ACL 2020

The 58th Annual Meeting of the Association for Computational Linguistics

Proceedings of the Conference

July 5 - 10, 2020

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-952148-25-5

Message from the General Chair (TODO)

Message from the Program Chairs (TODO)

Organizing Committee

Table of Contents

Towards a More Natural Controlled Language in Future Airbus Cockpits. A Psycho-linguistic Evaluation

Nataly Jahchan[1], Anne Condamines[2], Emmanuelle Cannesson[3], and Hélène Giraudo[4]

[1,3]Airbus Operation SAS
[1,2,4]CLLE-CNRS
{nataly.jahchan,emmanuelle.cannesson}@airbus.com
{anne.condamines,helene.giraudo}@univ-tlse2.fr

Abstract

The main goal of this research is to optimize an existing Airbus Cockpit Controlled Language in order to integrate it in future cockpit design. The current controlled language used aboard Airbus cockpit interfaces was carefully constructed to avoid ambiguity and complexity. In order to optimize the existing language, we set out to evaluate the appropriate levels of simplification that would achieve more accurate and faster comprehension with optimized pilot training time by using psycho-linguistic experimentation and cognitive science tools. We present in this paper a congruency task similar to traditional judgment tasks in behavioral experiments. It provides a firmly controlled environment to test linguistic hypotheses and CNL rules. Results show that what we sometimes mistakenly label as superfluous or empty syntactical elements could go a long way in ensuring better comprehension and faster information processing from a psycho-linguistic point of view.

1 Introduction

The main goal of this research is to optimize an existing Airbus Cockpit Controlled Language in order to integrate it in future cockpit design. The current controlled language used aboard Airbus cockpit interfaces was carefully constructed to avoid ambiguity and complexity (as are all comprehension oriented controlled languages, (Kuhn, 2014; Schwitter, 2010; Kitteridge, 2003) and is designed to help pilots operate and navigate the aircraft (with the help of cockpit screen interfaces) in normal and abnormal (in cases of emergency or failures) situations. The need for clear and unambiguous communication is vital in safety critical domains. This controlled language and the rules that make it were put in place at a time when design flexibility was limited (for example small screen sizes that restrict word and sentence length (Spaggiari et al., 2003; Jahchan et al., 2016; Jahchan, 2017). This results in a CNL which is non-conforming to natural language syntax, highly abbreviated, typographically variable, and color-coded (Jahchan, 2019). As we are addressing a more flexible disruptive cockpit design for future aircraft, these limitations are no longer immutable constraints, and the future controlled language need not be so coded and compact, or follow very strict simplification rules.

The goal being to take into consideration the disruptive cockpit design (possibly larger screen sizes (less character limitations), newer technology, etc.) which goes hand in hand with an adapted human-oriented controlled language and which is safe, suitable and easily accessible for a human operator.

Therefore, in order to optimize the existing language, we set out to evaluate the appropriate levels of simplification that would achieve more accurate and faster comprehension with optimized pilot training time by using psycho-linguistic experimentation and cognitive science tools. In order to determine the appropriate levels of simplification, one must carefully investigate the problem in context (operational piloting constraints, cockpit design constraints, linguistic ambiguities (syntactic, semantic, and terminological ones). In this sense, we are more particularly dealing with Ergonomic Linguistics (Condamines, 2021) in which linguistic models, theories, and hypotheses are used in specified work contexts (mainly in industry) to achieve precise goals efficiently and serve a real life operational purpose (one of the primary uses of Human-oriented CNLs). These hypotheses and propositions are derived from real language productions and theoretical linguistic theories (for example common CNL construction rules among several languages (O'Brien's, 2003) and should be evaluated using experimental techniques and acceptability tests to acquire empirical evidence to support their efficiency when it comes to comprehension and optimal performance for human operators (target users of CNLs). This concept is closely related to readability and usability. Our own definition of readability for the purposes of this research does not involve the traditional definition, i.e. ease of reading, reading proficiency, or the characteristics that make readers willing to carry on reading (Flesch Kincaid, Smog formula, (Flesch 1979), etc.). Readability in our sense is about usability of the text. Usability is defined as the *"extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use"* (ISO/DIS 9241-11.2 :2016).

To this date, CNL evaluations are not systematically enforced and very rarely put in place for human-oriented CNLs. There have been some evaluations of CNLs using NLP (natural language processing) tools in corpus linguistics-based approaches such as the verification of conformity of requirements (Condamines and Warnier, 2014; Warnier, 2018) or for text complexity (Tanguy and Tulechki, 2009), and machine translation (O'Brien and Roturier, 2007; Aikawa et al., 2007), or for syntactic transformations and corpus alignment of specialized corpora with existing simplified corpora (Cardon and Grabar 2018), etc. There have also been evaluations based on ontographs for knowledge representation and formal languages (Kuhn, 2010). In this paper, Kuhn (2010) contends that *"user studies are the only way to verify whether CNLs are indeed easier to understand than other languages"*. He argues that it is difficult to obtain reliable approaches with task-based and paraphrase-based evaluation approaches, and offers an alternative method for evaluating formal logic-based languages. Consequently, existing CNL research falls short on providing empirical proof on the effectiveness of comprehension-oriented CNLs on the human cognitive processes of language comprehension, for instance by measuring reaction times and accuracy in performance. We argue that the relative lack of cognitive behavioral evaluations is equivalent to rendering CNLs mere style guides or good authoring practices, and the reasons for adopting certain rules over others are unreliable.

Uncontrolled natural language is ambiguous and unsuitable for use in domains where ambiguity may be dangerous such as the aviation industry, but on the other hand, it represents an intricate part of our cognitive processes and its rules must not be excluded. Readability, text simplification, and text complexity research have focused on simplifying the language by making it less and less like natural language, and more like an unambiguous set of codes and regulations so that the resulting language veered away from the "natural" dimension. But to what extent is text simplification satisfactory and what are the limits at which it becomes counter-productive? When must natural language structures be respected? We constructed a more natural controlled language (MNL) by basing ourselves on the existing more codified controlled language (MCL) and its operational needs, syntactic and terminological rules) by using research that has been done on readability and text complexity and test, bit by bit, how we can add sentential elements that would make the language closer to natural language structure of English. At the same time, by adding a sentence structure we would be limiting the different possible interpretations, therefore avoiding, as much as possible, elliptical ambiguities (C.f Figure 1)

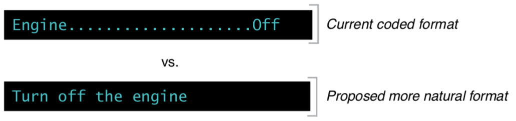
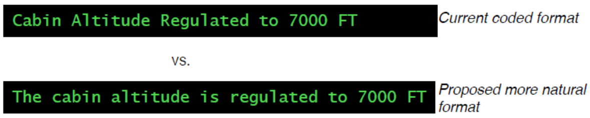


Figure 1, Example of MCL and MNL

Although pilots are trained to understand the meaning of the typographical ellipses (dots separating "engine" and "off" and color coding to mean an action that must be performed, the sentence structure (in the proposed more natural format) provides a fail-safe way of avoiding ambiguity. The sentence "Turn off the engine" adds two more words to the original statement "engine.....off " yet completely eliminates the second possible interpretation (the engine is off). Thus, information is solely contained in the linguistic elements, excluding color and typographical separation. There is only one possible way of interpreting and understanding the second sentence. In this way, we based ourselves on the MCL corpus (operational use and context, goal) and created new more natural structures (MNL) to be evaluated.

2 Method

As a first approach, we used congruency tasks to evaluate passive comprehension. To be able to use congruency tasks (commonly used in cognitive psychology experimentation) we had to limit ourselves to the use of the "information category" in our corpus, and more particularly, the constative messages informing pilots of the availability of a certain function such as "Galleys extraction available in Flight" or "Expect high cabin rate". These sentences do not require direct action but comprehension and awareness on the pilot's end (c.f Figure 2).



Cabin Altitude Regulated to 7000 FT

Current coded format

vs.

The cabin altitude is regulated to 7000 FT

Proposed more natural format

Figure 2, Example of MCL and MNL in an Informational Statement

2.1 Construction of Messages

In the following example case, the original coded and abbreviated message is L TK 17000 KG MAX AVAIL which when decoded without abbreviations means "left tank 17000 kilograms maximum available". It was relatively easy to construct the MCL messages since we could keep the same structure and same words when possible, and find or construct an image that is congruent to its meaning. However, constructing the equivalent MNL messages was a little more complicated as we had several options; there was at least 4 different ways of writing the sentence in the previous example in a more natural language (cf. Jahchan, 2019).

1. There are maximum 20 kilos available in the left container

2. There are 20 kilos maximum available in the left container

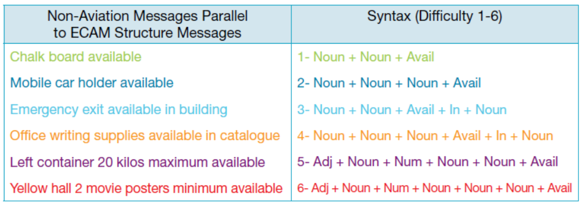
3. The left container has maximum 20 kilos available

4. The left container has 20 kilos maximum available

After careful consideration and in order not to multiply variables, we chose the first option for the MNL structure as the existential clause "there is/are" introduced by the expletive pronoun "there" + predicate "are" indicates the existence or the presence of something in a particular place or time, which in our experiment reinforced the idea of something available or not available in the target picture. The existential clause itself expresses a predicate of existence which sets the tone for the incoming noun phrase. While the second option also includes an existential clause, it was not deemed sufficiently plausible by English native speakers that we consulted. The existential clause introduced in the MNL structures also inverts the theme and rheme structure of the original MCL structure. The current controlled language uses the theme at the onset of the message "left container" followed by the rheme. One of the main differences between both languages is the addition of function words in the MNL stimuli. Leroy et al. (2010) affirms in a study about the effects of linguistic features and evaluation perspectives that *"complex noun phrases significantly increased perceived difficulty, while using more function words significantly decreased perceived difficulty. [...] Laypersons judged sentences to be easier when they contained a higher proportion of function words. A high proportion of function words leads to a different cadence closer to spoken language. It may also help space out individual concepts in text to facilitate assimilation."*

2.2 Stimuli

We created a new corpus of messages inspired by everyday life situations to test our hypothesis with naïve participants that are not familiar with aeronautical corpus terms. An example of this sentences is "parking spot is available", that emulate the syntax and intentions of our original corpus statements. As a first step, the newly proposed structures were purposefully tested on naïve participants (and not pilots) to avoid expert bias and determine comprehension and performance levels on a more general level. The corpus was divided in 6 difficulty categories that represent syntactical structure of the information availability statements. They went from 1 the easiest structure (noun + nous + available) to 6 most difficult (noun + noun + noun + available + in +noun) as length has been proven to be an effective and efficient index of syntactic difficulty Szmrecsanyi (2004). According to Szmrecsanyi (2004), sentence length (or a version of the Flesch-Kincaid tests) are as good a means of testing syntactic text complexity as counting syntactic nodes in a sentence. Szmrecsanyi reports comparing three methods of measuring syntactic complexity node counts, word counts, and 'Index of Syntactic Complexity' (which takes into consideration the number of nouns, verbs, subordinating conjunctions, and pronouns). She concludes that the three measures are near perfect proxies since they significantly correlate and can be used interchangeably. Once the messages were set, we looked for, constructed, or modified existing real life images which accurately portrayed the messages we previously concocted, which have similar syntactic structure and difficulty as messages present in the original corpus (MCL), and for which we created a corresponding MNL version (c.f. Figure 3)



Non-Aviation Messages Parallel to ECAM Structure Messages	Syntax (Difficulty 1-6)
Chalk board available	1- Noun + Noun + Avail
Mobile car holder available	2- Noun + Noun + Noun + Avail
Emergency exit available in building	3- Noun + Noun + Avail + In + Noun
Office writing supplies available in catalogue	4- Noun + Noun + Noun + Avail + In + Noun
Left container 20 kilos maximum available	5- Adj + Noun + Num + Noun + Noun + Avail
Yellow hall 2 movie posters minimum available	6- Adj + Noun + Num + Noun + Noun + Noun + Avail

Figure 3, Example of 6 conditions of difficulty

As messages were different in length, the allotted reading time was different depending on the number of words. MNL messages necessarily have more words than MCL messages. However, those words were only grammatical words such as "there is" or "a", or "the", etc. We decided to count only lexical words to calculate reading time. This choice might have inadvertently given a position of privilege to the MCL messages since MNL messages had more total words (grammatical and lexical) than the equivalent MCL messages yet they had the same reading time (same number of lexical words). We based ourselves on word per minute and reading time research to calculate the time the messages appeared on the screen (Trauzettel-Klosinski Dietz, 2012).

2.3 Experimental Design and Participant Task

Before beginning the experiment, participants filled out different forms: a general ethics and compliance consent form, a data sheet in which they specified their age, gender, dexterity, native language, English placement, knowledge of Airbus Control Language. All non-native English speakers also performed a quick English placement test online to determine their CEFR levels (Common European Framework of Reference for Languages). The levels range from A1 or breakthrough/ beginner to C2 or Mastery/Proficiency.

Participants started with a practice session composed of a different set of 24 semi-randomized stimuli representative of the difficulty and language conditions, and the same image construction methodology as the target stimuli in the main lists. They had noise cancelling headphones and were set in a quiet room with no distractions. Each list consisted of 48 target stimuli, split into 24 congruent stimuli (image congruent with the message, correct answer is a "yes") and 24 incongruent stimuli (image incongruent with the message, correct answer is a "no"). Participants had 5000 ms to respond. This time lapse was validated by doing several pretests to ascertain the adequate display time for reading the messages. In case of a non-answer the next stimulus appears and so on. Once the participant responds the image disappears and the next fixation cross appears. The task consisted of the participants reading a text written in either the More controlled Language (MCL) syntax or the More Natural Language (MNL) syntax (c.f. Figure 4)

The messages appear out of context preceded only by a 3000 ms fixation cross in the middle of the screen. We decreased that value to 150 words per minute (WPM), so that a message that has 3 lexical words would appear for 1.2 seconds (3 x 60/150) and a message that has 6 lexical words would appear for 2.8 seconds (6 x 60/150), etc. The text (the prime) then disappears and a target image appears, an image which could be congruent with the previously read text or incongruent. I.e. if the text says "bus stop available" and the image shows a bus stop then the participant has to press "yes" on the controller to indicate congruency, and if for instance the image shows an image of a car then the participant should press on "no" to indicate that the image is incongruent with the text.

Response times and precision in both language conditions were recorded. We chose sentences that could show an accurate visual description of a situation or scene.

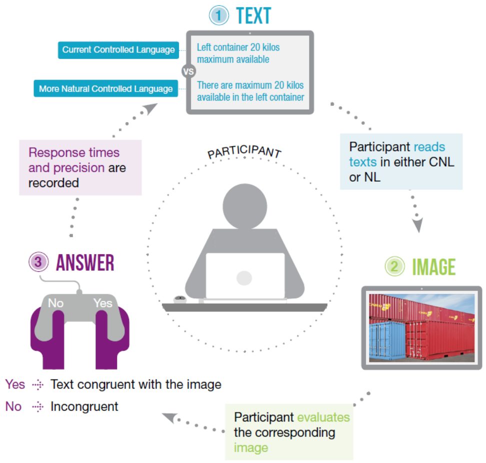

1 TEXT
Current Controlled Language
More Natural Controlled Language
VS
Left container 20 kilos maximum available
There are maximum 20 kilos available in the left container
Participant reads texts in either CNL or NL
2 IMAGE
Participant evaluates the corresponding image
3 ANSWER
No
Yes
Yes Text congruent with the image
No Incongruent
Response times and precision are recorded
PARTICIPANT

Figure 4, Representation of Task Performance

2.4 Participants

72 participants took part in the first experiment (12 native speakers of English and 60 non-native speakers whose placement levels ranged from A1 to C2 in CEFR). The non-native speakers' languages included Arabic, Chinese, Dutch, French, German, Portuguese, Spanish, Serbian, and Indonesian, with the overwhelming majority being French (45 out of 60). 38 participants had no knowledge whatsoever of controlled languages. 16 claimed had beginner knowledge of the Airbus controlled language (Airbus employees having rarely worked with the language or its rules). 14 had a more intermediate knowledge of the language. 5 participants had expert knowledge of the language as it could be part of their daily task.

2.5 Experimental Materials and Equipment

DMDX is a Win 32-based display system used in psychological laboratories to measure reaction times to visual and auditory stimuli. We used this software on a Dell Precision 3510 laptop to display the messages and images. For that, we developed 6 scripts which consisted of 3 semi-randomized lists of stimuli for right-handed participants and 3 for left-handed participants (same lists but the "yes" and "no" buttons were inverted for left handed participants).

2.6 Variables

The list of independent variables that we will evaluate are:

- Language (MCL-MNL)
- Syntactic Difficulty (1 to 6)
- Type (Congruents-Incongruents) Extraneous and participant variables:
- English placement level (Basic Intermediate, Proficient, Mastery, Native)
- Familiarity with Airbus CL (None, Beginner, Intermediate, Expert)

Dependent variables:

- Reaction time in ms, Accuracy (number of errors)

2.7 Hypotheses and Research Questions:

1. MNL messages produce shorter reaction times than MCL ones in different syntactic difficulty conditions.
2. MNL messages produce less errors (are more accurate) than MCL ones in different syntactic difficulty conditions.
3. Did the language factor play a different role for the different types of congruency responses regarding reaction times?
4. Did the language factor play a different role for different levels of English placement (Basic Intermediate, Mastery, Natives) regarding reaction times?

3 Results and Statistical Analysis

We reported the results below linked to each of the previously mentioned hypotheses. We used non-parametric statistical significance tests such as Wilcox signed rank as the data had a non-normal distribution (Gaussian distribution). These tests help determine whether the independent variables had an effect on reaction time and accuracy of comprehension (dependent variables) by calculating a statistical significance p-value (results are significant if they show a p-value less than 0.05, i.e. implying that it is acceptable to have less than 5% probability of incorrectly rejecting the true null hypothesis). **1. MNL messages produce shorter reaction times than MCL**

ones in different syntactic difficulty conditions. A Shapiro-wilk normality test was run on the reaction times and the results showed that the data is significantly non-normal (p = 2.054e-05) with abnormal skew, therefore we used non-parametric tests to test the main effect such as the Wilcox signed rank test because the same participants took part in both language conditions. Firstly, the general effect was compared regardless of difficulty for both language conditions. There was a significant difference in the scores for MCL (Median=2030.317 ms.) and MNL (Median= 1944.163 ms.) conditions; v=1692, p=0.0339, effect size calculated with Pearson's coefficient r=0.24998. With the hypothesis confirmed, we can conclude that the more natural language helped participants process the stimuli and provoked significantly faster reaction times than the more coded language format.

We then performed a linear regression model to ascertain the influence of the syntactic difficulty condition in both languages. A simple linear regression was calculated to predict the reaction times of the MCL responses based on the 6 syntactic difficulty conditions. A significant regression equation was found (F(1,1500) = 9.211, p < 0.002447), with an R2 of 0.006103. Participants' predicted reaction times is equal to 1873.77 + 42.55 ms for every additional difficulty condition. Therefore, reaction time increased 42.55 ms for each additional difficulty condition. A simple linear regression was also calculated to predict the reaction times of the MNL responses based on the 6 difficulty conditions. A significant regression equation was found (F(1,1450) = 12.68, p < 0.0003822), with an R2 of 0.008667. Participants' predicted reaction times is equal to 1801.64 + 47.81 ms for every additional difficulty condition. Therefore, reaction time increased 47.81 ms for each additional difficulty condition. Figure 5 is the graph that plots those two linear regression models for both languages in the 6 difficulty conditions. As we can see there is no interaction between the two languages (lines are parallel and do not intersect) but reaction times get slower when difficulty increases in both languages which confirms that syntactic difficulty based on length is a valid measure (confirms Szmrecsanyi (2004) findings). With the hypothesis confirmed, we can also conclude that MNL messages produced consistently faster reaction times than MCL messages in all difficulty conditions.

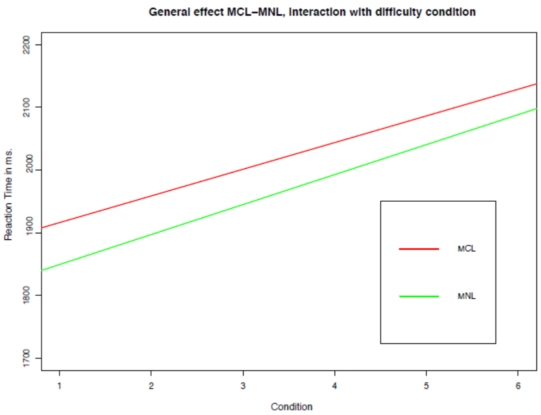


Figure 5, Linear Regression Models for MCL and MNL Difficulty Condition

2. MNL messages produce less errors (are more accurate) than MCL ones in different syntactic difficulty conditions. Accuracy was calculated using the average number of errors. Therefore, we started by comparing the general effect of accuracy regardless of difficulty for both language conditions using the Wilcox signed Rank test. There was no significant difference in the number of errors by subject produced in the MCL (Mean = 2.46 errors) and MNL (Mean = 2.9 errors) conditions; v = 549, p = 0.07121. We could interpret this by proposing that the difference in the syntax of the two languages was not different enough (a lot of the stimuli had only one or two grammatical articles added to them) to cause one language to have better performance with respect to errors, but those subtleties were manifested in the reaction times instead which stand to be more adequate measures of early/initial comprehension. Figure 6 is a histogram plot of the errors made in the different conditions of difficulty for both languages. As we can see the number of errors in both languages is not consistent across different difficulty conditions, but there is a tendency for both languages to have more and more mistakes as difficulty increases. The advance that the MCL has over the MNL in the easy difficulty conditions (probably due to having less words to read and the same time as MNL stimuli with more words to read) disappears the harder the stimuli get with the exception of mid-way difficulty level 4.

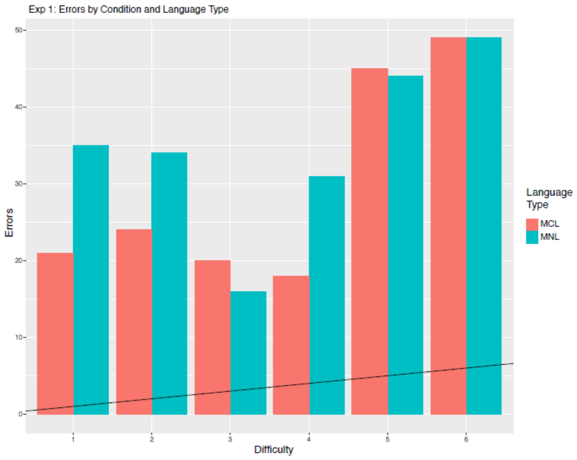


Figure 6, Histogram of errors in MNL and MCL in the 6 difficulty conditions

3. Did the language factor play a different role for the different types of congruency responses regarding reaction times? It was important to verify whether there was an effect of congruent stimuli versus incongruent stimuli (to the corresponding image) since congruent stimuli were deemed easier targets than incongruent ones, therefore understanding incongruent stimuli constitutes an extra difficulty condition in and of itself. To illustrate this with a concrete example: An image that shows an empty parking lot with a message that reads "Parking is available" is easier to interpret as a "yes congruent" than an image showing a desk lamp with a message that reads "Ceiling lamp is available" as a "no, incongruent". Confusion might arise from the presence of a lamp in the picture but which is not a ceiling lamp. Most incongruent images were purposefully chosen to include a little forced ambiguity, or an extra "trick" where the participant had to verify thoroughly the image before responding. Therefore, we compared the general effect of reaction times regardless of difficulty for congruent stimuli in both language conditions using the Wilcox signed Rank test. There was no significant difference in reaction times of the congruent stimuli produced in the MCL (Median = 1888.502 ms) and MNL (Median = 1879.167 ms) conditions; v = 1468, p = 0.3875. However, when performing the same test for the incongruent stimuli we found a significant difference in the MCL (Median = 2241.473ms) and the MNL (Median = 1927.541ms) conditions; v = 1475, p = 0.0308. As we can see from Table 2 the difference between medians in the incongruent condition is far superior than the congruent one and is statistically significant. We attribute this difference to the added difficulty in the interpretation of the incongruent stimuli, and we conclude that the MNL syntax helps process information faster than the MCL condition as the difficulty in the task and stimuli increase.

MCL Congruent	MNL Congruent	Difference	MCL Incongruent	MNL Incongruent	Difference
1888.502	1879.167	9.335	2241.473	1927.541	+313.932

Figure 7, Medians in ms of MCL and MNL reaction times in congruent and incongruent stimuli

4. Did the language factor play a different role for different levels of English placement (Basic Intermediate, Mastery, Natives) regarding reaction times?

We grouped the English placement levels into 3 categories. "Basic intermediate" regroups participants that were placed from levels A2 to C1, "Mastery" has participants that were placed in C2 level and "native" are the native English speaker participants. We did a series of t-tests (as reaction times for those sub-groups were not significantly non-normal so we could use a parametric test) to compare the two different language conditions in each of the English placement groups. For basic intermediate level, there was a significant difference in the scores for MCL (Mean = 2246.322 ms) and MNL (Mean = 2144.104 ms) conditions; t = 2.5416, p = 0.01644. For mastery level, there was no significant difference in the scores for MCL (Mean=1956.563ms) and MNL (Mean= 1954.745ms) conditions; t = 0.034395, p = 0.9728. For native level, there was no significant difference in the scores for MCL (Mean = 1690.904 ms) and MNL (Mean = 1588.062 ms) conditions; t = 1.8301, p = 0.09444. As we can see the only significant result is the basic intermediate level. We can conclude that MNL helps comprehension for the weaker levels of English levels as reaction times are significantly shorter for that group. While the native group does not show statistical significance, most probably because the group is made up of 12 participants only, it is interesting to note the difference in the average of the MNL and MCL which is equal to the difference for lower intermediates (averages which showed statistical significance). Native speakers often mentioned that they preferred the more natural language, and this is also apparent in their results. A simple linear regression was also calculated to predict the reaction times of the MCL responses based on the 3 English placement levels. A significant regression equation was found (F(2,432) = 21.83, p = 9.275e-10), with an R2 of 0.0918. Participants' predicted reaction times is equal to 2221.92 – 280.14 ms for every English placement level gained. Therefore, reaction time decreased 280.14 ms for every English placement level gained (cf. Figure 8).

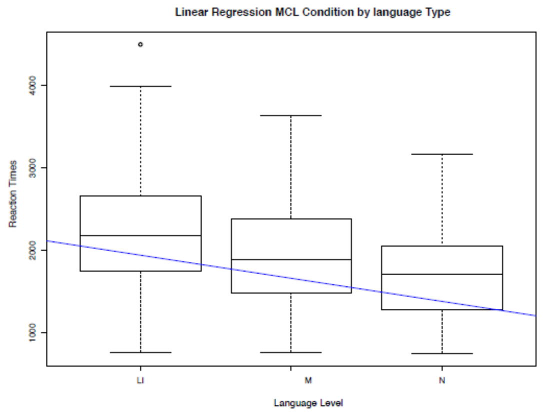


Figure 8, Linear regression of MCL in the different English Placement Levels

A simple linear regression was calculated to predict the reaction times of the MNL responses based on the 3 English placement levels. A significant regression equation was found (F(2,430) = 21.38, p = 2.288e-10), with an R2 of 0.0981. Participants' predicted reaction times is equal to (2146.50 ms – 190.20 ms) for every English placement level gained. Therefore, reaction times decreased 190.20 ms for every English placement level gained (cf. Figure 39).

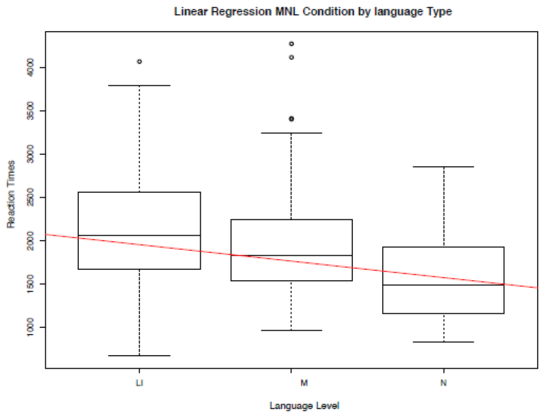


Figure 9, Linear regression of MNL reaction times in the different English Placement Levels

A graphical representation of both of those linear regressions is shown in Figure 10. As we can see, there is no interaction between these two languages for all three English level placements, but they both show decreasing reaction times with every additional level of English placement. The MNL proves to have consistently faster reaction times in all English placement levels, and therefore, we can conclude that MNL helps comprehension and information processing more than MCL regardless of participants' English placement level.

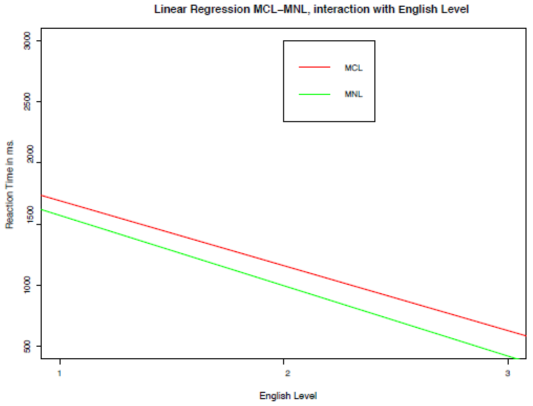


Figure 10, Linear regression for both MCL and MNL reaction times in the different English Placement Levels

4 Discussion

As shown in the results of hypothesis 1, MNL condition shows significantly faster reaction times than MCL condition, and both languages performed equally with regards to accuracy (in hypothesis 2). This could be explained by the fact that the syntactic changes (sentential elements in constative statements) between the two language conditions did not have enough disparities to warrant observable differences in accuracy, whereas the observed differences in reaction times were able to highlight the subtle syntactic variations that led to faster comprehension. In the experiment speed of stimuli presentation and to a certain degree the stress it provoked, accentuated the role of the more natural language in information processing. Additionally, there was no interaction between the two languages with regards to the 6 levels of syntactic difficulty, but reaction times get slower when difficulty increases in both languages. We can also conclude that MNL produced consistently faster reaction times than MCL in all syntactic difficulty conditions. As we illustrated in research question 3, incongruent stimuli had an additional touch of difficulty and that is reflected in the reaction times' discrepancies for congruency conditions in both language conditions. Incongruent stimuli showed significantly faster reaction times for the MNL condition over the incongruent MCL condition, while the congruent stimuli did not. Therefore, in cases of increased difficulty the more natural language helps ease comprehension. Concerning English placement levels (research question 4), MNL seems to facilitate comprehension for participants in the basic intermediate level placement, and this suggests that speakers with weaker levels of English proficiency would benefit more greatly from a more natural language than confirmed speakers, or at least we could say that the effect is more conspicuous. While native English speakers performed better on average in the MNL condition, the effect was not statistically significant and should be the object of further studies with bigger samples of native speakers. We could also conclude that there is no interaction between the reaction time of the two language conditions and the different English level placement (one language did not start out having better performance than the other but ended up performing worse in different level placements), however we do observe a downward tendency in reaction times the more proficient speakers become. Natives have significantly faster reaction times than basic intermediate English speakers.

5 Conclusion

We presented in this experiment a congruency task similar to traditional judgment tasks in behavioral experiments. It provided a firmly controlled environment to test linguistic hypotheses and CNL rules, nonetheless, the downside of using such experiments is that we are limited to evaluating passive comprehension, mainly of specific informative statements. It would be quite difficult to evaluate the comprehension of an order or an instruction using traditional judgment tasks. In subsequent experiments, the congruency tasks will be replaced by ecological performance tasks for injunctive statements (participants performed the action required and the accuracy and response times are recorded) which include the urgency factor (speed of stimuli, and stress generated by limited response time). We will also be recruiting more native speaker participants to have a more representative panel of the target population (pilots from all around the globe), and ascertain whether the different syntactic language conditions reflect equally on native and non-native English speakers.

The results from this experiment are somewhat satisfactory as they show that our initial hypothesis is validated in a certain number of conditions. In all cases, contrary to common misconceptions, results showed that more simplification and linguistic economies and ellipses hardly ever led to better performance (MCL conditions did in no condition show significantly better reaction times or accuracy than MNL conditions). Furthermore, this experiment brought us first elements of empirically tested data which question controlled language construction, and the limits of simplification in general. It showed that what we sometimes mistakenly label as superfluous or empty syntactical elements (such as grammatical words as opposed to lexical words) could go a long way in ensuring better comprehension and faster information processing from a psycho-linguistic point of view.

References

Aikawa, T., Schwartz, L., King, R., Corston-Oliver, M., Lozano, C. 2007. *Impact of Controlled Language on Translation Quality and Post- Editing in a Statistical Machine Translation Environment.* Proceedings of the MT Summit XI, 1-7.

Cardon, R., Grabar, N. 2018. *Identification of Parallel Sentences in Comparable Monolingual Corpora from Different Registers..* In Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis (pp. 83-93).

Condamines, A., Warnier, M. . 2014. *Linguistic Analysis of Requirements of a Space Project and Their Con-formity with the Recommendations Proposed by a Con-trolled Natural Language..* In International Workshop on Controlled Natural Language (pp. 33-43). Springer International Publishing.

Condamines, A. 2021. *Towards an ergonomic linguistics: Application to the design of controlled natural languages..* International Journal of Applied Linguistics, 31(1), 18-30.

Flesch, R. 1979. *How to Write Plain English: Let's Start with the Formula.* University of Canterbury.

ISO. 2016. *Ergonomics of human-system interaction: part 11: usability: definitions and concepts*

(ISO/DIS 9241-11.2:2016). German and English version prEN ISO 9241-11:2016.

Jahchan, N., Condamines, A., Cannesson, E. 2016. *To What Extent Does Text Simplification Entail a More Optimized Comprehension in Human-Oriented CNLs?*. In International Workshop on Controlled Natural Language (pp. 69-80). Springer International Publishing.

Jahchan, N. 2017. *The importance of using psycholinguistic tools for CNL evaluations*. Actes de Jetou, 99-105.

Jahchan, N. 2019. *To what extent does text simplification improve human comprehension?: cognitive evaluations for the optimization of the Airbus cockpit controlled language for future aircraft*. (Doctoral dissertation, Universite Toulouse le Mirail-Toulouse II).

Kittredge, R. I.. 2003. *Sublanguages and controlled languages*. In Ruslan Mitkov, editor, The Oxford Handbook of Computational Linguistics (pp. 430-447).

Kuhn, T. 2014. *A Survey and Classification of Controlled Natural Languages. Computational Linguistics*. 40(1) (pp. 121-170).

Kuhn, T. 2010. *An Evaluation Framework for Controlled Natural Languages*. In Norbert E. Fuchs, editor, Proceedings of the Workshop on Controlled Natural Language (CNL 2009), Lecture Notes in Computer Science 5972 (pp. 1-20). Springer, 2010.

Leroy, G., Helmreich, S., Cowie, J. R. 2010. *The effects of linguistic features and evaluation perspective on perceived difficulty of medical text*. In 43rd Hawaii International Conference on System Sciences (pp. 1-10). IEEE.

O Brien, S.. 2003. *Controlling Controlled English: An Analysis of Several Controlled Language Rule Sets*. In Proceedings of EAMT-CLAW, 3 (pp. 105-114), 33.

O Brien, S., Roturier, J. . 2007. *How Portable Are Controlled Language Rules? A comparison of Two Empirical MT Studies*. In Proceedings of MT summit XI (pp. 345-352).

Schwitter, R. 2010. *Controlled Natural Languages for Knowledge Representation*. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters (pp. 1113-1121). Association for Computational Linguistics.

Spaggiari, L., Beaujard, F., Cannesson, E. 2003. A Controlled Language at Airbus. *Proceedings of EAMT-CLAW03 (pp. 151-159)*.

Szmrecsanyi, B. 2004. *On operationalizing Syntactic Complexity. In Le poids des mots.*. Proceedings of the 7th International Conference on Textual Data Statistical Analysis. Louvain-la-Neuve Vol. 2 (pp. 1032-1039)

Tanguy, L., Tulechki, N.. 2009. *Sentence Complexity in French: A Corpus-based Approach.* In Intelligent information systems (IIS) (pp. 131-145).

Trauzettel-Klosinski, S., Dietz, K. 2012. *Standardized Assessment of Reading Performance: the New International Reading Speed Texts IReST.* . Investigative ophthalmology visual science, 53(9) (pp. 5452-5461)

Warnier, M. . 2018. *Contribution de la linguistique de corpus a la constitution de langues controlee pour la redaction technique : l exemple des exigences de projets spatiaux.*. Ph.D. thesis, University of Toulouse - Jean Jaures.

Grammar-Based Concept Alignment for Domain-Specific Machine Translation

Arianna Masciolini
Digital Grammars
arianna@digitalgrammars.com

Aarne Ranta
University of Gothenburg,
Department of Computer Science
and Engineering;
Digital Grammars
aarne.ranta@cse.gu.se

Abstract

Grammar-based domain-specific MT systems are a common use case for CNLs. High-quality translation lexica are a crucial part of such systems, but involve time consuming work and significant linguistic knowledge. With parallel example sentences available, statistical alignment tools can help automate part of the process, but they are not suitable for small datasets and do not always perform well with complex multiword expressions. In addition, the correspondences between word forms obtained in this way cannot be used directly. Addressing these problems, we propose a grammar-based approach to this task and put it to test in a simple translation pipeline.

1 Introduction

Grammar-based translation pipelines such as those based on Grammatical Framework (GF) have been successfully employed in domain-specific Machine Translation (MT) (Ranta et al., 2020). What makes these systems well suited to the task is the fact that, when we constrain ourselves to a specific domain, where precision is often more important than coverage, they can provide strong guarantees of grammatical correctness.

However, lexical exactness is, in this context, just as important as grammaticality. An important part of the design of a Controlled Natural Language (CNL) is the creation of a high-quality translation lexicon, preserving both semantics and grammatical correctness. A translation lexicon is often built manually, which is a time consuming task requiring significant linguistic knowledge. When the task is based on a corpus of parallel example sentences, part of this process can be automated by means of statistical word and phrase alignment techniques (Brown et al., 1993; Och and Ney, 2000; Dyer et al., 2013). None of them is, however, suitable for the common case in which only a small amount of example data is available — typically, with just one occurrence of each relevant lexical item.

In this paper, we propose an alternative approach to the automation of this task. While still being data-driven, our method is also grammar-based and, as such, capable of extracting meaningful correspondences even from individual sentence pairs.

A further advantage of performing syntactic analysis is that we do not have to choose *a priori* whether to focus on the word or phrase level. Instead, we can simultaneously operate at different levels of abstraction, extracting both single- and multiword, even non-contiguous, correspondences. For this reason, we refer to the task our system attempts to automate as *Concept Alignment* (CA). A *concept* is a semantic unit expressed by a word or a construction, which is also a unit of *compositional translation*, where translation is performed by mapping concepts to concepts in a shared syntactic structure.

Conceiving concepts as *lemmas* equipped with morphological variations rather than fixed word forms or phrases allows us to generate translation lexica complete with grammatical category and inflection, so that correct target language forms can be selected in each syntactic context.

This paper is structured as follows. Section 2 starts by giving an overview of our approach to CA and comparing it with related work, followed by a description of our CA algorithm. Section 3 presents the results obtained in a first evaluation of the system. Section 4 summarizes our conclusions and discusses some ideas for future work.

2 Methodology

The objective of CA is to find semantical correspondences between parts of multilingual parallel texts. We call *concepts* the abstract units of translation, composed of any number of words, identified through this process, and represent them as *alignments*, i.e. tuples of equivalent concrete expressions in different languages.

The basic use case for CA, which we refer to specifically as *Concept Extraction* (CE), is the generation of a translation lexicon from a multilingual parallel text. This is analogous to the well-known earlier word and phrase alignment techniques.

An interesting and less studied variant of CA is *Concept Propagation* (CP), useful for cases where a set of concepts is already known and the goal is to identify the expressions corresponding to each of them in a new language, potentially even working with a different text in the same domain. While our system does implement basic CP functionalities, in this paper we focus on its most mature portion: CE. Because results analogous to those that could be obtained via multilingual extraction can be obtained more easily with a combination of CE and CP, we restrict ourselves, for the time being, to bilingual corpora.

As stated in the Introduction, most existing alignment solutions are based on statistical approaches and are, as a consequence, unsuitable for small datasets. Grammar-based approaches, making use of parallel treebanks and collectively referred to as *tree-to-tree alignment methods*, have also been proposed (Tiedemann, 2011), but have historically suffered from the inconsistencies between the formalisms used to define the grammars of different languages and from the lack of robustness of parsers. This work is a new attempt in the same direction, enabled by two multilinguality-oriented grammar formalisms developed over the course of the last 25 years: Grammatical Framework (GF) (Ranta, 2011) and Universal Dependencies (UD) (Rademaker and Tyers, 2019).

GF is a constituency grammar formalism and programming language in which grammars are represented as pairs of an *abstract syntax*, playing the role of an interlingua, and a set of *concrete syntaxes* capturing the specificities of the various natural languages. In the case of translation, similarly to what happens in programming language compilation, strings in the source language are *parsed* to Abstract Syntax Trees (ASTs), which are then *linearized* to target language strings.

UD, on the other hand, is a dependency grammar formalism meant for cross-linguistically consistent grammatical annotation. As opposed to constituency, *dependency* is a word-to-word correspondence: each word is put in relation with the one it depends on, called its *head*, via a directed labelled link specifying the syntactic relation between them. Importantly for our application, the standard format for UD trees, CoNNL-U, gives information not only on the syntactic role of each word, but also on its Part-Of-Speech (POS) tag, lemma, and morphological features.

While both formalisms independently solve the issues related to having to work with grammars that are inconsistent with each other, UD is especially appealing since, being dependency trees an easier target, several robust parsers, such as (Straka et al., 2016) and (Chen and Manning, 2014) are available. Alone, UD trees are sufficient to extract (or propagate) tree-to-tree alignments, but not to automate the generation of a morphologically-aware translation lexicon for a generative grammar. This is where GF comes into play: after correspondences are inferred from a parallel text, our system is able to convert them to GF grammar rules, easy to embed in a domain-specific grammar but also making it immediate to carry out small-scale translation experiments using pre-existing grammatical constructions implemented in GF's Resource Grammar Library (RGL), which covers the morphology and basic syntax of over 30 languages. This is enabled by `gf-ud`, a conversion tool described in (Kolachina and Ranta, 2016) and (Ranta and Kolachina, 2017). Concretely, then, the system we propose consists of a UD parser, an alignment module based on UD tree comparison and a program, based on `gf-ud`, that converts them into the rules of a GF translation lexicon.

2.1 Extracting concepts

The core part of the system outlined above is the alignment module. Its function is to extract alignments from parallel bilingual UD treebanks. The outline of the algorithm is given in the following pseudocode:

```
procedure EXTRACT($criteria$,$(t, u)$)
    $alignments$ = $\emptyset$
    if $(t, u)$ matches any alignment $criteria$ then
        $alignments$ += $(t, u)$
        for $(t', u')$ in SORT(SUBTS($t$)) $\times$ SORT(SUBTS($u$))
do
            extract($criteria$,$(t', u')$)
    return $alignments$
```

Here, the input consists of a list of priority-sorted *alignment criteria*, i.e. rules to determine whether two dependency trees should be aligned with each other, and a pair (t, u) of UD trees to align. An example alignment criterion is sameness of syntactic label, which makes it so that, for instance, subjects are aligned with subjects and objects with objects; the details will be discussed in Section 2.1.1. From an implementation point of view, UD trees are rose trees (trees with arbitrary numbers of branches) where each node represents a word with its dependents as subtrees (see Figure 1). The rose tree is easily obtained from the CoNLL-U notation that UD parsers produce.

As a first step, the program checks whether the two full sentence trees can be aligned with each other, i.e. if they match one or more alignment criteria. In the case of the example criterion discussed above, this means that their roots are labelled the same. If this is the case, they are added to a collection of alignments, which are represented as pairs of UD (sub)trees associated with some metadata, such as the id of the sentence they were extracted from. Such a collection is what the function will return after aligning all the dependency subtrees. The same procedure is applied recursively to all pairs of immediate subtrees of each sentence, until the leaves are reached or alignment is no longer possible due to lack of matching criteria. Subtrees are sorted based on their dependency label to give higher priority to pairs whose heads have the same label (cf. SORT in the pseudocode).

A simple but useful refinement is that, depending on which alignment criteria a pair of trees matches, the heads of the two trees may or may not also be added to the collection of alignments. This is done in order not to miss one-word correspondences that cannot be captured in any other way, for instance between the root verbs of two full sentences. A relevant implementation detail is that, in this context, the head of a tree is not simply defined as its root. Instead, if the root is part of a compound written as two or more separate words or a verb with auxiliaries, the root nodes of the corresponding subtrees are also considered parts of it.

When working on multiple sentences, the algorithm can be applied in an iterative fashion, so that knowledge gathered when a sentence pair is aligned can be used when working on later sentences and to keep track of the number of occurrences of each alignment throughout the entire text. Furthermore, it is possible to initialize the algorithm with a nonempty set of alignments, obtained with the same program or by means of a statistical tool outputting alignments in Pharaoh format and to combine the results of several extraction processes into a single translation lexicon.

2.1.1 Alignment criteria

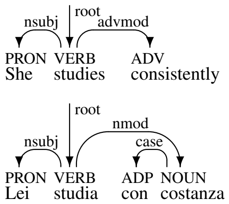
While the alignment criteria are customizable, to allow for a better understanding of the extraction algorithm described above, we explain the criteria that our implementation utilizes by default.

Matching UD labels The most obvious, but also most effective idea is to determine alignability based on comparing the dependency labels of the members of the candidate UD tree pair. In particular, according to this idea, two subtrees in *matching context*, i.e. attached to aligned heads, constitute an alignment if their roots share the same dependency label, meaning that they are in the same syntactic relation with their heads. Note that, since the root of a UD tree is always attached to a fake node with an arc labelled `root`, this criterion also implies that full sentences are always considered to align with each other. This is desirable since we assume that the parallel texts that are fed to our program are sentence-aligned.

Part-Of-Speech equivalence As noted above, the CoNNL-U notation provides information on the grammatical categories of each word, represented as Universal POS tags (Petrov et al., 2012). Intuitively, if the nodes of two trees in matching contexts have the same POS tags, the two trees are more likely to correspond to each other than if not. This is especially true if we focus, for instance, solely on the open class words (defined as in the UD documentation[1]), thus ignoring function words such as prepositions, determiners and auxiliary verbs, which tend to behave differently across different languages. A useful relation to define between dependency trees is, then, that of *POS-equivalence*: two dependency trees t_1, t_2 are POS-equivalent if $M_1 = M_2 \neq \emptyset$, where M_i is defined as the multiset of POS tags of all the open class word nodes of t_i. Applied as a backup for label matching, this criterion can be used to capture correspondences that would otherwise be missed, thus increasing recall, but a decrease in precision is also to be expected. However, since alignment criteria are defined as boolean functions, it is easy to combine them so to that they have to apply simultaneously. This can be useful in cases where precision is more important than recall.

Known translation divergence Parallel texts often present significant, systematic cross-linguistic

[1] `universaldependencies.org/u/pos/all.html`



```
1 She PRON 2 nsubj
2 studies VERB 0 root
3 consistently ADV 2 advmod
```

```
2 studies VERB 0 root
  3 consistently ADV 2 advmod
  1 She PRON 2 nsubj
```

```
1 Lei PRON 2 nsubj
2 studia VERB 0 root
3 con ADP 4 case
4 costanza NOUN 2 obl
```

```
2 studia VERB 0 root
  1 Lei PRON 2 nsubj
  4 costanza NOUN 2 obl
    3 con ADP 4 case
```

Figure 1: The graphical, simplified CoNNL-U and sorted rose tree representation of a pair of UD sentences. With the default criteria, which among other things allow for matching adverbial with adjectival modifiers, the resulting alignments are: ⟨*She studies consistently, Lei studia con costanza*⟩ (matching `root` label), ⟨*studies, studia*⟩ (head alignment), ⟨*she, lei*⟩ (matching `nsubj` label) and ⟨*consistently, con costanza*⟩ (translation divergence; `amod` and `advmod` treated as equivalent).

grammatical distinctions. When this is the case, it is often straightforward to define alignment criteria based on recognizing the corresponding patterns. While many distinctions of this kind are specific to particular language pairs or even stylistical, some of them occur independently of what languages are involved and do not depend on idiomatic usage nor aspectual, discourse, domain or word knowledge. Drawing inspiration from (Dorr, 1994), we refer to them as *translation divergences* and handle some of the most common ones explicitly. For instance, *categorial divergences* occur when the POS tag of a word in the source language changes in its translation. An ubiquitous example of this is that of adverbial modifiers (with the UD label `advmod`) translated as prepositional phrases (with the UD label `obl`, for *oblique*), such as in the English-Italian pair ⟨*She studies **consistently**, Lei studia **con costanza***⟩ (see Figure 1). *Structural divergences*, where a direct object in one language is rendered as an oblique in the other, as in the English-Swedish pair ⟨*I told **him**, Jag berättade **för honom**.*⟩), are also frequently encountered.

Known alignment Another case in which it is trivially desirable for two subtrees in matching context to be aligned is when an equivalent alignment is already known, for instance due to a previous iteration of the extraction algorithm. When referred to pairs of alignments, the term *equivalent* indicates that the two alignments, linearized, correspond to the same string.

At a first glance, this might look like a criterion with no practical applications. However, it can be useful when, instead of starting with an empty set of correspondences, we initialize the program with some alignments that are either inserted manually or, most interestingly, obtained with some other alignment technique. For instance, in this way it is possible to combine the system proposed in this paper with a statistical tool and give more credit to correspondences identified by both.

2.1.2 Pattern matching

So far, we have described how CE can be used in a setting where the objective is to generate a comprehensive translation lexicon based on set of example sentence pairs. We pointed out that the program can be configured to prioritize precision or coverage, but we never restricted our search to a particular type of alignments. However, there are cases in which only certain syntactic structures are of interest: for instance, we might be looking for adverbs or noun phrases exclusively.

To handle such cases, the CE module can filter the results based on a `gf-ud` *tree pattern*. `gf-ud` supports in fact both simple pattern matching, which is integrated in the CE module itself, and pattern replacement[2]. Combining them, for instance by pruning the UD trees extracted by the alignment module, allows us to extract correspondences that cannot be identified by CE alone.

For example, pattern matching can extract verb phrases by looking for full clauses and dropping the subtrees corresponding to subjects. By means of replacements, one can obtain *predication patterns*, i.e. correspondences that specify the argument structure of verbs, such as the following English-Italian one:

⟨X *gives* Y Z, X *dà* Z *a* Y⟩.

2.2 Generating grammar rules

The alignments outputted by the CE module described so far are represented as pairs of UD (sub)trees in CoNLL-U format. While converting

[2] documentation is available at github.com/GrammaticalFramework/gf-ud

them to GF ASTs is one of gf-ud's core functionalities, such trees also need be converted into the grammar rules of a compilable GF translation lexicon. To this end, our grammar generation module requires a *morphological dictionary* of the languages at hand and an *extraction grammar*.

Morphological dictionaries, implemented for several languages as part of the RGL, are large collections of lemmas associated with their inflectional forms.

An extraction grammar, on the other hand, defines the syntactic categories and functions the entries of the automatically generated lexicon are built with. For example, entries can be prepositional phrases (PP) or verb phrases (VP) constructed by the following GF functions:

```
PrepNP : Prep -> NP -> PP # case head
PrepPP : VP   -> PP -> VP # head obl
```

The dependency labels appended to the function types instruct gf-ud to build GF trees from UD trees that match these labels.

The final translation lexicon, i.e. a GF grammar that extends the extraction grammar, is then derived from these GF trees. In its abstract syntax, the name of each concept is associated with its grammatical category, i.e. the category of the root of the GF tree. For example:

```
fun in_the_field__inom_område_PP : PP ;
```

The concrete syntaxes, on the other hand, contain the linearization rules for each concept, directly based on the trees obtained from gf-ud. For instance, in English:

```
lin in_the_field__inom_område_PP =
  PrepNP
    in_Prep
    (DetCN the_Det (UseN field_N)) ;
```

Most function words, such as in_Prep, and many content words, such as field_N, are available through the morphological dictionaries. When this is not the case, they are assumed to be regular and an additional rule is generated for them. For instance, if the English morphological dictionary didn't contain the word "*field*", we would have:

```
oper field_N = mkN "field" ;
```

3 Evaluation

In this section, we evaluate the system proposed above. We first discuss the data used in the evaluation. After that, we describe our experiments, aimed at putting both the CE module *per se* and the entire system from parsing to lexicon generation to the test, and present our results.

3.1 Data

Because we want part of our evaluation to be independent from the quality of UD parsing, some of the experiments are carried out on treebanks instead of raw text. To this end, we use a 100-sentence subset of the Parallel UD (PUD) corpus, a set of 1000 manually annotated or validated sentences in CoNLL-U format. Of the over 20 languages PUD treebanks are available in, we selected English, Italian and Swedish. Using this data limits the amount of alignment errors that are due to annotation issues to a minimum, even though a small number of inconsistencies is present even in this corpus.

When it comes to testing the program on raw text, we use two small (< 1000 sentences) bilingual sentence-aligned corpora consisting of course plans from the Department of Mathematics and Computer Science (DMI) of the University of Perugia (for English-Italian) and from the Department of Computer Science and Engineering (CSE) shared between the University of Gothenburg and the Chalmers University of Technology (for English-Swedish). For brevity, we will refer to these two datasets as to the DMI and the CSE corpora. This data, available in the project repository, was collected and sentence-aligned specifically for this work and a related Bachelor's thesis project (Eriksson et al., 2020). When using raw text, our parser of choice is UDPipe (Straka et al., 2016). In particular, we use the ParTUT English and Italian models for the DMI corpus and models trained on the bilingual LinES English-Swedish treebank for the CSE corpus [3].

3.2 Evaluating CE

While we focus mostly on the MT applications of CA, automatic translation, and much less GF-based domain-specific translation, is not the only context in which CA can be put to use. For instance, it is easy to imagine using it to build translation memories to be use as an aid for human translation. For this reason, a first set of experiments is aimed at evaluating the alignments obtained with our CA module independently from the other stages of our lexicon generation pipeline.

[3] The pretrained UDPipe models and information about their performance are available at ufal.mff.cuni.cz/udpipe/1/models

	CE		`fast_align` (100 sentences)		`fast_align` (full dataset)	
	en-it	en-sv	en-it	en-sv	en-it	en-sv
distinct alignments	536	638	1242	1044	1286	1065
correct	392 (73%)	514 (80%)	346 (28%)	538 (52%)	540 (42%)	677 (64%)
usable in MT	363 (68%)	503 (79%)	316 (25%)	525 (50%)	510 (40%)	666 (63%)

Table 1: Comparison between our grammar-based CE module and `fast_align` on PUD data, training the statistical model both on 100 and 1000 sentences and discarding the alignments obtained for sentences 101-1000 in the latter case.

We first assess the correctness of the alignments the CE module is able to extract from the PUD treebanks, comparing our results with those obtained with a statistical tool, `fast_align` (Dyer et al., 2013). In addition, we try to quantify the impact of automated UD parsing on the performance of the CE module by comparing the above results with those obtained on the DMI and CSE corpora.

While precision and recall are two well-known performance metrics, the lack of a gold standard for CE forces us to, before calculating the ratio between the number of correct alignments and the total number of extracted alignments, manually assess the correctness of each alignment. What is more, since some alignments are only correct in the specific context of the sentence pair in which they occur, we make a further distinction between correct alignments that are relevant for a translation lexicon and alignments that are useful for comparing the sentences but should not be used for MT. As an extreme example of a pair-specific alignment, consider the sentences ⟨*He missed the boat, Ha perso il treno*⟩. In both languages, the idea of missing a chance is expressed with idiomatic expressions very similar to each other. However, the Italian translation mentions a train ("*treno*") in the place of a boat.

3.2.1 Results on manually annotated treebanks

In Table 1, we compare the results obtained with our grammar-based module to those obtained statistically on the PUD treebanks. Of course, `fast_align` does not make use of the information present in the CoNLL-U files except with regards to tokenization. On the other hand, the relatively large size of the PUD treebanks makes it possible also to train the statistical tool on the full dataset instead of just using the chosen 100-sentences subset, allowing for a fairer comparison. In both cases, `fast_align` is run with the recommended parameters and the CE program is configured to only extract one-to-many and many-to-one word alignments, as `fast_align` does not align larger phrases. This explains CE's seemingly low recall. To get a better idea, Table 1 can be compared with Table 2, which summarizes the results of an experiment where no constraints are placed on the size of the extracted alignments.

While `fast_align` is designed to align every word in the text (or explicitly state that a word has no counterpart in its translation) and, consequently, extracts around twice as many correspondences, the percentage of correct correspondences is definitely in favor of our system, even though `fast_align` gets significantly more precise when trained on the full 1000-sentence dataset.

3.2.2 Results on raw text

The course plan corpora are significantly harder to work with, the additional challenges being the inexactness of many translations (which is the direct cause of some of the alignment errors encountered in our evaluation) and the fact that our CE module relies, in this case, on the quality of automatic lemmatization, POS-tagging and parsing.

In Table 2, we compare the results obtained on manually annotated data and the course plans corpora parsed with UDPipe.

What is immediately evident, but not unexpected, is a decrease in precision. The percentage of correct alignments, however, stays significantly higher than that obtained with `fast_align` in the previous experiment, even with the model trained on the full PUD corpus. In fact, even though percentages seem similar for English-Swedish, the CSE corpus is roughly half the size of the full PUD corpus.

The results are less encouraging in terms of recall: the number of alignments extracted from the course plan corpora is similar to that obtained from the PUD treebank sample, despite the difference in size. This is explained in part by the presence, in the course plans corpora, of a large amount of very short sentences, and in part by the fact that parse

	PUD (100 sentences)		course plans	
	en-it	en-sv	DMI (881 sentences)	CSE (539 sentences)
distinct alignments	1197	1325	1823	1950
correct	916 (77%)	1112 (85%)	1205 (66%)	1296 (66%)
usable in MT	880 (74%)	1099 (84%)	1157 (63%)	1248 (64%)

Table 2: Comparison between the grammar-based extraction of alignments of any size from manually annotated PUD treebanks and from automatically parsed sentences from the course plans corpora.

errors introduced by UDPipe make it impossible to align many subsentences without a significant loss in terms of precision.

Our system is, on the other hand, capable of extracting multiword alignments that are unlikely to be identified by a statistical tool, especially in the case of such a small dataset. Examples of this are the noun phrases ⟨*the aim of the course, l'obiettivo del corso, syftet med kursen*⟩ (a concept found in both corpora and, as such, trilingual), ⟨*Natural Language Processing, språkteknologi*⟩ and ⟨*object oriented programming, programmazione ad oggetti*⟩.

3.3 MT experiments

The second set of experiments has the objective of assessing the quality of the final output of the system we propose: GF translation lexica. Because we are now focusing on using CA in the context of domain-specific MT, we do not make use of the PUD treebanks, where sentences come from a variety of different sources, but just of the course plans corpora. We do not construct a grammar specific to such domain: for small-scale MT experiments, it is sufficient to extend the extraction grammar itself with preexisting syntax rules defined in the RGL.

The idea is to automatically translate a set of English sentences to Italian and Swedish, ask native speakers of the target languages to produce a set of reference translations, and compare them to the original machine-generated ones by computing BLEU scores. Due to the small size of the datasets and the consequently low coverage of the extracted lexicon, we generate the sentences to translate directly in the GF shell rather than trying to parse arbitrary sentences from other course plans. In order to do that, we make use of GF's random AST generation functionality but at the same time manually select semantically plausible sentences to facilitate the task of the human translators. The results of this process are two small testing corpora, one for the DMI and one for the CSE corpus, each consisting of 50 English sentences. Reference translations are obtained by asking participants to compare the original English sentences to their automatically translated counterparts and correct the latter with the minimal changes necessary to obtain a set of grammatically and semantically correct translations. This is important as, if the reference translations are obtained independently, BLEU scores can easily become misleading.

3.3.1 Results

Corpus-level BLEU scores for the automatic translations of the 50+50 sentences of the testing corpora are summarized in Table 3. Following conventions, we report the cumulative n-gram scores for values of n from 1 to 4 (BLEU-1 to BLEU-4). However, being a significant portion of the sentences of length 4 or less, we also report BLEU-1 to BLEU-3 scores, BLEU-1 to BLEU-2 scores and scores obtained considering unigrams only.

	DMI (en-it)	CSE (en-sv)
BLEU-1 to 4	55	61
BLEU-1 to 3	63	68
BLEU-1 to 2	70	74
BLEU-1	79	81

Table 3: BLEU scores for automatic translations based on the course plans grammars.

These synthetic figures are useful to give an idea of the general quality of the translations: overall, although with relatively low scores, English-to-Swedish translation works significantly better than English-to-Italian. Looking back at the results reported in Section 3.2.2, the reason for this is not immediately clear, as the difference in precision between the two language pairs is negligible in the course plan corpora.

Looking at sentence-level scores can, however, be more insightful. For both corpora, scores assigned to individual segments range from the minimum possible value of 0 to the perfect score of 100, which indicates a perfect correspondence between the automatic and reference translation. Examples of sentences that were assigned a perfect BLEU-1 to 4 score are "*the library provides useful textbooks*" (translated to Italian as "*la biblioteca fornisce libri utili*") in the DMI corpus and "*this lab is more dif-*

ficult than the exam" (whose Swedish translation is "*den här laborationen är svårare än tentamen*") in the CSE corpus. On the other hand, it is easy for shorter sentences to be assigned the minimum BLEU-1 to 4 score even when they only contain a single grammatical or semantic error.

Furthermore, a problem with using the BLEU score as the only evaluation metric is the fact that it makes no distinction between content and function words, thus not allowing an evaluation focused specifically on the extracted concepts. The small size of the corpus, however, allows for some error analysis. From the participants' observations about the kind of errors encountered when manually editing the automatic translations, summarized in Table 4, we can conclude that while most errors are in fact due to wrong alignments, the main difference between two corpora lies in the number of translations that only contain grammatical errors. This explains the significant difference observed in the cumulative BLEU scores shown in Table 3.

	DMI (en-it)	**CSE (en-sv)**
semantical	23 (46%)	23 (46%)
grammatical	10 (20%)	3 (6%)
both	3 (6%)	4 (8%)

Table 4: Types of errors encountered in the automatically translated sentences.

Among other things, many Italian contractions such as "*del*" ("*di*" + "*il*", in English "*of the*") are systematically rendered as two separate words due to UDPipe tokenization. Grammatical errors in Swedish are less common and less systematic. Only in one case, for instance, gender is incorrect ("*programbibliotek**en***"). These errors are easy to handle when writing a domain-specific grammar or, in cases like the latter, by making small adjustments to the morphological dictionaries.

Some errors regarding the extracted concepts are also interesting to analyze: the alignment ⟨*class, classe*⟩, for instance, causes the sentence *"I will attend the class"* to be (incorrectly) translated as *"io seguirò la classe"* instead of *"io seguirò la lezione"* even though the correspondence is in fact valid in most contexts in which (within the same domain!) *"class"* is not to be intended as a synonym of *"lesson"* but as teaching group.

4 Conclusions

We have presented a syntax-based alignment method with a focus on its applications in domain-specific translation lexicon generation. Compared with the existing statistical tools, our system has the following advantages:

- it performs consistently well even on small parallel corpora
- it is able to simultaneously extract correspondences between individual words, multiword expressions and longer phrases, including discontinuous constituents
- in conjunction with `gf-ud` pattern matching, it can be used to extract specific types of correspondences, such as predication patterns
- it can automatically generate compilable, morphology-aware GF translation lexica
- it can be configured to easily handle systematic, possibly language pair- or corpus-specific translation divergences.

While it requires manual corrections and completions to an extent that varies according to the quality of the data and the strictness of the chosen criteria, using the alignments obtained with our method can reduce the time required for bootstrapping the translation lexicon building process for a domain-specific CNL significantly. In fact, especially if a comprehensive morphological dictionary is available, part of the alignments will be ready to use in a GF-based system without any intervention.

The tangible fruits of this work are a Haskell library and a number of executables offering a user-friendly interface to perform CE, lexicon generation and various kinds of evaluations. The source code, including a preliminary implementation of CP, is available on GitHub[4]. The software has already been used to analyse customer-provided data in two commercial projects at Digital Grammars.

4.1 Current and future work

Our results, while encouraging, suggest that there is room for improvement in many different directions.

An obvious possible development is optimizing the current, initial implementation of Concept Propagation (CP) for its two use cases: propagating alignments to a new language looking for correspondences using a translation of the same text they were extracted from or using a different text in the same domain. An alternative to the former is to make CE, now working on bilingual texts, n-lingual.

[4] `github.com/harisont/concept-alignment`

When large enough amounts of data are available, using our system in conjunction with a statistical tool seems promising. As discussed above, this is already partially supported and it could prove useful to develop CA as an actual hybrid system.

Finally, since the freedom that generally characterizes human translation and the quality of currently available UD parsers make maximizing both alignment precision and recall unrealistic, tools to make it easier to postprocess the automatically generated lexica are under development.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Bonnie J. Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Eriksson, Gabrielsson, Hedgreen, Klingberg, Vestlund, and Ödin. 2020. Grammar-based translation of computer science and engineering terminology.

Prasanth Kolachina and Aarnte Ranta. 2016. From abstract syntax to Universal Dependencies. In *Linguistic Issues in Language Technology, Volume 13, 2016*. CSLI Publications.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Alexandre Rademaker and Francis Tyers, editors. 2019. *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*. Association for Computational Linguistics, Paris, France.

Aarne Ranta. 2011. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford.

Aarne Ranta, Krasimir Angelov, Normunds Gruzitis, and Prasanth Kolachina. 2020. Abstract syntax as interlingua: Scaling up the grammatical framework from controlled languages to robust pipelines. *Computational Linguistics*, 46(2):425–486.

Aarne Ranta and Prasanth Kolachina. 2017. From Universal Dependencies to abstract syntax. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 107–116, Gothenburg, Sweden. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Jörg Tiedemann. 2011. Bitext alignment. *Synthesis Lectures on Human Language Technologies*, 4(2):1–165.