Noname manuscript No.
(will be inserted by the editor)

Metrics of syntactic equivalence to assess translation difficulty

Bram Vanroy · Orphée De Clercq · Arda Tezcan · Joke Daems · Lieve Macken

Received: date / Accepted: date

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

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 Schwarm

Bram Vanroy
Groot-Brittanniëlaan 45, 9000 Gent, Belgium
Tel.: +32 9 33 11 939
E-mail: Bram.Vanroy@UGent.be

and Ostendorf (2005); Francois and Miltsakaki (2012); De Clercq et al. (2014); De Clercq and Hoste (2016), 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 and 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:

(1) 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 2 we will first discuss background literature concerning the importance of syntactic equivalence with respect to translatability and previous research of equivalence. In Section 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 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. 5). Finally, we end with a conclusion and thoughts for future work concerning quantifying syntactic equivalence (Sec. 6).

2 Related research

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 and 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 and Carl, 2014), 'literal translation' is defined by three criteria:

(2) 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 2a, they use word translation entropy, 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 2b and 2c for literal translation relate closely to what we consider syntactic equivalence as described in 1. 1a (differences in word (group) order) relates to criterion 2b (identical word order) above, and 2c is most similar to 1c: 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. 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 et al. (2019b) (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 et al. (2019b), 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 (Skut et al., 1997; Jurafsky and Martin, 2008). 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 1.

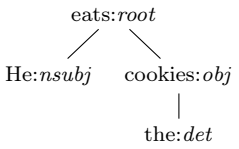

Fig. 1 Example of a dependency tree of the sentence 'He eats the cookies'

In recent years, research on automatic parsing methods has increased due to the availability of linguistically annotated corpora (treebanks) for many different languages (Hajič and Zeman, 2017; Zeman et al., 2018; Peng et al., 2019). However, despite their availability, the annotation schemes in treebanks vary significantly across languages, such as between the Swedish Treebank (Nivre and Megyesi, 2007), the Danish Dependency Treebank (Kromann, 2003), and Stanford Typed Dependencies (de Marneffe and 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[1] (UD) is an initiative to mitigate this problem

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

by developing a framework for cross-linguistically consistent morphosyntactic annotation (Nivre et al., 2016), which we will discuss further in Section 3.1.

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 and 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 4c, 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 and Ney, 2000, Figure 1), but in this paper we opt for line diagrams such as Figure 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 aligning those corpora is not feasible because of their size. Instead, we rely on automatic alignment systems. In previous research (Vanroy et al., 2019b), we justified using GIZA++ (Och and Ney, 2003) in favor of another tool, `fast_align` (Dyer et al., 2013), because of its lower Alignment Error Rate (Och and Ney, 2000; Mihalcea and Pedersen, 2003).

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. 3) 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 3b.

(3) 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 3.2),

we want to capture the alignment of word groups. In previous research (Vanroy et al., 2019b), 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.

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.[2] Vanroy et al. (2019b) 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 4 (taken from Vanroy et al., 2019b, 104) is visualised in Figure 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 the `word_cross` value. The source and target segments can be aligned as shown in Example 4c. 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.

(4) 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

[2] 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.

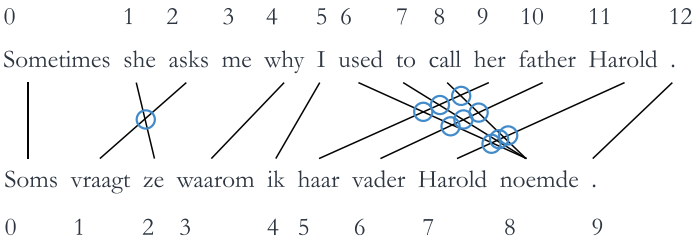

Fig. 2 Visualisation of `cross` in Ex. 4 with a `word_seq` value of $^{10}/_{12} = 0.83$. (modified from Vanroy et al., 2019b)

This approach is word-based, but as discussed in Section 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 et al. (2019b) 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 et al., 2019b, 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 2 can be modified so that instead of word movement, group movement is quantified (Figure 3).

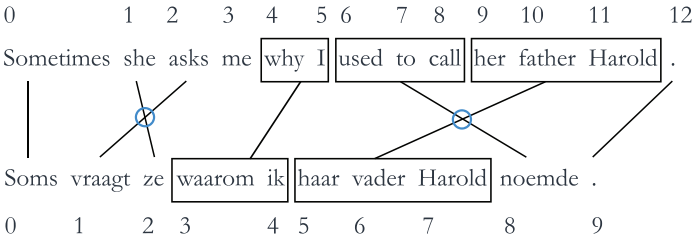

Fig. 3 Example of `seq_cross` in Ex. 4 with a total value of $^{2}/_{7} = 0.286$ (modified from Vanroy et al., 2019b)

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 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.

3 Metrics

As discussed in Section 1, we restrict ourselves to three sub-components of syntactic equivalence,[3] 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. 3.2), then we discuss how label changes play a role (Sec. 3.3), and finally we introduce a method to calculate aligned syntactic tree edit distance (ASTrED, Sec. 3.4). A concise overview table of the metrics is given in Section 3.5. As all three metrics are based on comparing the syntactic structures of the source and target sentences using dependency representations, we start by explaining the chosen paradigm, Universal Dependencies, in closer detail.

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. 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 4 are visualised in Figure 4[4] and 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

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

[4] 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 and Megyesi, 2007; Nivre, 2015; Nivre et al., 2016). 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).

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.

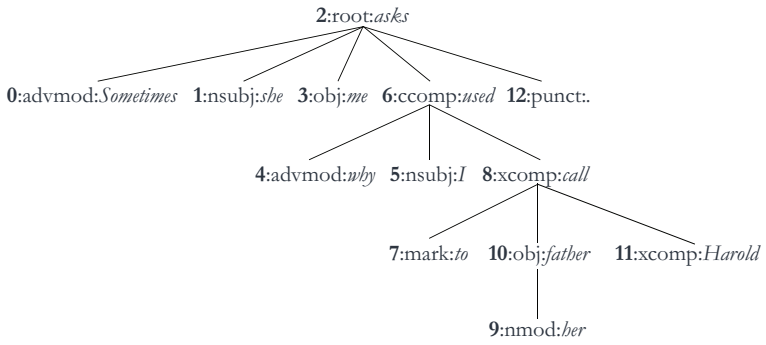

Fig. 4 Source dependency tree of Ex. 4: 'Sometimes she asks me why I used to call her father Harold .'.

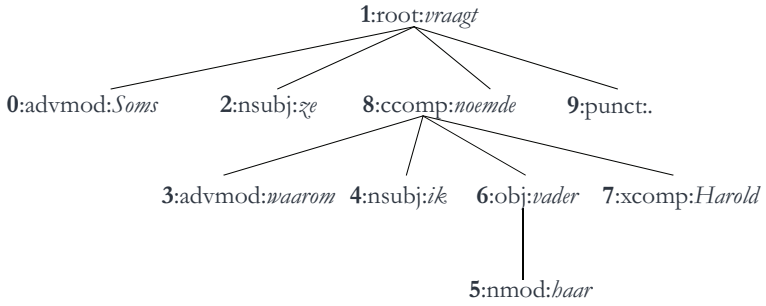

Fig. 5 Target dependency tree of Ex. 4: 'Soms vraagt ze waarom ik haar vader Harold noemde .'.

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`.

3.2 Syntactically aware cross

In Section 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 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'.[5]

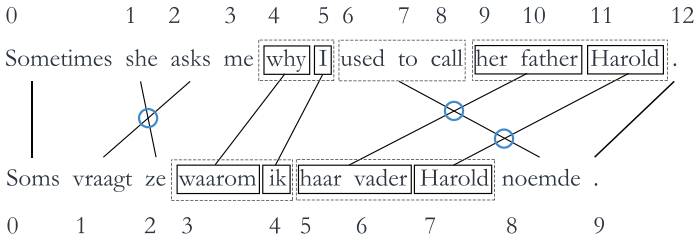

Fig. 6 Example of SACr with a total value of $3/9 = 0.33$. Dotted boxes indicate the initial groups of `seq_cross`. When required, these groups are split up into linguistically motivated SACr groups (solid boxes)

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

[5] 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'.

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 7 and 8 illustrate which of the proposed sequence groups (cf. dotted boxes in Figure 3) 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.

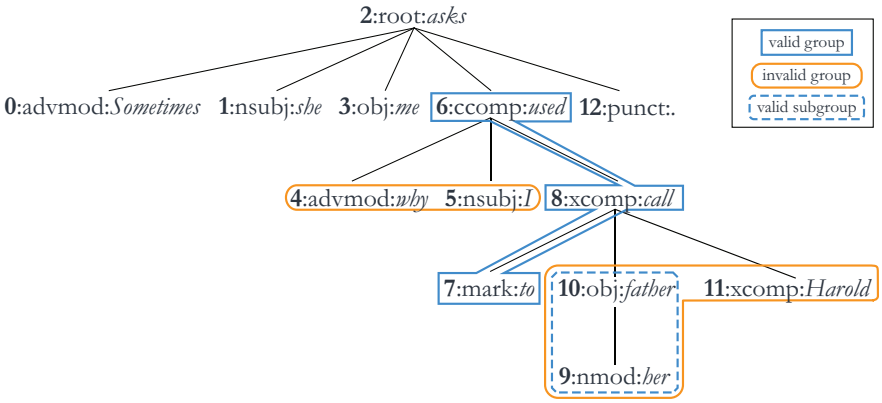

Fig. 7 Source dependency tree of Ex. 4 with highlighted groups: 'Sometimes she asks me why I used to call her father Harold .'.

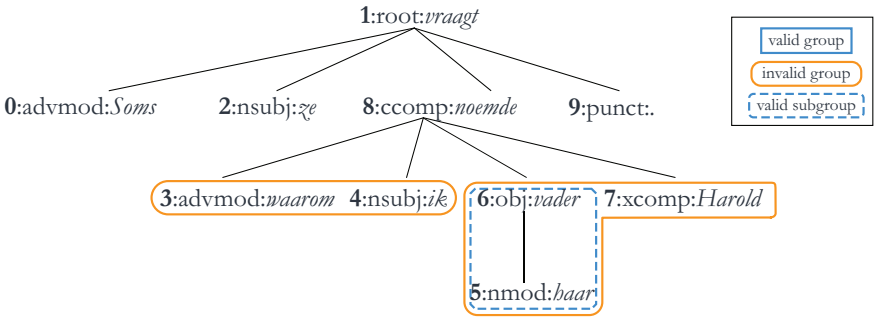

Fig. 8 Target dependency tree of Ex. 4 with highlighted groups: 'Soms vraagt ze waarom ik haar vader Harold noemde .'.

Figure 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 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. 2.3).

3.2.1 Cross summary

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.

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. 1)[6].

$$L = \#\{(src, tgt) \in A : src \neq tgt\} \tag{1}$$

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. 5a, which has been translated into a passive construction (Ex. 5b), and their word alignment (Ex. 5c).

(5) 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 9.

[6] 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.

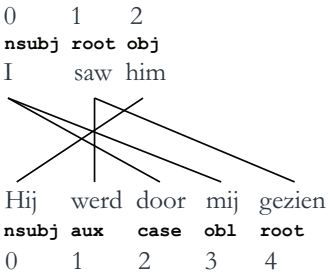

Fig. 9 Word alignment visualisation of Ex. 5

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. 5 in Table 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: $4/5 = \mathbf{0.8}$)	

Table 1 Label changes for Ex. 5.

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. 6 and its word alignment (visualised in Figure 10). This example will be used to explain ASTrED in the following subsections.

(6) 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

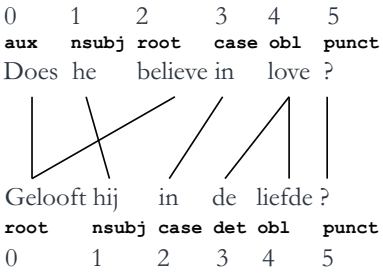

Fig. 10 Word alignment visualisation of Ex. 6

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.

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. 3.1). As an example, let us take the previously mentioned example Ex. 6. 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 11, 12).

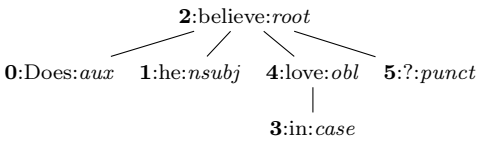

Fig. 11 Source dependency tree of Ex. 6a: 'Does he believe in love ?'.

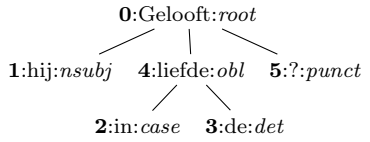

Fig. 12 Target dependency tree of Ex. 6b: 'Gelooft hij in de liefde ?'.

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.[7] 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 6, we can distinguish five groups (Example 7) where the corresponding words are given between brackets:

(7)
- `aux:root|root:root` (does:gelooft|believe:gelooft)
- `nsubj:nsubj` (he:hij)
- `case:case` (in:in)
- `obl:det,obl` (love:de,liefde)
- `punct:punct` (?:?)

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

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

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 13 and 14 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'.

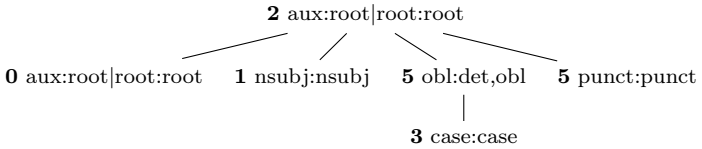

Fig. 13 Modified source dependency tree of Example 6a: 'Does he believe in love ?'

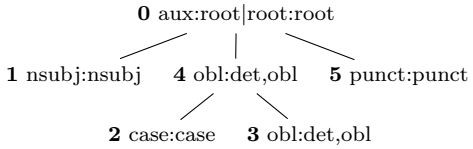

Fig. 14 Modified target dependency tree of Example 6b: 'Gelooft hij in de liefde ?'

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 13 to the modified target tree in Figure 14, two operations are needed, as visualised in Figure 15:

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. 4.2 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 $^2/_6 = 0.33$.

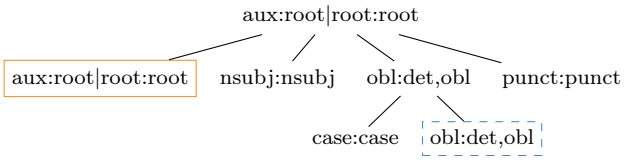

Fig. 15 A visualisation of the two needed edits to go from modified source tree in Figure 13 to the modified target tree in Figure 14. The orange solid box indicates the source node that needs to be deleted and the dashed blue box highlights the target node that needs to inserted.

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.

3.5 Metrics overview

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

Table 2 An overview of the metrics introduced in this paper.

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.

4.1 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. 8 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'.

(8) 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 16. The `word_cross` value is $2/7 = 0.29$.

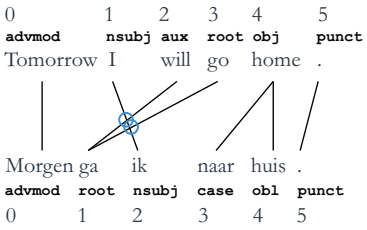

Fig. 16 Visualisation of word alignment of Ex. 8. And a `word_cross` value of $2/7 = 0.29$.

Vanroy et al. (2019b) suggested a sequential approach to word reordering where consecutive words are grouped together following a given set of criteria

(cf. Sec.2.3). In the example above, this can be visualised as in Figure 17, showing a `seq_cross` value of $1/4 = 0.25$.

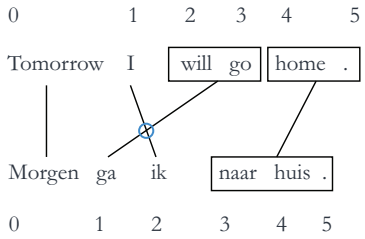

Fig. 17 `seq_cross` representation of Ex. 8 with a value of $1/4 = 0.25$.

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 on other word in the group. The valid and invalid groups are shown for both the source and target dependency trees in Figures 18 and 19.

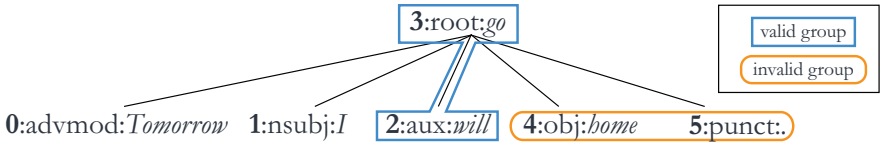

Fig. 18 Source dependency tree of Ex. 8, highlighting valid and invalid groups.

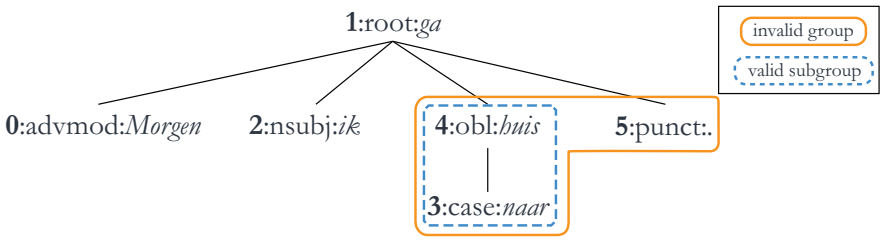

Fig. 19 Target dependency tree of Ex. 8, highlighting an invalid group and a valid SACr subgroup.

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 $1/5 = 0.2$.

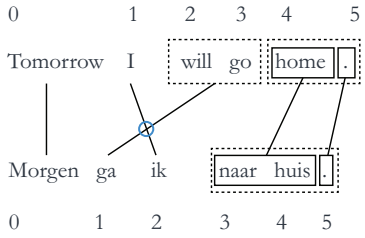

Fig. 20 SACr representation of Ex. 8 with a value of $1/5 = 0.2$. Dotted boxes indicates the groups of `seq_cross`, which, when required, are split up into linguistically motivated SACr groups (solid boxes)

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 16, we find three alignments where the labels of the source word have changed (Table 3), which when normalised gives a value of $3/6 = \mathbf{0.5}$.

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: $3/6 = \mathbf{0.5}$)	

Table 3 Label changes for Ex. 8.

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. 8, we can distinguish five groups (Ex. 9).

(9)
- `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 18 and 19) by changing the label of each node to the serialised group that it belongs to. The modified trees are given in:

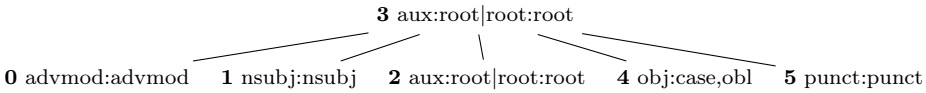

Fig. 21 Modified source dependency tree of Ex. 8: 'Tomorrow I will go home .'.

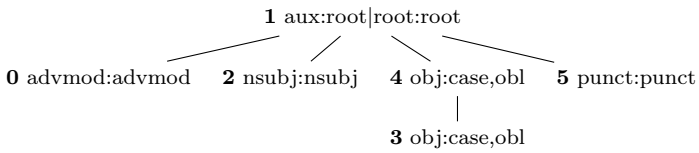

Fig. 22 Modified target dependency tree of Ex. 8: 'Morgen ga ik naar huis .'.

These modified trees can then finally be used to calculate tree edit distance. Figure 23 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 $^2/_6 = 0.33$.

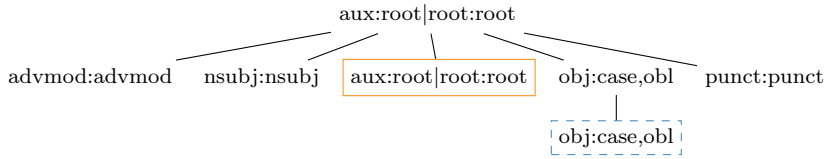

Fig. 23 A visualisation of the two needed edits to go from the modified source tree in Figure 21 to the modified target tree in Figure 22. The orange solid box indicates the source node that needs to be deleted and the dashed blue box highlights the target node that needs to inserted.

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 20). 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 23) 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 23). The results of all metrics for this example are summarised in Table 4.

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

Table 4 Summary of the results of all metrics for Ex. 8 (rounded to two decimals).

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[8] to the verb 'leek' *seemed*, namely 'mij' *to me*. Because of this explicitation, 'mij' cannot be aligned with a source word.

(10) 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 10c can be visualised in Figure 24, 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 $2/6 = 0.33$.

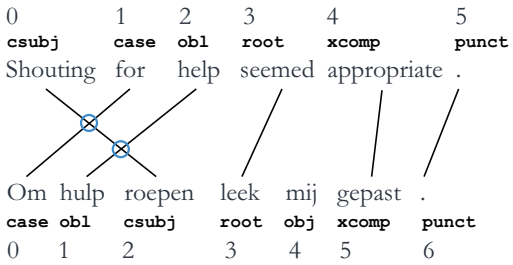

Fig. 24 Visualisation of word alignment in Ex. 10. And a `word_cross` value of $2/6 = 0.33$

When grouping consecutive words, as discussed in Section 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 25. Grouping 'for help' and 'Om hulp' leads to a reduction in crossing links: now there is only one crossing. The `seq_cross` value is $1/4 = 0.25$.

[8] 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`)

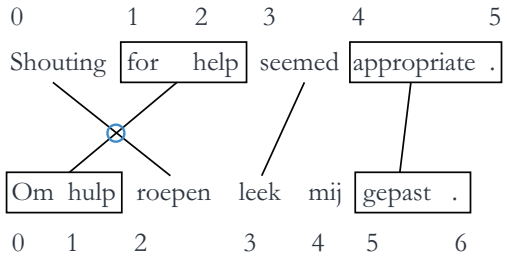

Fig. 25 `seq_cross` representation of Ex. 10 with a value of $1/4 = 0.25$

However, as discussed in Section 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 26 and 27 respectively.

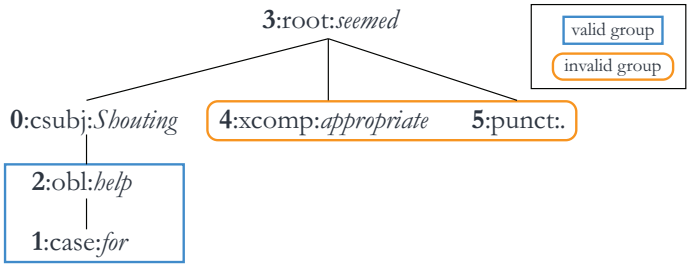

Fig. 26 Source dependency tree of Ex. 10, highlighting an invalid group and a valid SACr subgroup

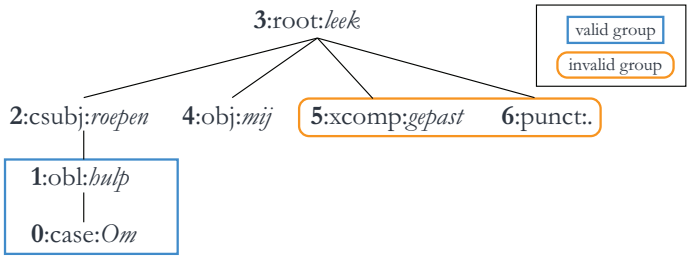

Fig. 27 Target dependency tree of Ex. 10, highlighting an invalid group and a valid SACr subgroup

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 28. 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.

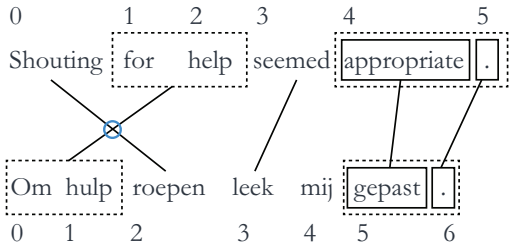

Fig. 28 SACr representation of Ex. 10 with a value of $1/5 = 0.2$. Dotted boxes indicates the groups of `seq_cross`, which, when required, are split up into linguistically motivated SACr groups (solid boxes)

The label changes in this example are quite self-explanatory: looking at the word alignments in Figure 24 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 11, we can find six groups and also one unaligned word ('mij' *me*).

(11)
- `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 29 and 30).

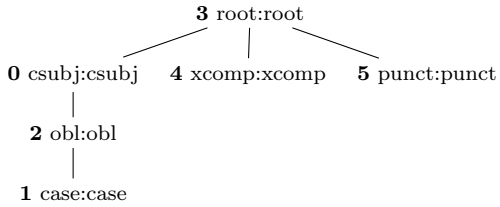

Fig. 29 Modified source dependency tree of Ex. 10: 'Shouting for help seemed appropriate .'

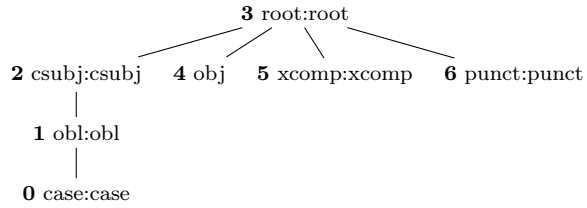

Fig. 30 Modified target dependency tree of Ex. 10: 'Om hulp roepen leek me gepast .' Note the unalgined `obj` node.

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 31). 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 $1/6.5 = 0.15$.

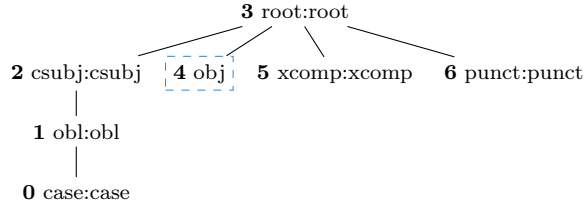

Fig. 31 A visualisation of the edit (insertion, the dashed blue box) to go from the modified source tree in Figure 29 to the modified target tree in Figure 30.

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 5.

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

Table 5 Summary of the results of all metrics for Ex. 10 (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 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 5 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.

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 et al., 2019a)[9]. We built mixed-effect models in R (R Core Team, 2019), using the lme4 package (Bates et al., 2015) with lmerTest (Kuznetsova

[9] 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).

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 using the effects package (Fox and 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).[10] 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 32. This indicates that translators needed more time to translate those source segments that required more label changes when translating.

[10] 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.

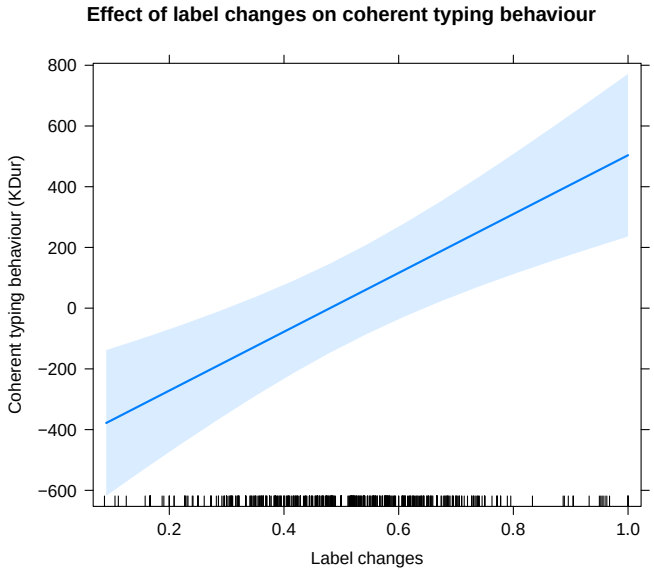

Fig. 32 Effect plot for the main effect of label changes on coherent typing behaviour.

Source reading time (TrtS) was best predicted by SACr only, although the model which included both participants and items as random effects gave 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 33. 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.

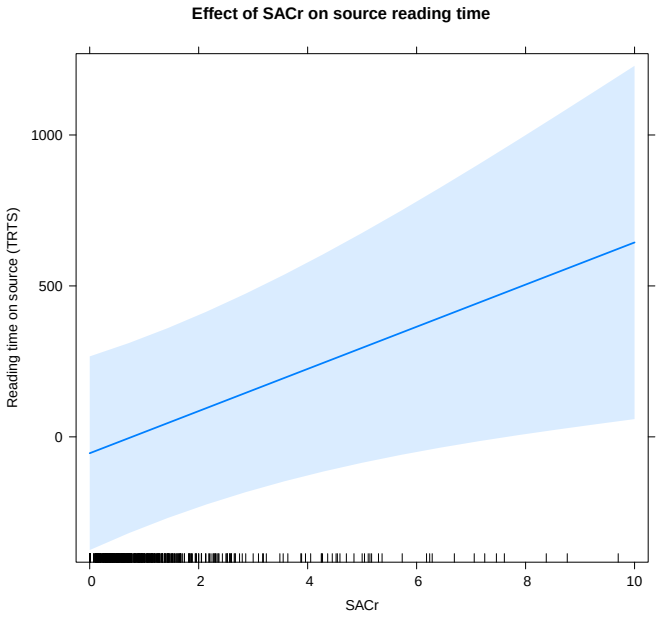

Fig. 33 Effect plot for the main effect of SACr on source text reading time.

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 6. The interaction effect is visualised in Figure 34. 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.

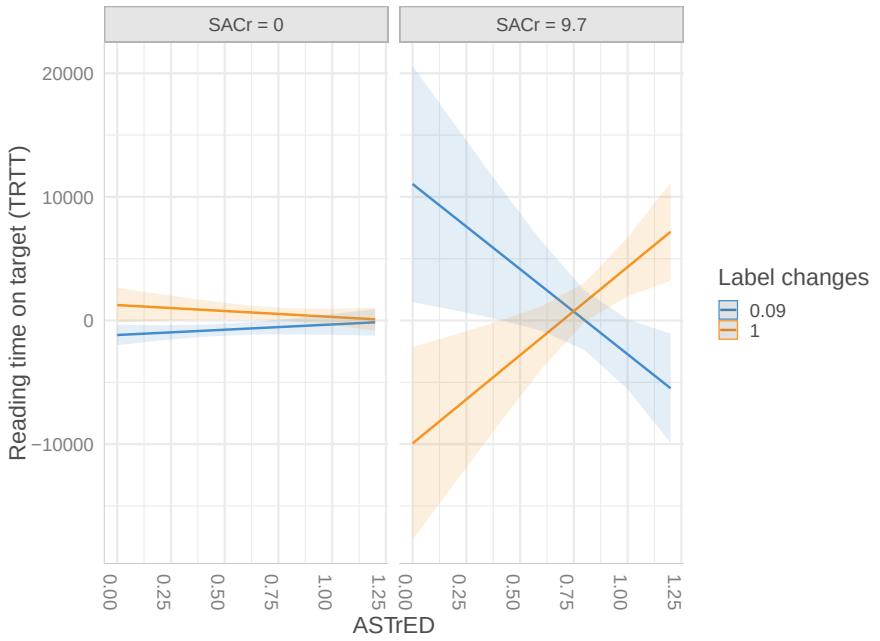

Fig. 34 Effect plot for the three-way interaction effect of ASTrED, label changes, and SACr on target reading time.

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 **

Table 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 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

Table 7 Kendall correlation between normalised metrics: ASTrED, label changes, and SACr ($p < .01$).

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 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.

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 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 et al., 2019a,b). 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 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.

References

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

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

Bangalore S, Behrens B, Carl M, Ghankot M, Heilmann A, Nitzke J, Schaeffer M, Sturm A (2015) The role of syntactic variation in translation and post-editing. Translation Spaces 4(1):119–144

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

Borrillo JM (2000) Register analysis in literary translation: A functional approach. Babel 46(1):1–19, DOI 10.1075/babel.46.1.02bor

Campbell S (1999) A cognitive approach to source text difficulty in translation. Target 11(1):33–63

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

Carl M, Schaeffer MJ (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 MJ, Bangalore S (2016) The CRITT translation process research database. In: Carl M, Bangalore S, Schaeffer MJ (eds) New direc-

tions in empirical translation process research, New frontiers in translation studies, Springer, Cham, Switzerland, pp 13–54

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

Catford JC (1965) A linguistic theory of translation: An essay in applied linguistics. Oxford University Press

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

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

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: O'Brien S, Simard M, Specia L (eds) MT Summit XIV Workshop on Post-editing Technology and Practice, Proceedings, European Association for Machine Translation, pp 63–71

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, URL http://dx.doi.org/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, URL http://dx.doi.org/10.1017/S1351324912000344

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

Dyer C, Chahuneau V, Smith NA (2013) A simple, fast, and effective reparameterization of IBM model 2. In: Proceedings of NAACL-HLT 2013, Association for Computational Linguistics, Atlanta, Georgia, USA, pp 644–648

Fox J, Weisberg S (2019) An R companion to applied regression, 3rd edn. Sage, Thousand Oaks CA

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), Montréal, Québec, Canada, pp 49–57

Gunning R (1952) The technique of clear writing. McGraw-Hill, New York

Hajič J, Zeman D (eds) (2017) Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Vancouver, Canada, DOI 10.18653/v1/K17-3, URL https://www.aclweb.org/anthology/K17-3000

Hansen-Schirra S, Nitzke J, Oster K (2017) Predicting cognate translation. Empirical Modelling of Translation and Interpreting 7:3

Jurafsky D, Martin JH (2008) Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing. Upper Saddle River, NJ: Prentice Hall

Kay M, Roscheisen M (1993) Text-translation alignment. Computational Linguistics 19(1):121–142

Kincaid JP, Fishburne RP, Rogers RL, Chissom BS (1975) Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Research branch report RBR-8-75, Naval Technical Training Command Millington Tenn Research Branch, Springfield, Virginia

Koster J (1975) Dutch as an SOV language. Linguistic Analysis pp 111–136

Kromann M (2003) The Danish dependency treebank and the DTAG treebank tool. In: Proceedings of the 2nd International Workshop on Treebanks and Linguistic Theories

Kuznetsova A, Brockhoff PB, Christensen RHB (2017) lmerTest package: Tests in linear mixed effects models. Journal of Statistical Software 82(13):1–26, DOI 10.18637/jss.v082.i13

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

de Marneffe MC, Manning CD (2008) The Stanford typed dependencies representation. In: Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, Coling 2008 Organizing Committee, Manchester, UK, pp 1–8, URL https://www.aclweb.org/anthology/W08-1301

Matthews P (1981) Syntax. Cambridge Textbooks in Linguistics, Cambridge University Press

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, Association for Computational Linguistics, Edmonton, Canada, vol 3, pp 1–10, DOI 10.3115/1118905.1118906

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), Sofia, Bulgaria, pp 346–351

Nivre J (2015) Towards a universal grammar for natural language processing. In: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, pp 3–16, URL https://link.springer.com/chapter/10.1007/978-3-319-18111-0_1

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

Nivre J, De Marneffe MC, Ginter F, Goldberg Y, Hajic J, Manning CD, McDonald R, Petrov S, Pyysalo S, Silveira N, et al. (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

Och FJ, Ney H (2000) A comparison of alignment models for statistical machine translation. In: Proceedings of the 18th conference on Computational Linguistics, Association for Computational Linguistics, Saarbrücken, Germany, vol 2, pp 1086–1090, DOI 10.3115/992730.992810

Och FJ, Ney H (2003) A systematic comparison of various statistical alignment models. Computational Linguistics 29(1):19–51, DOI 10.1162/089120103321337421

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

Pawlik M, Augsten N (2015) Efficient computation of the tree edit distance. ACM Transactions on Database Systems 40(1), DOI 10.1145/2699485, URL http://dl.acm.org/citation.cfm?doid=2751312.2699485

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, URL https://linkinghub.elsevier.com/retrieve/pii/S0306437915001611

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, Springer, pp 760–771

Pym A (2014) Exploring translation theories, 2nd edn. Routledge, London; New York

Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD (2020) Stanza: A Python natural language processing toolkit for many human languages. 2003.07082

R Core Team (2019) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/

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, Association for Computational Linguistics, Gothenburg, Sweden, pp 29–37, DOI 10.3115/v1/W14-0306

Schwarm SE, 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), Ann Arbor, Michigan, USA, pp 523–530, DOI 10.3115/1219840.1219905

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, Association for Computational Linguistics, pp 88–95, DOI 10.3115/974557.974571, URL https://www.aclweb.org/anthology/A97-1014

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 (2015) Measuring translation difficulty: Theoretical and methodological considerations. Across Languages and Cultures 16(1):29–54, DOI 10.1556/084.2015.16.1.2, URL http://www.akademiai.com/doi/abs/10.1556/084.2015.16.1.2

Sun S, Shreve GM (2014) Measuring translation difficulty: An empirical study. Target 26(1):98–127, DOI 10.1075/target.26.1.04sun, URL https://benjamins.com/online/target/articles/target.26.1.04sun

Tirkkonen-Condit S (2005) The monitor model revisited: Evidence from process research. Meta: Journal des Traducteurs/Meta: Translators' Journal 50(2):405–414

Vanroy B, De Clercq O, Macken L (2019a) Correlating process and product data to get an insight into translation difficulty. Perspectives 27(6):924–941, DOI 10.1080/0907676X.2019.1594319, URL https://doi.org/10.1080/0907676X.2019.1594319

Vanroy B, Tezcan A, Macken L (2019b) Predicting syntactic equivalence between source and target sentences. Computational Linguistics in the Netherlands Journal pp 101–116, URL https://www.clinjournal.org/clinj/article/view/95

Zeman D, Hajič J, Popel M, Potthast M, Straka M, Ginter F, Nivre J, 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, Association for Computational Linguistics, Brussels, Belgium, pp 1–21, DOI 10.18653/v1/K18-2001, URL https://www.aclweb.org/anthology/K18-2001

Zwart CJW (1994) Dutch is head-initial. The Linguistic Review 11(3-4), DOI 10.1515/tlir.1994.11.3-4.377