

Syntax-Based Concept Alignment for Domain-Specific Machine Translation

Anonymous ACL submission

Abstract

In grammar-based domain-specific machine translation, constructing high-quality translation lexica is a crucial but time consuming task requiring significant linguistic knowledge. In presence of a corpus of parallel example sentences, statistical word and phrase alignment tools allow to automate part of this process, but are not suitable for small datasets and often fail to identify discontinuous multiword correspondences. Addressing both problems, we propose a data-driven, syntax-based approach to the automation of this task and put it to the test in a simple grammar-based 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 most often more important than coverage, they can provide strong guarantees in terms of grammatical correctness.

Nevertheless, lexical exactness is, in this context, just as important as grammaticality. An important part of the design of the Controlled Natural Language (CNL) the grammar in such a system describes becomes, then, the creation of a translation lexicon. In many cases, this is done for the most part manually, resulting in a time consuming task requiring significant linguistic knowledge. When the grammar is designed based on a corpus of parallel example sentences, it is possible to automate part of this process 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 limited number of example data is available.

In this paper, we propose an alternative approach to the automation of this task. While still being data-driven, our method is 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, thus 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).

This paper is structured as follows. Section 2 starts by giving an overview of our approach to CA, comparing it to related work, to then focus on our algorithm for extracting correspondences. It is concluded with a description of our method for converting the alignments obtained in this way to a GF translation lexicon. After that, Section 3 presents the results of our first evaluation of the system. Finally, Section 4 consists of a discussion of such results and 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 can be directly compared to the numerous existing 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 implements basic CP functionalities, however, in this paper we focus on CE and restrict its application to bilingual corpora.

As stated in the Introduction, most existing extraction 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 various formalisms used to define the grammars for different languages and from the insufficient degree of robustness of existing parses.

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: Universal Dependencies (UD) (Rade-maker and Tyers, 2019) and Grammatical Framework (GF) (Ranta et al., 2020).

While both formalisms, the former a dependency and the latter a constituency grammar, independently solve the former issue, UD is especially appealing since dependency trees are an easier target for parsing. As a consequence, several robust dependency parsers, such as (Straka et al., 2016) and (Chen and Manning, 2014) are available.

UD parsing alone would then be 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 proposed system is able to convert them to GF grammar rules, easy to embed in a domain-specific grammar but also making it possible to immediately evaluate the system by carrying out small-scale translation experiments using pre-existing grammatical constructions implemented in GF’s Resource Grammar Library (RGL). This is made possible by gf-ud, a conversion tool described in (Kolachina and Ranta, 2016) and (Ranta and Kolachina, 2017).

Concretely, the system we propose requires the following elements, whose reciprocal relations are shown in Figure 1:

- a UD parser

- an alignment module based on dependency tree comparison
- a program, based on gf-ud, that converts the alignments into GF grammar rules and uses them to construct a translation lexicon.

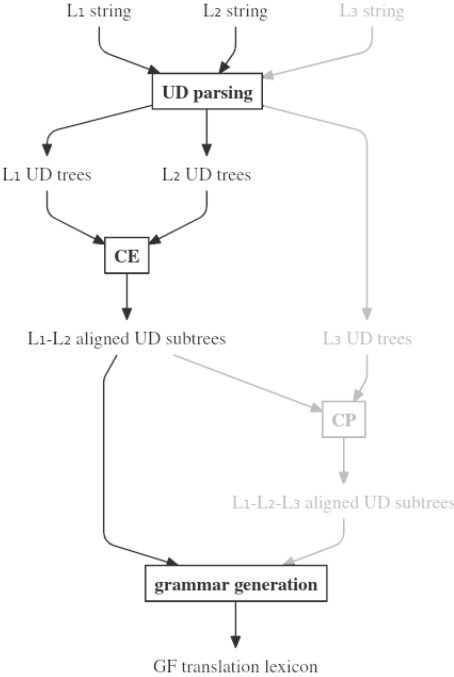


Figure 1: System overview. Parts in grey are currently at the prototype stage and will not be further discussed in this paper.

2.1 Extracting concepts

The core part of the system outlined above is of course the alignment module. In this section, we present our method for extracting alignments from parallel bilingual UD treebanks, which can be obtained from sentence-aligned texts with any UD parser.

The pseudocode of the basic extraction algorithm is shown in Figure ?? . The input is a list of priority-sorted *alignment criteria*, i.e. rules to determine whether two dependency trees should be aligned with each other, which will be discussed in greater detail in Section 2.1.1, and UD tree representation of the pair of sentences to align. From an implementation point of view, UD trees are, as in gf-ud, rose trees where each node represents a token, obtained from the CoNLL-U files produced by the UD parser.

As a first step, the program checks whether the two full sentence trees can be aligned with each other, i.e. if they match any of the alignment criteria. If this is the case, they are added to a collection of alignments, which are also represented as a pair of UD trees associated with some metadata. Such collection is what the function will return after aligning all the dependency subtrees.

Depending on which alignment criteria they match, the head of the two trees may or may not also be added to such collection. This is important...

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 a lack of matching criteria.

When working on corpora consisting of multiple sentences, this algorithm can be applied in an iterative fashion, so that knowledge gathered when a sentence pair is aligned can be reused when working on the following ones and to keep track of the number of occurrences of each alignment throughout the entire text.

2.1.1 Alignment criteria

While alignment criteria are customizable, to better understand the extraction algorithm described in the above we dedicate this section to describing the ones that are currently used 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 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. Intuitively, this means that they are in the same relation with their heads.

Note that, since the root of UD trees is always attached to a fake node with an arc labelled `root`, this criterion also makes it so that full sentences are always considered to align with each other, something which is desirable since we assume the parallel texts that are fed to our program to be sentence-aligned.

Part-Of-Speech equivalence CoNNL-U files provide information not only on the syntactic role of the tokens a sentence is composed of, but also on their grammatical categories, represented as Universal Part-Of-Speech (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 (in the current implementation, defined as in the UD documentation¹), thus ignoring prepositions, determiners, auxiliary verbs and all other function words, which tend to behave differently across different languages.

As a consequence, a useful relation to define between dependency trees is that of *POS-equivalence*: two dependency trees t, u are POS-equivalent if $M_1 = M_2 \neq \emptyset$, where M_i is defined as the multi-set 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 apply them simultaneously: combining these first two criteria, for instance, is useful in cases where precision is more important than recall.

Known translation divergence Parallel texts often present significant, systematic cross-linguistic grammatical distinctions. When this is the case, it is often straightforward to define alignment criteria based on recognizing the corresponding patterns. If 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 have nothing to do with idiomatic usage or aspectual, discourse, domain or word knowledge. Drawing inspiration from (Dorr, 1994), we refer to them as *translation divergences* and handle the most common ones via alignment criteria that are used by default.

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 seem a criterion with no practical applications. However, it is particularly

¹universaldependencies.org/u/pos/all.html

add a 243
examples
(≤ one 244
per type 245
of diver- 246
gence) 247

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 in combination with a statistical tool and give more credit to the correspondences identified by both systems.

2.1.2 Pattern matching

So far, we 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 so to prioritize precision or coverage, but never restricted our search to a particular type of alignments. Nonetheless, there can be cases in which only certain syntactic structures are of interest: for instance, we might be looking for adverbs or noun phrases exclusively.

To handle these situations, the CE module can filter the results based on a `gf-ud` pattern. Pattern matching in `gf-ud`...

Combining pattern matching with pattern replacement, for instance by pruning the UD trees extracted by the alignment module, allows us even to extract correspondences that, given the structure of UD trees, cannot be identified by CE alone. To give an example that turned out to be useful in practice, a possibility is to extract verb phrases only, by looking for full clauses and pruning the subtree corresponding to the subject. More advanced replacements are also possible: instead of simply pruning clauses to extract the verb phrases they contain, we can perform some actual replacement to obtain *predication patterns*, such as

$\langle X \text{ tells } Y Z, X \text{ dice } Z \text{ a } Y \rangle$

2.2 Generating grammar rules

The CE module described so far outputs pairs of UD trees. While converting them to GF Abstract Syntax Trees (ASTs) is `gf-ud`'s core functionality, to build a compilable GF translation lexicon it is necessary to turn such trees into grammar rules.

In order to do that, our grammar generation module requires a morphological dictionary of the languages at hand and an *extraction grammar*. What we refer to as an extraction grammar is of course a

GF grammar, whose objective is to define the set of basic categories and syntactic rules the entries of the automatically generated lexicon is going to use.

3 Evaluation

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 using manually annotated data. To this purpose, we use a 100-sentences subset of the Parallel UD (PUD) corpus, a set of 1000 handcrafted² multilingual treebanks in CoNLL-U format created for the CoNLL 2018 shared task on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2018). PUD treebanks are available in over 20 languages, of which we selected English, Italian and Swedish. Using this data limits the amount of alignment errors that are due annotation issues to a minimum, even though a small number of inconsistencies and imprecisions is present even in this corpus.

When it comes to testing the program on raw text, we use two bilingual sentence-aligned corpora consisting of course plans from the Department of Mathematics and Computer Science 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, or collectively as to the course plans datasets. 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.

3.2 Evaluating CE

While in this work 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 our system can be put to use. For instance, it is easy to imagine using it

²Annotation is sometimes semi-automatic, but all treebanks are at least manually checked.

not sure how important this section is and what (if anything) to add to it

not even I fear. Check this

to improve the UI of online translation memories or to facilitate reading parallel texts in the context of language learning. For this reason, a first set of experiments is aimed at evaluating the alignments obtained with our CA module independently from the subsequent stages of our proposed lexicon generation pipeline.

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

For measuring the performance of our system, while precision and recall are two well-known metrics that would be very well suited to the task, the lack of a gold standard for CE imposes us to approximate them with respectively:

- the number of correct alignments the program is able to extract
- the ratio between such number and the total number of alignments extracted.

However, 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, marked as + in the following tables, and alignments that are useful for comparing the sentences but should not be used for MT, marked as = instead. Incorrect correspondences are marked as -.

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 excepts with regards to tokenization. On the other hand, the relatively large size of the PUD treebanks makes it possible to also 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.

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 as our CE module, 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, both for our CE module and `fast_align`. In the former case, the additional challenge is relying on the quality of automatic parsing, while `fast_align` suffers from the small size of the datasets, which are fed in their entirety to both systems.

The results of this comparison are summarized in Table 2.

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 focussing on using CA in the context of domain-specific MT, in this case we do not make use of the PUD treebanks, where sentences come from a variety of different sources, but rather of the two course plans corpora. For this evaluation, we do not construct a grammar specific to the course plans domain: it is sufficient to extend the extraction grammar itself with preexisting syntax rules defined in the RGL, so to allow for more variation.

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, as a consequence, the low coverage of the extracted lexicon, we generate the sentences to translate directly in the GF shell, making use of its random AST generation functionality but at the same time filtering out semantically implausible sentences to facilitate the task of the human translators. The results of this process are two small testing corpora, each consisting of 50 sentences in English, generated using the abstract syntax and the English concrete syntax of the DMI and the CSE grammar respectively.

Reference translations are obtained by asking

position properly; run experiments this is new and actual results
add results discussion

	CE		fast_align (100)		fast_align (1000)	
	en-it	en-sv	en-it	en-sv	en-it	en-sv
tot. alignments	536	638	1242	1044	1286	1065
+ and =	392 (73%)	514 (80%)	346 (28%)	538 (52%)	540 (42%)	677 (64%)
+	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. The middle column shows the results obtained by training fast_align on the 100-sentence subset fed to the extraction module. On the right, the results obtained by training fast_align on the full PUD treebanks dataset, discarding the alignments obtained for sentences 101-1000 for the final comparison. Row 1 reports the total number of distinct alignments obtain. Rows 2 and 3 give the percentages of correct alignments, with and without considering sentence pair-specific correspondences as correct.

	PUD (100 sentences)		course plans	
	en-it	en-sv	DMI (en-it) (881 sentences)	CSE (en-sv) (539 sentences)
tot. alignments	1197	1325	1823	1950
+ and =	916 (77%)	1112 (85%)	1205 (66%)	1296 (66%)
+	880 (74%)	1099 (84%)	1157 (63%)	1248 (64%)

Table 2: TODO:

two native speakers of Italian and Swedish to compare the original English sentences to their automatically translated counterparts and correct the latter. Participants were instructed to only make the minimal changes necessary to obtain, starting from the automatic translations generated by our system, a set of grammatically and semantically correct translations. This is important to get meaningful results, as if the reference translations are obtained independently from the automatic ones, BLEU scores can easily become misleading.

3.3.1 Experimental results

Corpus-level BLEU scores for the automatic translations of the 50+50 sentences of the testing corpora generated as described in the above 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.

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

	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 obtained by comparing one candidate automatic translation per sentence to a reference translation obtained by making the necessary corrections to the automatically generated ones. Both the initial English sentences and their automatic translations were generated using the two course plans grammars.

between the two language pairs is negligible on the course plans 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 the 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 difficult 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. This is the case, for instance, of the sentence “*the test is oral*”, whose last word, “*oral*” is translated as “*dura*” (“*hard*”) instead of “*orale*” due to an alignment error. Nonetheless, it is worth noticing that this is one of the many cases in which the correct alignment (“*oral*”, “*orale*”) is also identified by the CE module, but hidden by the fact that, for simplicity, we only take into account the first translation candidate. This suggests that, in contexts where higher precision is necessary, there are simple improvements that can be done either in the choice of the alignment criteria or at a later stage, in terms alignment selection. In other contexts, it can be more practical to aim for higher coverage and not perform any automatic selection, leaving lexicon refinements to a human, possibly with the help of the postprocessing tools that are currently under development.

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. While postprocessing the automatic translations, the participants were asked to indicate what kind of errors they encountered in each sentence (grammatical, semantical or both). Their observations are summarized in Table 4.

type of errors	DMI (en-it)	CSE (en-sv)
semantical	23 (46%)	23 (46%)
grammatical	10 (20%)	3 (6%)
semantical and grammatical	3 (6%)	4 (8%)

Table 4: Number of automatically translated sentences containing only semantical, only grammatical and both semantical and grammatical errors in the two synthetic course plans corpora.

Interestingly, while most errors are due to wrong alignments, the main difference between two corpora lies in the number of translations that only contain grammatical errors, which explains the significant difference observed in the cumulative BLEU scores shown in Table 3.

In Italian, grammar errors often involve contractions such as “*del*” (“*di*” + “*il*”, in English “*of the*”), some of which are systematically rendered as two separate words due to UDPipe tokenization. Another common case is that of wrong (or at least very confusing) adjective collocation,

such as in the translation “*il libro presenta una tecnica con miglioramenti vari utile*” (“*the textbook presents a useful technique with various improvements*”), where the adjective “*utile*” (“*useful*”), referred to “*technique*” (“*tecnica*”) is placed far from such noun, making the sentence hard to interpret³. Grammatical errors in Swedish are less common and, apart from long adjectives never being turned to the comparative degree periphrastically (e.g. “*relevantare*” instead of “*mer relevant*”), less systematic. Only in one case, for instance, gender is incorrect (“*programbiblioteken*”). All of these grammatical errors are easy to handle when writing a domain-specific grammar or, for instance in the case of misgendered nouns, by making small adjustments to the morphological dictionaries.

Some errors regarding the extracted concepts are also interesting to analyze. In English-Swedish, while several compounds, such as (“*computer science*”, “*datavetenskap*”), are aligned correctly, there are cases in which a single-root noun is translated as a compound (e.g. (“*theory*”, “*automatate-ori*”). This is not necessarily an actual alignment error, as it might well be the case that the English version of the text was being less specific than its Swedish counterpart, thus producing an alignment that, during evaluation, would have been marked as “correct but not relevant for a translation lexicon” (=). “Reasonable” alignment errors appear in the DMI corpus sentences too. The English phrase “*attendance of lessons*”, for instance, becomes simply “*frequenza*” in the Italian translation. Such word is in fact often used alone to replace longer expressions such as “*frequenza delle lezioni*”, just like in English “*of lessons*” can be omitted when what must be attended is evident from the context. Another interesting example is the alignment (“*class*”, “*classe*”), which 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 of the numerous contexts in which (within the same domain!) “*class*” is not to be intended as a synonym of “*lesson*”.

4 Conclusions

We presented a syntax-based alignment method with a focus on its applications in domain-specific translation lexica generation.

³The manually postprocessed translation is “*il libro presenta una utile tecnica con miglioramenti vari*”.

Compared to the existing statistical tools, our system has the following advantages:

- it performs consistently well even on extremely small parallel corpora
- it is able to simultaneously extract correspondences between individual words, multiword expressions and full phrases, even when their constituents are discontinuous
- 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 language-pair specific translation divergences.

The tangible fruit of this work are a Haskell library and a number of executables offering an easy to use and configure interface to perform CE, lexicon generation and a variety of kinds of evaluations. The source code, which also implements a preliminary implementation of CP not discussed in depth in this paper, is available at github.com/harison/concept-alignment.

4.1 Current and future work

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

An obvious possible development is to improve the current implementation of CP, optimizing it for its two different use cases: propagating alignments to a new language looking for correspondences in a different text in the same domain and using a translation of the same text they were obtained from. An alternative to the latter is to generalize CE to n languages, instead of limiting it to bilingual cases.

For cases where a relatively large amount of data is available, using our system in conjunction with a statistical tool seems promising. As discussed in Section 2.1.1, some integration between our system and other sources of alignments is already supported, and it is possible to keep track of the reason why a particular correspondence was extracted. This could be further developed in the direction of an actual hybrid system. A first attempt in this direction is the development of a simple conversion tool between the “pharaoh format” used

by tools such as `fast_align` and the format we use to represent alignments, available in the same repository.

Finally, since the freedom that generally characterizes human translation and the quality of currently available UD parsers make the expectation of maximizing both alignment precision and recall unrealistic, it seems logical to develop tools to make it easier to postprocess automatically generated lexica. With respects to this, current work is represented by a synoptic CoNLL-U viewer, available at github.com/DigitalGrammarsAB/synoptic-conllu-viewer, which makes it possible to inspect the UD trees a grammar rule was derived from.

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, Gabriellsson, 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, 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. UD-Pipe: 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.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 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*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.