouver, Canada.

Hvelplund, K. T. (2011). *Allocation of Cognitive Resources in Translation: An Eye-tracking and Key-logging Study* (PhD thesis). Copenhagen Business School, Frederiksberg.

Immonen, S. (2011, dec). Unravelling the processing units of translation. *Across Languages and Cultures*, *12*(2), 235–257. doi: 10.1556/Acr.12.2011.2.6

Immonen, S., & Mäkisalo, J. (2010). Pauses reflecting the processing of syntactic units in monolingual text production and translation. *HERMES - Journal of Language and Communication in Business*, *23*(44), 45–61.

Ivir, V. (1981). Formal correspondence vs. translation equivalence revisited. *Poetics Today*, *2*(4), 51. doi: 10.2307/1772485

Jacob, G., Katsika, K., Family, N., & Allen, S. E. M. (2017, March). The role of constituent order and level of embedding in cross-linguistic structural priming. *Bilingualism: Language and Cognition*, *20*(2), 269–282. doi: 10.1017/S1366728916000717

Jakobsen, A. L. (2011). Tracking Translators' Keystrokes and Eye Movements with Translog. In C. Alvstad, A. Hild, & E. Tiselius (Eds.), *Methods and Strategies of Process Research: Integrative Approaches in Translation Studies* (Vol. 94, pp. 37–55). Amsterdam, The Netherlands: John Benjamins Publishing Company.

Jakobsen, A. L., & Jensen, K. T. H. (2008). Eye Movement Behaviour across Four Different Types of Reading Task. In S. Göpferich, A. L. Jakobsen, & I. Mees (Eds.), *Copenhagen Studies in Language* (Vol. 36, pp. 103–124). Copenhagen, Denmark: Samfundslitteratur.

Jakobson, R. (1971). On linguistic aspects of translation. In *Selected Writings: Word and Language* (Vol. 2, pp. 260–266). De Gruyter Mouton. doi: 10.1515/9783110873269.260

Jaynes, E. T. (1963). Information theory and statistical mechanics. In *Statistical Physics* (pp. 181–218). New York, USA: Benjamin.

Jaynes, E. T. (1968). Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, *sec-4*(3), 227–241.

Jensen, K. T. H. (2009). Indicators of Text Complexity. In S. Göpferich, A. L. Jakobsen, & I. Mees (Eds.), *Copenhagen Studies in Language* (Vol. 37, pp. 61–80). Copenhagen, Denmark: Samfundslitteratur.

Joshi, A. K., Levy, L. S., & Takahashi, M. (1975, February). Tree adjunct grammars. *Journal of Computer and System Sciences*, *10*(1), 136–163. doi: 10.1016/S0022-0000(75)80019-5

Jurafsky, D., & Martin, J. (2008). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (Second ed.). Prentice Hall.

Kade, O. (1968). *Zufall und Gesetzmässigkeit in der Übersetzung.* Leipzig, Germany:

Verlag Enzyklopädie.

Kay, M., & Roscheisen, M. (1993). Text-translation alignment. *Computational Linguistics*, *19*(1), 121–142.

Kim, H.-Y. (2013, February). Statistical notes for clinical researchers: assessing normal distribution (2) using skewness and kurtosis. *Restorative Dentistry & Endodontics*, *38*(1), 52–54. doi: 10.5395/rde.2013.38.1.52

Kincaid, J. P., Fishburne, R. P., Rogers, R. L., & Chissom., B. S. (1975). *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel* (Research branch report No. RBR-8-75). Springfield, Virginia: National Technical Information Service / Naval Technical Training Command Millington Tenn Research Branch.

Kiraly, D. C. (1995). *Pathways to translation: Pedagogy and process* (No. 3). Kent, Ohio: Kent State University Press.

Klare, G. R. (1976). A second look at the validity of readability formulas. *Journal of Reading Behavior*, *8*(2), 129–152. doi: 10.1080/10862967609547171

Klare, G. R. (1984). Readability. In P. D. Pearson & R. Barr (Eds.), *Handbook of Reading Research* (Vol. 1, pp. 681–744). New York, USA: Longman.

Koehn, P. (2004). Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In R. E. Frederking & K. B. Taylor (Eds.), *Machine Translation: From Real Users to Research* (pp. 115–124). Berlin, Heidelberg: Springer. doi: 10.1007/978-3-540-30194-3_13

Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., & Talbot, D. (2005). Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT}.*

Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of NAACL-HLT 2003* (pp. 48–54). Edmonton, Canada.

Koehn, P., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E., ... Moran, C. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 demo and poster sessions* (pp. 177–180). Prague, Czech Republic. doi: 10.3115/1557769.1557821

Komsta, L., & Novomestky, F. (2015). *Moments: Moments, cumulants, skweness, kurtosis and related tests.* Retrieved from `https://cran.r-project.org/package=moments`

Koster, J. (1975). Dutch as an SOV language. *Linguistic Analysis*, 111–136.

Krings, H. (2001). *Repairing texts: Empirical investigations of machine translation post-editing processes* (G. Koby, Ed.). Kent, Ohio, US: The Kent State University Press.

Kroll, J. F., & Stewart, E. (1994, April). Category interference in translation and picture naming: Evidence for asymmetric connections between bilingual memory representations. *Journal of Memory and Language*, *33*(2), 149–174. doi: 10.1006/jmla.1994.1008

Kromann, M. T. (2003). The Danish dependency treebank and the DTAG treebank tool. In *Proceedings of the 2nd international workshop on treebanks and linguistic theories.*

Kumpulainen, M. (2015). On the operationalisation of ’pauses’ in translation process research. *Translation & Interpreting*, *7*(1), 47–58. doi: ti.106201.2015.a04

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, *82*(13), 1–26.

doi: 10.18637/jss.v082.i13

Lacruz, I. (2017). Cognitive effort in translation, editing, and post-editing. In J. Schwieter & A. Ferreira (Eds.), *The Handbook of Translation and Cognition* (pp. 386–401). Malden, Massachusetts, US: John Wiley & Sons, Inc.

Lacruz, I., Shreve, G. M., & Angelone, E. (2012). Average pause ratio as an indicator of cognitive effort in post-editing: A case study. In *Proceedings of AMTA 2012 Workshop on Post-Editing Technology and Practice* (pp. 21–30). San Diego, California, USA.

Lederer, M. (1994). *La traduction aujourd'hui: Le modèle interprétatif.* Vanves: Hachette Français Langue Etrangère.

Lederer, M. (2003). *Translation: The interpretive model* (Second ed.). New York, USA: Routledge.

Leijten, M., & Van Waes, L. (2013). Keystroke logging in writing research: Using inputlog to analyze and visualize writing processes. *Written Communication*, *30*(3), 358–392. doi: 10.1177/0741088313491692

Levelt, W. J. M. (1989). *Speaking: From intention to articulation.* Cambridge, MA, US: The MIT Press.

Levelt, W. J. M., & Kelter, S. (1982, January). Surface form and memory in question answering. *Cognitive Psychology*, *14*(1), 78–106. doi: 10.1016/0010-0285(82)90005-6

Liu, Y., Zheng, B., & Zhou, H. (2019). Measuring the difficulty of text translation: The combination of text-focused and translator-oriented approaches. *Target*, *31*(1), 125–149. doi: 10.1075/target.18036.zhe

Loebell, H., & Bock, K. (2003, January). Structural priming across languages. *Linguistics*, *41*(5). doi: 10.1515/ling.2003.026

Macizo, P., & Bajo, M. T. (2004). When translation makes the difference: Sentence processing in reading and translation. *Psicológica*(25), 181–205.

Macizo, P., & Bajo, M. T. (2006, February). Reading for repetition and reading for translation: Do they involve the same processes? *Cognition*, *99*(1), 1–34. doi: 10.1016/j.cognition.2004.09.012

Macken, L. (2010). *Sub-sentential alignment of translational correspondences* (PhD thesis). Ghent University, Ghent, Belgium.

Macken, L., De Clercq, O., & Paulussen, H. (2011). Dutch parallel corpus: A balanced copyright-cleared parallel corpus. *Meta: Journal des Traducteurs/Meta: Translators' Journal*, *56*(2), 374–390. doi: 10.7202/1006182ar

Malmkjær, K. (2006). Translation units. In K. Brown (Ed.), *Encyclopedia of language & linguistics (second edition)* (Second Edition ed., pp. 92–93). Oxford: Elsevier. doi: 10.1016/B0-08-044854-2/00491-0

Matthews, P. (1981). *Syntax.* Cambridge University Press.

Matthiessen, C. M. (2001). The environment of translation. In E. Steiner & C. Yallop (Eds.), *Exploring Translation and Multilingual Text Production: Beyond Content* (pp. 41–124). Berlin; New York: Mouton de Gruyer.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., … Lee, J. (2013, August). Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 92–97). Sofia, Bulgaria: Association for Computational Linguistics (ACL).

Mihalcea, R., & Pedersen, T. (2003). An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and* *using parallel texts data driven machine translation and beyond* (Vol. 3, pp. 1–10). Edmonton, Canada: Association for Computational Linguistics. doi: 10.3115/1118905.1118906

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv e-prints.*

Mishra, A., Bhattacharyya, P., & Carl, M. (2013). Automatically predicting sentence translation difficulty. In *Proceedings of the 51st Annual Meeting on Association for Computational Linguistics (ACL 2013)* (pp. 346–351). Sofia, Bulgaria.

Muñoz Martín, R. (2012, January). Just a matter of scope. *Translation Spaces*, *1*(1), 169–188. doi: 10.1075/ts.1.08mun

Naskar, S. K., Toral, A., Gaspari, F., & Way, A. (2011). A framework for diagnostic evaluation of MT based on linguistic checkpoints. In *Proceedings of the 13th Machine Translation Summit* (pp. 529–536). Xiamen, China.

Nida, E. (1964). *Toward a science of translating.* Leiden, Netherlands: E.J. Brill.

Niehues, J., & Cho, E. (2017). Exploiting linguistic resources for neural machine translation using multi-task learning. In *Proceedings of the second conference on machine translation* (pp. 80–89).

Nikolaev, D., Arviv, O., Karidi, T., Kenneth, N., Mitnik, V., Saeboe, L. M., & Abend, O. (2020, July). Fine-grained analysis of cross-linguistic syntactic divergences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 1159–1176). Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.109

Nivre, J. (2015). Towards a universal grammar for natural language processing. In *International conference on intelligent text processing and computational linguistics* (pp. 3–16).

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., ... others (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the tenth international conference on language resources and evaluation (LREC'16)* (pp. 1659–1666).

Nivre, J., & Megyesi, B. (2007). Bootstrapping a Swedish Treebank using cross-corpus harmonization and annotation projection. In *Proceedings of the 6th international workshop on treebanks and linguistic theories* (pp. 97–102).

Nord, C. (2005). *Text Analysis in Translation* (Second ed.). Amsterdam: Rodopi.

O'Brien, S. (2004). Machine translatability and post-editing effort: How do they relate? In *Proceedings of the Twenty-Sixth International Conference on Translating and the Computer.* London, UK.

O'Brien, S. (2005). Methodologies for measuring the correlations between post-editing effort and machine translatability. *Machine Translation*, *19*(1), 37–58. doi: 10.1007/s10590-005-2467-1

O'Brien, S. (2006). Pauses as indicators of cognitive effort in post-editing machine translation output. *Across Languages and Cultures*, *7*(1), 1–21. doi: 10.1556/Acr.7.2006.1.1

O'Brien, S. (2007). Eye-tracking and translation memory matches. *Perspectives: Studies in Translatology*, *14*(3), 185–205. doi: 10.1080/09076760708669037

Och, F. J., Gildea, D., Khudanpur, S., & Sarkar, A. (2004). A smorgasbord of features for statistical machine translation. In *Proceedings of the human language technology conference of the North American chapter of the association for computational linguistics.*

Och, F. J., & Ney, H. (2000). A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on computational linguistics* (Vol. 2, pp. 1086–1090). Saarbrücken, Germany: Association for Computational Linguistics. doi: 10.3115/992730.992810

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, *29*(1), 19–51. doi: 10.1162/089120103321337421

Och, F. J., Tillmann, C., & Ney, H. (1999). Improved alignment models for statistical machine translation. In *Proceedings of the Joint SIGDAT Conference* (pp. 20–28). Maryland, USA.

Osborne, T., & Gerdes, K. (2019). The status of function words in dependency grammar: A critique of Universal Dependencies (UD). *Glossa: a journal of general linguistics*, *4*(1), 17. doi: 10.5334/gjgl.537

Panou, D. (2013, January). Equivalence in translation theories: A critical evaluation. *Theory and Practice in Language Studies*, *3*(1), 1–6. doi: 10.4304/tpls.3.1.1-6

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc.

Pawlik, M., & Augsten, N. (2015). Efficient computation of the tree edit distance. *ACM Transactions on Database Systems*, *40*(1). doi: 10.1145/2699485

Pawlik, M., & Augsten, N. (2016). Tree edit distance: Robust and memory-efficient. *Information Systems*, *56*, 157–173. doi: 10.1016/j.is.2015.08.004

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Cournapeau, D. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*, 2825–2830.

Peng, X., Li, Z., Zhang, M., Wang, R., Zhang, Y., & Si, L. (2019). Overview of the NLPCC 2019 shared task: Cross-domain dependency parsing. In *CCF international conference on natural language processing and chinese computing* (pp. 760–771).

Pennington, J., Socher, R., & Manning, C. (2014, oct). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics. doi: 10.3115/v1/D14-1162

Peter, J.-T., Nix, A., & Ney, H. (2017, jun). Generating alignments using target foresight in attention-based neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, *108*(1), 27–36. doi: 10.1515/pralin-2017-0006

Petrov, S., Das, D., & McDonald, R. (2012, May). A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)* (pp. 2089–2096). Istanbul, Turkey: European Language Resources Association (ELRA).

Prior, A., MacWhinney, B., & Kroll, J. F. (2007, November). Translation norms for English and Spanish: The role of lexical variables, word class, and L2 proficiency in negotiating translation ambiguity. *Behavior Research Methods*, *39*(4), 1029–1038. doi: 10.3758/BF03193001

Pym, A. (2014). *Exploring translation theories* (Second Edition ed.). London ; New York: Routledge.

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, C. D. (2020, jul). Stanza: A Python natural language processing toolkit for many human languages. In

Proceedings of the 58th annual meeting of the association for computational linguistics: System demonstrations (pp. 101–108). Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.14

R Core Team. (2019). *R: A language and environment for statistical computing* [manual]. Vienna, Austria.

R Core Team. (2020). *R: A Language and Environment for Statistical Computing* [manual]. Vienna, Austria: R Foundation for Statistical Computing.

Reichle, E. D., Warren, T., & McConnell, K. (2009, February). Using E-Z Reader to model the effects of higher level language processing on eye movements during reading. *Psychonomic Bulletin & Review*, *16*(1), 1–21. doi: 10.3758/PBR.16.1.1

Ruiz, C., Paredes, N., Macizo, P., & Bajo, M. (2008, July). Activation of lexical and syntactic target language properties in translation. *Acta Psychologica*, *128*(3), 490–500. doi: 10.1016/j.actpsy.2007.08.004

Ruíz, J. O., & Macizo, P. (2019, August). Lexical and syntactic target language interactions in translation. *Acta Psychologica*, *199*, 102924. doi: 10.1016/j.actpsy.2019.102924

Schaeffer, M., & Carl, M. (2013, December). Shared representations and the translation process: A recursive model. *Translation and Interpreting Studies*, *8*(2), 169–190. doi: 10.1075/tis.8.2.03sch

Schaeffer, M., & Carl, M. (2014). Measuring the cognitive effort of literal translation processes. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation* (pp. 29–37). Gothenburg, Sweden: Association for Computational Linguistics. doi: 10.3115/v1/W14-0306

Schaeffer, M., & Carl, M. (2017). Language processing and translation. In S. Hansen-Schirra, O. Czulo, & S. Hofmann (Eds.), *Empirical Modelling of Translation and Interpreting* (pp. 117–154). Berlin, Germany: Language Science Press.

Schaeffer, M., Carl, M., Lacruz, I., & Aizawa, A. (2016). Measuring cognitive translation effort with activity units. *Baltic Journal of Modern Computing*, *4*(2), 331–345.

Schaeffer, M., Dragsted, B., Hvelplund, K. T., Balling, L. W., & Carl, M. (2016). Word translation entropy: Evidence of early target language activation during reading for translation. In M. Carl, S. Bangalore, & M. Schaeffer (Eds.), *New directions in empirical translation process research* (pp. 183–210). Cham, Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-20358-4_9

Schwarm, S. E., & Ostendorf, M. (2005). Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)* (pp. 523–530). Ann Arbor, Michigan, USA. doi: 10.3115/1219840.1219905

Schwarz, G. (1978, March). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464. doi: 10.1214/aos/1176344136

Seleskovitch, D. (1976). Interpretation: A psychological approach to translating. In R. W. Brislin (Ed.), *Translation: Applications and Research* (pp. 92–116). New York, USA: Gardner Press Inc.

Sennrich, R., & Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the first conference on machine translation: Volume 1, research papers* (pp. 83–91).

Serbina, T., Hintzen, S., Niemietz, P., & Neumann, S. (2017). Changes of word class during translation: Insights from a combined analysis of corpus, keystroke logging and eye-tracking data. In S. Hansen-Schirra, O. Czulo, & S. Hofmann

(Eds.), *Empirical modelling of translation and interpreting* (Vol. 7, pp. 177–208). Berlin, Germany: Language Science Press.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423. doi: 10.1002/j.1538-7305.1948.tb01338.x

Sharmin, S., Špakov, O., Räihä, K.-J., & Jakobsen, A. L. (2008, March). Effects of time pressure and text complexity on translators' fixations. In *Proceedings of the 2008 symposium on Eye tracking research & applications* (pp. 123–126). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1344471.1344503

Shuttleworth, M., & Cowie, M. (2014). *Dictionary of Translation Studies* (eBook publication of 1997 ed.). Routledge. doi: 10.4324/9781315760490

Si, L., & Callan, J. (2001). A statistical model for scientific readability. In *Proceedings of the Tenth International Conference on Information and Knowledge Management* (pp. 574–576). Atlanta, Georgia, USA. doi: 10.1145/502585.502695

Skut, W., Krenn, B., Brants, T., & Uszkoreit, H. (1997). An annotation scheme for free word order languages. In *Fifth conference on applied natural language processing* (pp. 88–95). Association for Computational Linguistics. doi: 10.3115/974557.974571

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas.*

Socher, R., Lin, C. C., Ng, A. Y., & Manning, C. D. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 26th international conference on machine learning (ICML).*

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 151–161). Stroudsburg, PA, USA: Association for Computational Linguistics.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of machine learning research*, *15*(1), 1929–1958.

Steiner, E. (2004). Ideational grammatical metaphor: Exploring some implications for the overall model. *Languages in Contrast*, *4*(1), 137–164. doi: 10.1075/lic.4.1.07ste

Sun, S. (2012). *Measuring difficulty in English-Chinese translation: Towards a general model of translation difficulty* (PhD thesis). Kent State University, Kent, Ohio, USA.

Sun, S. (2015). Measuring translation difficulty: Theoretical and methodological considerations. *Across languages and cultures*, *16*(1), 29–54. doi: 10.1556/084.2015.16.1.2

Sun, S., & Shreve, G. M. (2014). Measuring Translation Difficulty: An Empirical Study. *Target*, *26*(1), 98–127. doi: 10.1075/target.26.1.04sun

Tezcan, A., Hoste, V., & Macken, L. (2017, jun). A neural network architecture for detecting grammatical errors in statistical machine translation. *The Prague bulletin of mathematical linguistics*, *108*(1), 133–145. doi: 10.1515/pralin-2017-0015

Tezcan, A., Hoste, V., & Macken, L. (2019, may). Estimating post-editing time using a gold-standard set of machine translation errors. *Computer Speech & Language*, *55*, 120–144. doi: 10.1016/j.csl.2018.10.005

Tirkkonen-Condit, S. (2005). The monitor model revisited: Evidence from process research. *Meta: Journal des Traducteurs/Meta: Translators' Journal*, *50*(2), 405–414.

Tokowicz, N., & Kroll, J. F. (2007, August). Number of meanings and concreteness: Consequences of ambiguity within and across languages. *Language and Cognitive Processes*, *22*(5), 727–779. doi: 10.1080/01690960601057068

Toral, A., & Sánchez-Cartagena, V. M. (2017). A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics* (Vol. 1, pp. 1063–1073). Valencia, Spain: Association for computational linguistics. doi: 10.18653/v1/E17-1100

Toury, G. (1995). *Descriptive Translation Studies and Beyond.* Amsterdam ; Philadelphia: John Benjamins Publishing Company.

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 384–394). Uppsala, Sweden: Association for Computational Linguistics.

Underwood, N., & Jongejan, B. (2001). Translatability checker: A tool to help decide whether to use MT. In *Proceedings of the 8th Machine Translation Summit* (pp. 363–368). Santiago de Compostela, Spain.

Van Brussel, L., Tezcan, A., & Macken, L. (2018). A fine-grained error analysis of NMT, PBMT and RBMT output for English-to-Dutch. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation* (pp. 3799–3804). Miyazaki, Japan: European Language Resources Association (ELRA).

Vanroy, B., De Clercq, O., & Macken, L. (2019). Correlating process and product data to get an insight into translation difficulty. *Perspectives*, *27*(6), 924–941. doi: 10.1080/0907676X.2019.1594319

Vanroy, B., De Clercq, O., Tezcan, A., Daems, J., & Macken, L. (in press). Metrics of syntactic equivalence to assess translation difficulty. In M. Carl (Ed.), *Explorations in empirical translation process research.* Springer International Publishing.

Vanroy, B., Schaeffer, M., & Macken, L. (2021). Comparing the effect of product-based metrics on the translation process.

Vanroy, B., Tezcan, A., & Macken, L. (2019). Predicting syntactic equivalence between source and target sentences. *Computational Linguistics in the Netherlands Journal*, 101–116.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Proceedings of NeurIPS 2017* (pp. 1–15). Long Beach, CA, USA.

Vinay, J.-P., & Darbelnet, J. (1995). *Comparative stylistics of French and English: A methodology for translation* (Vol. 11; J. C. Sager & M.-J. Hamel, Trans.). Amsterdam ; Philadelphia: John Benjamins Publishing Company.

Xia, F., & McCord, M. (2004). Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the conference on computational linguistics* (pp. 508–514). Bombay, India.

Yamada, K., & Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of the 39th annual meeting on association for computational linguistics* (pp. 523–530). Toulouse, France: Association for Computational Linguistics.

doi: 10.3115/1073012.1073079

Yamada, K., & Knight, K. (2002, July). A decoder for syntax-based statistical MT. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 303–310). USA: Association for Computational Linguistics. doi: 10.3115/1073083.1073134

Zeman, D., Hajič, J., Popel, M., Potthast, M., Straka, M., Ginter, F., ... Petrov, S. (2018). CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies* (pp. 1–21). Brussels, Belgium: Association for Computational Linguistics. doi: 10.18653/v1/K18-2001

Zhang, J., Wang, M., Liu, Q., & Zhou, J. (2017). Incorporating word reordering knowledge into attention-based neural machine translation. In *Proceedings of the 55th annual meeting of the association for computational linguistics* (Vol. 1, pp. 1524–1534). Vancouver, Canada: Association for Computational Linguistics. doi: 10.18653/v1/P17-1140

Zwart, C. J.-W. (1994). Dutch is head-initial. *The Linguistic Review*, *11*(3-4). doi: 10.1515/tlir.1994.11.3-4.377