



Recent advances in Apertium, a free/open-source rule-based machine translation platform for low-resource languages

Tanmai Khanna¹ · Jonathan N. Washington² · Francis M. Tyers^{3,8} ·
Sevilay Bayatlı⁴ · Daniel G. Swanson² · Tommi A. Pirinen⁵ · Irene Tang⁶ ·
Hèctor Alòs i Font⁷

Received: 17 March 2020 / Accepted: 1 March 2021 / Published online: 18 October 2021
© The Author(s) 2021

Abstract

This paper presents an overview of Apertium, a free and open-source rule-based machine translation platform. Translation in Apertium happens through a pipeline of modular tools, and the platform continues to be improved as more language pairs are added. Several advances have been implemented since the last publication, including some new optional modules: a module that allows rules to process recursive structures at the structural transfer stage, a module that deals with contiguous and discontiguous multi-word expressions, and a module that resolves anaphora to aid translation. Also highlighted is the hybridisation of Apertium through statistical modules that augment the pipeline, and statistical methods that augment existing modules. This includes morphological disambiguation, weighted structural transfer, and lexical selection modules that learn from limited data. The paper also discusses how a platform like Apertium can be a critical part of access to language technology for so-called low-resource languages, which might be ignored or deemed unapproachable by popular corpus-based translation technologies. Finally, the paper presents some of the released and unreleased language pairs, concluding with a brief look at some supplementary Apertium tools that prove valuable to users as well as language developers. All Apertium-related code, including language data, is free/open-source and available at <https://github.com/apertium>.

Keywords Machine translation · Low-resource languages · Rule-based machine translation · Hybrid machine translation

Several of the advances described in this paper were supported by Google Summer of Code funding, for which the authors are very grateful. The authors also recognise the Hungerford Faculty Support Fund at Swarthmore College and the University of Chicago's Department of Linguistics Graduate Research Aid Initiative in Linguistics (GRAIL) for their generous support in covering open access publication costs.

✉ Jonathan N. Washington
jonathan.washington@swarthmore.edu

Extended author information available on the last page of the article

1 Introduction

Apertium (Forcada et al. 2011) is a free/open-source platform for rule-based machine translation (RBMT). It was designed to use the shallow transfer based approach to translation, and most modules in the pipeline work on rules written by language developers and linguists. The platform provides an accessible way to create language data and rules, such that apart from experienced language developers, speakers of a language with a limited understanding of programming and/or linguistics can create decent translation systems for their languages as well. This is a superior model for creating translation systems for low-resource languages both because it involves stakeholders from the language communities, and because most languages lack widely available corpora that would be needed for fully data-driven approaches. Apart from developing RBMT systems for low-resource languages, the Apertium open source organisation also develops and supports tools for the creation of RBMT systems.

Several advances to the Apertium platform (Release version 3.6) have been implemented since the previous publication (Forcada et al. 2011). These include organisational improvements, additional tools, additional methods to augment RBMT with corpus-based methods, new modules for more precise translation, a few additional tools not directly involved in the RBMT pipeline, and resources for many more languages and translation pairs.

Organisational changes include a migration of the codebase from subversion (hosted by SourceForge) to git (hosted by GitHub), a switch from two-letter ISO codes (ISO 639-1) to three-letter ISO codes (ISO 639-3), and a three-directory model for translation pairs (one for components specific to each language, and one for the common components). Additionally, morphological transducers for a number of languages make use of Helsinki Finite-State Technology (HFST) (Lindén et al. 2011), morphological disambiguation has been improved in many languages by using Visual Interactive Syntax Learning Constraint Grammar (VISL CG-3) (Bick and Didriksen 2015), and several new features have been incorporated into the lexical selection module.

Section 2 overviews the design of the Apertium RBMT platform. Section 3 discusses modules used by Apertium to augment RBMT using corpus-based methods. Section 4 introduces the new modules in the pipeline: a module that allows rules to process recursive structures at the structural transfer stage, a module that deals with contiguous and discontinuous multiword expressions, and one that resolves anaphors to aid translation. Section 5 discusses Apertium's contribution to language revitalisation and reclamation efforts. Section 6 introduces several supplementary Apertium tools. Section 7 concludes.

2 Overview of the Apertium platform

The overall design of Apertium is a pipeline with a series of modules. Each stage of the pipeline reads from and writes to text streams in a consistent format so that modules can easily be added or removed according to the needs of the languages in question.

Apertium consists of both the management of the pipeline (the main `apertium` executable) and all the stages in this pipeline, except where outside tools (such as HFST for morphological analysis and generation, or CG for morphological disambiguation) are used. Each stage consists of a general processor which modifies the stream based on hand-crafted “rules” (coded linguistic generalisations) for a given language or language pair. Figure 1 shows the entire pipeline, including optional modules.

A short overview of each of the stages of the pipeline is provided below. The new ones are discussed in further detail in Sect. 4.

- **Deformatter:** Encapsulates any document formatting tags so that they go through the rest of the translation pipeline untouched. This is a language-agnostic part of Apertium. Deformatter modules are available for OpenDocument Text, Microsoft Word, HTML, and other popular text formats.
- **Source Language morphological analyser:** Segments the surface form of text (into words or multi-word lexical units) using a finite-state transducer (FST) and delivers one or more lexical forms (or “analyses”), each of which includes a lemma and a part-of-speech label (encoded as a “tag”), as well as any relevant subcategory and grammatical (e.g., inflectional) information (also encoded as tags).
- **Source Language morphological disambiguator:** Tries to choose the best sequence of morphological analyses for an ambiguous sentence. The original Apertium disambiguator used a first-order hidden Markov model (HMM). Other statistical models, such as averaged weighted Perceptron, have since been added and are currently in use for various languages. Additionally, CG (Bick and Didriksen 2015) is often combined with a statistical model for a two-step process. The different approaches are discussed in Sect. 3.1.
- **Source Language retokenization:** Adjusts token boundaries for multi-word expressions, which can be non-contiguous (such as separable verbs in Germanic languages), in preparation for translation. Often this consists of combining component parts into single multi-word expressions. This module is discussed in more detail in Sect. 4.2.
- **Lexical transfer:** Reads each source-language (SL) lexical form and delivers a set of corresponding target-language (TL) lexical forms by looking it up in a bilingual dictionary, implemented as an FST.
- **Lexical selection:** Hand-crafted or learned language-specific context-sensitive rules, also implemented using FST technology, choose the most adequate translation of SL lexical forms with multiple TL translations. The original module (Tyers et al. 2012a) has been extended with new features like macros. The hybridisation of this module is discussed in more detail in Sect. 3.2.

- **Source Language anaphora resolution:** Attempts to resolve references to earlier items in discourse using language-specific saliency metrics. This module attaches the lexical unit of the antecedent to its corresponding anaphor to aid translation. This module is discussed in more detail in Sect. 4.3.
- **Shallow structural transfer:** Apertium's shallow structural transfer module applies a sequence of one or more finite-state constraint rules to the output of the lexical selection module. It generally consists of three sub-modules: a chunker mode, an interchunk mode, and a postchunk mode. Apertium 1.0 provided a single structural transfer step. This was considered enough for the translators between the closely related Iberian Romance languages which constituted the first Apertium translators. The one-step strategy is still used in the current released versions of many of them, including the Catalan-Spanish translation pair, which since then has been continuously improved and is widely used.¹ Beginning with the implementation of the Spanish-English and Catalan-English language pairs, a three-step transfer architecture was developed, leading to the release of Apertium 2.0. The first step creates chunks of lexical units in the source language and reorders words inside the chunk as per the transfer rules. The second step reorders chunks based on the target language syntax, and the final step makes the stream ready for the generator. This is currently the standard Apertium structural transfer architecture. Several pairs have additional transfer steps, such as Catalan-Esperanto (5 steps) and French-Occitan (4 steps). In the Catalan-Esperanto translator there are three "interchunk" steps aimed at a deeper syntactic analysis, with the overarching objective of generating the correct case morphology on various types of nominals in the target language (Esperanto), since the source language (Catalan) lacks case morphology except in its pronominal system. The shallow transfer system is used creatively in other ways as well.
- **Recursive structural transfer:** This module is a recently developed alternative to the shallow structural transfer module (chunker, interchunk, and postchunk). Its linguistic data is specified as context-free grammars (CFGs) and it uses a Generalized Left-right Right-reduce (GLR) parser rather than finite-state chunking to more effectively implement long-distance reordering. This module is discussed further in Sect. 4.1.
- **Target Language retokenization:** Adjusts token boundaries for multi-word expressions, which can be non-contiguous (such as separable verbs in Germanic languages), in preparation for target-language morphological generation. Often this consists of separating multi-word expressions into their component parts. This module is discussed in more detail in Sect. 4.2.
- **Target Language morphological generator:** Delivers the sequence of TL surface forms for each corresponding TL lexical form received from earlier modules in the pipeline.

¹ In 2020, the Softcatalà-hosted Apertium translators served an average of 4.6 million requests per month from Spanish to Catalan and 1.1 million from Catalan to Spanish (data kindly provided by Xavier Ivars).

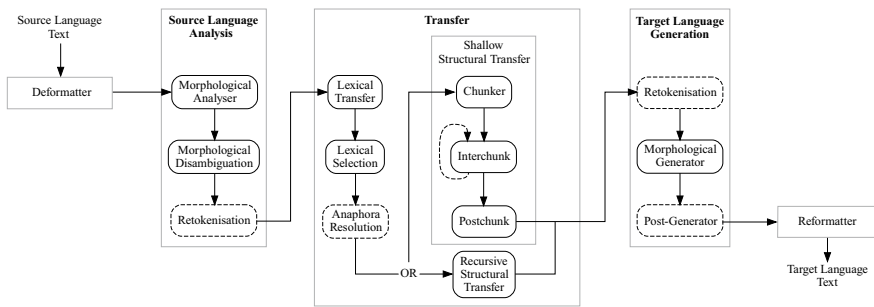


Fig. 1 The architecture of Apertium, a transfer-based machine translation system. Each rounded box is a module available for language-specific or pair-specific development. Broken lines show optional modules. Lines with arrows represent the flow of data through the pipeline. The stages in the pipeline are grouped by whether they are relevant to source-language analysis, bilingual transfer, or target-language generation—the three logical sections of the pipeline. The deformatter and reformatter are language-agnostic and provided by Apertium

- **Target Language Post-generator:** Performs mainly orthographic operations across tokens, for example elision (such as *lo + òme = l'òme* in Occitan), fusion (such as *da + il = dal* in Italian), epenthesis (such as *a > an* in English, or *c > co* and *o > ob* in Russian), or dissimilation (such as *la + agua > el agua* in Spanish).
- **Reformatter:** De-encapsulates any formatting information to prepare a finally formatted document in the target language. This is a language-agnostic part of Apertium. Reformatter modules are available for OpenDocument Text, Microsoft Word, HTML, and other popular text formats.

The reader is referred to the Apertium wiki² for more information about file naming conventions, mode naming conventions, and dates of introduction for each stage of the pipeline. Any further additions to the pipeline will be documented on this wiki.

It should be highlighted that a major difference in the organisation of Apertium language pairs as compared to the original model is the three-directory structure currently used for most (but not all) of the released translation pairs. Initially, every translation pair was developed in a single self-contained repository that included all relevant linguistic data. Currently, monolingual data, such as morphological dictionaries, morphological disambiguators and post-generators, are shared by different translators, allowing much easier reuse of data and cooperation in the improvement of linguistic resources (Marting and Unhammer 2014). Thus, for instance, compiling the *apertium-spa-cat* pair now depends on the *apertium-spa* and *apertium-cat* modules, which are also used by other translation pairs.

² <http://wiki.apertium.org/wiki/Pipeline>.

3 Use of corpus-based approaches in Apertium modules

Several methods of incorporating corpus-based approaches into Apertium RBMT systems are currently available. These methods fall into the domains of morphological disambiguation (Sect. 3.1), lexical selection (Sect. 3.2), and structural transfer (Sect. 3.3).

3.1 Morphological disambiguation

The goal of morphological disambiguation is to choose the correct morphological analysis if there are multiple possible analyses for a given lexical unit.

The oldest and most commonly used morphological disambiguation method in Apertium (Sánchez-Martínez et al. 2006, 2007) is a module that relies on patterns learned from a corpus. Two statistical methods are implemented, a bigram-based first-order Hidden Markov Model (HMM) (Cutting et al. 1992) and an Averaged Perceptron tagger (Zhang and Clark 2011). The HMM method for morphological disambiguation chooses one analysis from among those returned by the morphological analyser based on a probabilistic model of sequences of part-of speech tags given the surrounding context. The perceptron-based tagger learns a set of weights based on features defined by the language-pair developer.

Furthermore, VISL CG-3 (Bick and Didriksen 2015) has become a popular method among Apertium developers of implementing morphological disambiguation. Using this method, language-specific constraint grammar heuristics (Karlsson et al. 1995) are hand-crafted. This rule-based approach may be combined with a corpus-based approach by chaining them—i.e., by first running the output of the morphological analyser through the constraint-grammar module, and then resolving any remaining ambiguity using a trained model.

3.2 Lexical selection

The goal of lexical selection is to choose an adequate translation in the target language from among several possible translations for a given source-language lexical unit. An FST-based module that allows the writing of rules has been in use for some time (Tyers et al. 2012a).

Apart from manually written rules, a system has also been developed that learns rules through a maximum-entropy model trained in an unsupervised manner (Tyers et al. 2014). The training method requires only a source language corpus, a statistical target-language language model, and the RBMT system itself. All possible translations are scored against the TL language model, and these scores are normalized to provide fractional counts to train source-language maximum-entropy lexical selection models. The output is weighted lexical selection rules that can be processed by the existing lexical selection module.

3.3 Structural transfer module

Structural transfer handles differences between the source and target languages in terms of word order and morphological information by applying transfer rules. In the chunker

module, these transfer rules function by matching a source language pattern of lexical items, grouping them into chunks, and applying a sequence of actions to convert the word order and morphological properties of the chunk as per the target language.

Work has been done to extract, or “learn”, chunking rules using Alignment Templates (Sánchez-Martínez and Ney 2006; Sánchez-Martínez and Forcada 2007; Martínez 2008; Sánchez-Martínez and Forcada 2009; Sánchez-Cartagena et al. 2015). A parallel corpus is searched for sequences of lexical units that exhibit differences in order or morphological information.

There can, however, be more than one potential sequence of actions for each source language pattern, as well as overlapping patterns. The default approach to resolving these ambiguities is that transfer rules are applied to the input using a left-right-longest match algorithm. To improve translation accuracy, though, chunker rules can now be weighted so as to apply different rules in different overlapping lexical environments. These weights can be learned using an unsupervised maximum entropy approach (Bayatli et al. 2020).

The basic goal of this method is to choose between conflicting structural transfer rules based on the lexical environment. For example, the Spanish sentence *Encontré el pastel muy bueno* has (at least) two different (hypothetical) translations to English depending on the syntactic parse in Spanish: (a) “I found the cake very good” or (b) “I found the very good cake”. That is, *muy bueno* may be parsed (a) as a complement to the verb *encontré* or (b) as a modifier to the noun *el pastel*. These parses correspond to different sets of transfer rules, each of which could be matched: (a) a single verb phrase consisting of V DET N ADV ADJ, or (b) a verb V, followed by a noun phrase DET N ADV ADJ. The noun phrase rule would specify that the elements be output in a different order, DET ADV ADJ N, and both rules that match the verb would add a lexical unit for the TL pronominal subject.

A model is produced by running SL text through all possible transfer rules, comparing the potential translations that are output to a TL language model, and dividing the scores by the series of SL lemmas that matched each transfer rule pattern for a given potential translation. If the example above were part of the training data, then the potential translation *I found the cake very good* would score higher than *I found the very good cake* against an English language model due to having a higher probability. These different scores are then distributed as weights, along with the input lemmas, attached to the rules that each translation is the result of. In this example, the weight assigned to the V DET N ADV ADJ rule for the Spanish lemmas is higher than the sum of the weights assigned to the V and DET N ADV ADJ rules for these same lemmas, and hence the V DET N ADV ADJ rule will be selected.

During translation, when a string of SL text matches multiple transfer rules, the system is able to choose between them (infer the “correct” one) based on the weights associated with the rules that the SL lemmas trigger. For example, if this same sentence were being translated, the V DET N ADV ADJ rule would be matched, resulting in the output “I found the cake very good”.

A contrasting example, *Encontré un pastel muy bueno*, would match the same two sets of rules, but would result in translation occurring through the other rule set. This is because the lemmas of the potential translation *I found a very good cake* would result in higher combined weights for the V and DET N ADV ADJ rules

than the V DET N ADV ADJ rule. This reason for this is that translations containing these lemmas would have scored higher against an English language model than translations like *I found a cake very good*, resulting in higher weights for this set of Spanish lemmas attached to this set of rules.

In both examples of Spanish inputs, using this approach and a suitable corpus to train an English language model, the set of transfer rules that results in the more likely English translation is chosen.

This method has been tested using the Kazakh–Turkish, Kyrgyz–Turkish, and Spanish–English translation pairs, and it has been observed that the results are better when there is a greater number of ambiguous rules. The module has not yet been included in any released translation system.

4 New modules

Several previously unpublished modules are now available for the Apertium pipeline. Discussed in this section are `apertium-recursive`, which provides for true recursive transfer (Sect. 4.1); `apertium-separable`, which enables the processing of multi-word expressions (Sect. 4.2); and `apertium-anaphora`, which allows the resolution of anaphors in the source text (Sect. 4.3).

4.1 Recursive structural transfer

Given the range of possible syntactic structures, it is common for any two languages to have significantly different word orders. For example, in Welsh, verbs tend to be at the beginning of a sentence; in English they tend to be in the middle; and in Kyrgyz, they tend to be at the end.

These differences are problematic for Apertium’s finite-state chunking module, which matches fixed sequences of words that must be contiguous. This limitation means it is fairly easy to write rules which perform operations such as changing the order of nouns and adjectives, since these are usually adjacent, but changing larger structures is much harder. Switching the order of the subject and the main verb, for instance, would generally require writing a rule for each sequence of words that can make up each of those parts. The English–Spanish pair has more than 30 chunking rules for handling noun phrases with different numbers of determiners and adjectives, and those rules don’t attempt to deal with all structures that may occur in noun phrases, such as relative clauses.

To deal with the limitations of finite-state chunking, the `apertium-recursive` module (Swanson et al. 2021) was developed by Daniel Swanson as part of Google Summer of Code 2019³ to apply structural transfer rules recursively using context-free grammars (CFGs) and a Generalized Left-right Right-reduce (GLR) parser. This makes it possible to process nested structures such as relative clauses or prepositional

³ <https://summerofcode.withgoogle.com/archive/2019/projects/6746718069063680/>.

phrases within prepositional phrases. An example of the latter is shown in Figs. 3 and 4, with the relevant rules in Fig. 2. In this example, the word order of a set of nested prepositional phrases needs to be completely reversed (or in linguistic terms, the order of noun phrases (NPs) and adpositional phrases (PPs) each needs to be reversed), regardless of the number of prepositional phrases involved in order to translate from English to Basque.

A recursive approach to transfer can be helpful for translation pairs between syntactically more similar languages as well. For example, in the case of the English–Spanish noun phrase rules mentioned above, the more than 30 rules required for handling determiners, adjectives, and nouns can be simplified to less than 10 rules in `apertium-recursive` because more complicated structures can be handled by composing simpler ones. In fact, the majority of these can be covered by just 3 rules saying that a noun phrase is composed of a noun, or an adjective and a noun phrase, or a determiner and a noun phrase. These 3 rules can immediately handle any number of determiners and adjectives.

4.2 Processing multi-word expressions

Multi-word expressions (MWEs) are compound expressions composed of two or more words, such as phrasal verbs (*take out*, *wake up*, *make a call*) and phrasal nouns (*telephone pole*). Separable multi-word expressions are those that may be split by an intermediary word or phrase (such as *take out* in *take the trash out*). This phenomenon can be seen in a number of languages. In English, the multi-word *take away* can remain unified, such as in *take away the item*, or be split up, such as in *take the item away*—both phrasings have identical meanings. This phenomenon can also be seen in some German verbs, where the separable particle can detach from its lexical core, such as with the separable verb *anrufen* ‘to call’: *rufe meine Freundin an* ‘call my friend’. See Constant et al. (2017) for more on this phenomenon.

Separable MWEs are particularly problematic for Apertium’s rule-based translation. Prior to the introduction of the `apertium-separable` module, the individual components of both non-separable and separable multi-words were translated as individual tokens, often leading to less-optimal translations. For example, during the English-to-Spanish translation of *take the trash away*, the phrase’s individual components were translated to produce *tomar la basura fuera* which isn’t a phrase that native speakers of Spanish would produce. The more optimal solution is to process *take away* as a single unit in order to obtain the correct expression *sacar la basura*. Similarly, the Arpitan verbal expression *tornar fâre* ‘to redo’ has negative forms of the type *tornar pas fâre* which were not previously recognised nor correctly generated.

`Apertium-separable` provides a framework to address mistranslations arising from this sort of non-contiguous word ordering. Section 4.2.1 describes the module and Sect. 4.2.2 describes its usage.

```

NP -> det n { 2 + 1 } |
      NP PP { 2 _ 1 } ;

PP -> pr NP { 2 + 1 } ;

```

Fig. 2 A simple set of recursive rules translating a subset of noun phrases and prepositional phrases from English to Basque. A noun phrase (NP) in the source language consists of a determiner (det) and a noun (n), and may optionally include a prepositional phrase (PP), and a prepositional phrase consists of a preposition (pr) and a noun phrase. All three output rules reverse the order of the two nodes: the order of a determiner and a noun is reversed, the order of a noun phrase and a prepositional phrase is reversed, and the order of an adposition (preposition/postposition) and a noun phrase is reversed. The action part of the rules (building up the target translation) appears between braces { . . . }. The indices, 1 and 2, indicate the position of the unit matched in the input, _ represents a space in the output, and + indicates that the words on either side of it will be conjoined without a space

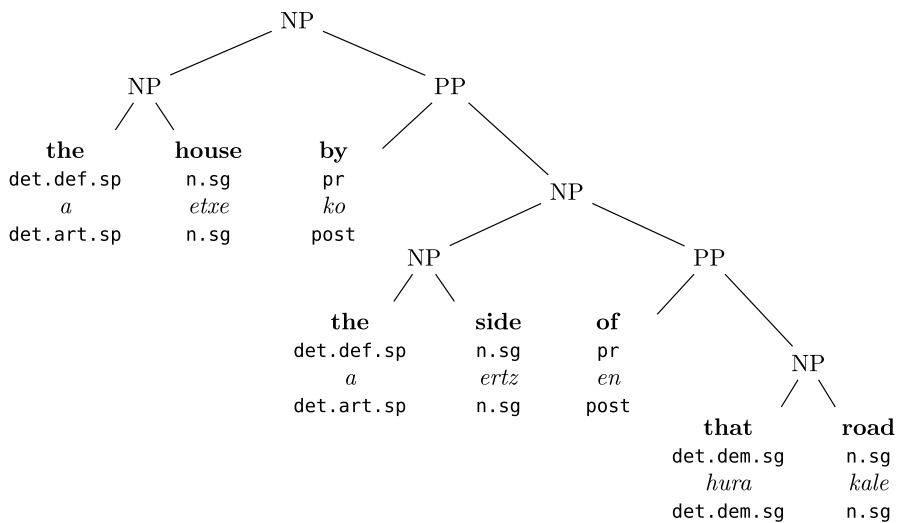


Fig. 3 A source language parse tree for the phrase *the house by the side of that road* built using the rules in Fig. 2. When no further application of the rules is possible, this tree will be transformed into the tree shown in Fig. 4

4.2.1 The apertium-separable module

The *apertium-separable* module was developed by Irene Tang as part of Google Summer of Code 2017⁴ to handle both contiguous and discontinuous (or “separable”) MWEs. The compiler accepts an XML-format dictionary as input, which contains a list of phrase types and a list of mappings between MWEs and their component elements—and in the case of non-contiguous MWEs, a specification of the possible phrase types that might separate the elements of the MWE.

⁴ <https://summerofcode.withgoogle.com/archive/2017/projects/4690909727817728/>.

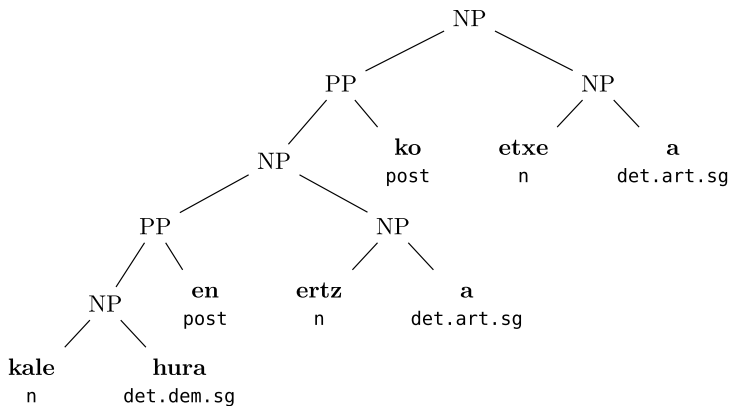


Fig. 4 The target language tree resulting from applying the action steps of the rules in Fig. 2 to the tree in Fig. 3. The analyses yielded by this tree will generate the Basque phrase *kale haren ertzeko etxe a* ‘the house by the side of the road’. The final step of combining definite articles and postpositions with the immediately preceding words is not shown

As an example, one phrase-type entry that the `eng` dictionary might include is the definition of a noun phrase (NP) as (among other patterns) any sequence of words such that the first contains a `<det>` tag, the second an `<adj>` tag, and the last a `<n>` tag. The `eng` dictionary should then also include an entry specifying how *take away* as an MWE followed by such a noun phrase may be mapped to its component elements. These phrase-type and vocabulary entries work together as a framework for handling MWEs.

The XML dictionary is compiled into a finite state transducer. As a parser feeds the input text into the transducer one character or tag at a time, it looks out for sequences of characters and tags that match anything in the dictionary. If a match is found, then the parser outputs the corresponding substitution.

Processors for this module may be included in two places in the Apertium RBMT pipeline: immediately following morphological tagging and preceding lexical transfer, and immediately following structural transfer and preceding morphological generation. The former use allows “assembly” of source-language MWEs for transfer, and the latter “disassembles” transferred target-language MWEs for morphological generation.

4.2.2 Usage

Processing seemingly simple contiguous MWEs in this way allows for more robust bilingual dictionary entries with fairly vanilla morphological transducers. For example, it may not make sense to have an entry for *little brother* in an English morphological transducer that already contains the component words, but it is useful to have an entry like this in a bilingual dictionary with a language like Kyrgyz, which has two words for brother with the difference in meaning associated with relative age to a sibling. In this situation, the *apertium-separable* module processes the analysis of *little brother* as an adjective and a noun (`^little<adj>$ ^brother<n><sg>$`) and retokenizes it as a multi-word noun (`^little brother<n><sg>$`). Note that the assembly of the MWE (as described here) would occur in the English-Kyrgyz translation direction before bilingual dictionary lookup, and the disassembly of the MWE (the reverse) would occur in the Kyrgyz-English translation direction before morphological generation.

The module is used extensively in the French–Catalan pair, particularly to handle the phrasal verbs included in the dictionaries. Thus, for example, it is defined in the pair’s bilingual dictionary that *faire appel* ‘to do appeal’ should be translated as *apel·lar* ‘to appeal’. However, this needs to be handled as a discontinuous MWE, since there are often adverbs between the verb *faire* and the noun *appel*, for example when negated: *ne fait pas appel* ‘does not appeal’. The module is used to reorder the phrase before lexical transfer as *ne fait appel pas* (with *fait appel* as a single lexical unit). Since the adverb now follows the multi-word verb instead of appearing between its components, structural transfer does not need to treat such a sentence any differently than sentences containing single-word verbs. Similar examples are found in the [unreleased] Kazakh–Kyrgyz pair.

4.3 Anaphora resolution

The *apertium-anaphora* module was developed by Tanmai Khanna as part of Google Summer of Code 2019⁵ to handle anaphora resolution in the Apertium pipeline. Anaphora resolution is the process of resolving references to earlier items in the discourse. This is necessary in a Machine Translation pipeline as languages have different ways of using anaphors, and sometimes it is necessary to know the antecedent of an anaphor to translate it correctly.

For example, in Catalan, the masculine singular possessive determiner is *el seu*. Its gender and number are inflectional properties relating to how it agrees with nouns, but its referent may be any gender or number. Hence it could be translated to English as any of *his/her/its/their*, the gender and number of which relate to the referent and not to a modified noun. To pick the correct translation in English, then, it is necessary to know what *el seu* refers to. Without a module in an Apertium

⁵ <https://summerofcode.withgoogle.com/archive/2019/projects/5434868157120512/>.

translation pipeline to do this, a default translation of the anaphor appears in the target language. For instance, in the case of English possessive determiners, the default is currently *his*.

While there are several statistical methods to resolve anaphors using machine learning, Apertium is focused on supporting low-resource language pairs, which usually don't have enough data available for these methods to be viable. Common rule-based approaches, on the other hand, often use parse trees (Lappin and Leass 1994; Baldwin 1997; Trouilleux 2002; Lee et al. 2013; Loáiciga and Wehrli 2015; Zeldes and Zhang 2016). The `apertium-anaphora` module uses a rule-based approach to anaphora resolution which does not require any training data, nor rely on parse trees. Based on Mitkov's algorithm (Mitkov 1999), it gives saliency scores to candidate antecedents in the context (the current and previous three sentences) based on **saliency indicators**, which are syntactic or lexical indicators that are expected to correlate to a higher or lower likelihood that a candidate antecedent is the correct one, using positive and negative scores respectively. For example, indefinite nouns can be given a small negative score and proper nouns can be given a small positive score, as it has been shown empirically that they are less or more likely to be the antecedent of anaphors, respectively (Mitkov 1999). After the scores of all the indicators are applied, the candidate with the highest score, hence considered most salient, is chosen as the antecedent. A complete example of this is presented in Sect. 4.3.2. These saliency indicators are specified as manually written rules. These rules are written for and are applied based on source-language forms only. Because of this, a ruleset can be reused for multiple translation pairs with the same source language.

Apart from manually written rules, a universal indicator is the Referential Distance indicator. This indicator, which was also discovered empirically (Mitkov 1999), tells the algorithm that as the distance between the anaphor and candidate antecedent increases, the candidate is less likely to be the correct antecedent of the anaphor. Penalisation of candidates that are further from the anaphor is implemented by adding to candidates in the same sentence as the anaphor a +1 score, candidates in the preceding sentence a +0 score, in the sentence before the preceding sentence a -1 score, and so on.

In the next few sections, some unique features of this module are discussed (Sect. 4.3.1), an example highlighting the process and benefit of having anaphora resolution in the Machine Translation pipeline is shown (Sect. 4.3.2), a preliminary evaluation of the module is presented (Sect. 4.3.3), and future work for this module is outlined (Sect. 4.3.4).

4.3.1 Some unique features

Unlike Mitkov's (1999) original algorithm, this module is extremely customisable. The linguistic patterns to be detected and the scores to be assigned are all defined in an XML file specific to each translation direction. These patterns help identify and

rank potential antecedents, and can include references to various types of surrounding words and even the anaphor whose antecedent is being resolved. The translation pair developer also has the ability to define multiple types of anaphors—such as possessive determiners, reflexive pronouns, zero anaphors, etc.—so as to be able to write separate rules for the resolution of each of them.

4.3.2 Example usage

A sample translation which highlights the usage of *apertium-anaphora* has been given in Table 1. The source sentence goes through a series of modules in the translation pipeline, as described in Sect. 2. The output of the lexical selection module contains a stream of lexical units, including the morphological analysis and the translation of each lexical unit. This is taken as the input to the *apertium-anaphora* module. The lexical unit of the example anaphor, *el seu*, at this stage in the stream is as follows: `^el seu<det><pos><m><sg>/his<det><pos><m><sg>$`

The antecedent of the possessive determiner *el seu* is *els grups* ‘the groups’, which is plural, and hence it should be translated as *their* in English and not *his*. The anaphora resolution module attempts to resolve this anaphor and identify its antecedent by applying all rules that match the context. For instance, the *First NP* rule gives a positive score to the first noun of the sentence (*grups*), as the first noun of a sentence is more likely to be the antecedent of an anaphor. The *Preposition NP* rule gives a negative score to a noun that is part of a prepositional phrase (*Parlament*), as a noun inside a prepositional phrase is less likely to be the antecedent of an anaphor. Both of these tendencies have been observed empirically (Mitkov 1999), and have been implemented as language-specific rules.

After application of all the rules on all candidate antecedents, the candidate with the highest score is considered the most salient antecedent for the anaphor. If the rules are successful, then the correct antecedent should have the highest score (in this case, *bands*). The anaphora resolution module then attaches this antecedent (in the target language) to the lexical unit of the anaphor: `^el seu<det><pos><m><sg>/his<det><pos><m><sg>/band<n><pl>$`

Based on the properties of the attached antecedent, the anaphor is modified during structural transfer (*his* → *their*, as the antecedent is plural), resulting in the following lexical unit: `^their<det><pos><m><sg>$`

The final *Apertium* translation, after each lexical unit output from structural transfer has gone through morphological generation, can be seen in Table 1. The translation of the anaphor is fixed due to the use of *apertium-anaphora* in the pipeline. While the final *Apertium* translation is still not ideal, other aspects of this translation may be fixed through adjustments to other modules in the translation pipeline: the preposition *for* instead of *at* in lexical selection, not using the article *the* with *Parliament* in structural transfer or lexical selection, the placement of the adjunct *this Tuesday* in structural transfer, and *groups* for *bands* in lexical selection.

Table 1 A Catalan–English translation example which highlights a use case for *apertium-anaphora*

Input sentence (Catalan)	Els grups del Parlament han mostrat aquest dimarts <i>el seu</i> suport al batle d'Alaró.
Reference translation (English)	Parliamentary groups showed <i>their</i> support for the mayor of Alaró on Tuesday.
Apertium translation without <i>apertium-anaphora</i> (English)	The bands of the Parliament have shown this Tuesday <i>his</i> support at the mayor of Alaró.
Apertium translation with <i>apertium-anaphora</i> (English)	The bands of the Parliament have shown this Tuesday <i>their</i> support at the mayor of Alaró.

4.3.3 Preliminary evaluation

The *apertium-anaphora* module has been manually evaluated on two language pairs—Spanish–English and Catalan–Italian—by rating the translation of anaphors with and without the module in the pipeline. Since this is a preliminary evaluation, only third person possessive determiners were marked as anaphors.

In Spanish, there is a possessive determiner *su*, which can be translated to English as *his/her/its/their* depending on the gender, number, and animacy of the antecedent. The first 1000 sentences from the Spanish Europarl corpus were translated to English using Apertium with and without *apertium-anaphora* in the translation pipeline, and a basic rule-set was used for the anaphora resolution.⁶ 120 sentences out of these had at least one possessive determiner, and a manual evaluation was done to check the accuracy of the translated anaphors in English.

In Catalan, there is a possessive determiner *el seu* which can translate as *il suo* ‘his/her/its’ or *il loro* ‘their’ in Italian, depending on the number of the antecedent. A corpus was created using articles from *Kataluna Esperantisto*,⁷ a freely available journal, and random paragraphs were translated. A total of 108 sentences had at least one possessive determiner, and a manual evaluation was done to check the accuracy of the translated anaphors in Italian.

The results of these evaluations⁸ are shown in Table 2. Without a module for anaphora resolution, the anaphor just translates to whatever is provided in the bilingual dictionary, which in these pairs is the male singular possessive determiner.

For Spanish–English translation, use of the module led to an increase in accuracy of anaphor translation, but for Catalan–Italian it resulted in a slight decrease in the accuracy of resolution. It is important to note here that the results without anaphora resolution for Catalan–Italian, where all targeted anaphors by default are translated as

⁶ The rule set used is the one contained in the revision of <https://github.com/apertium/apertium-eng-spa/blob/anaphora-transfer/apertium-eng-spa.spa-eng.arx> as of the time of publication.

⁷ The journal can be found at <https://esperanto.cat/Kataluna-Esperantisto>.

⁸ The complete evaluation data can be found at <https://github.com/apertium/apertium-anaphora/tree/master/evaluation>.

Table 2 A preliminary evaluation of translation with and without anaphora resolution (AR) in the pipeline

Systems	Number of anaphors evaluated	Accuracy (%)	
		Without AR	With AR
Spanish–English	120	29.2	54.2
Catalan–Italian	108	83.3	75.0

Accuracy is the percentage of anaphors translated correctly

singular, still showed an accuracy of 83.3%. This indicates that the test data was not evenly distributed in terms of the grammatical number of the antecedents of these anaphors.

It is also important to note that the saliency indicators and their respective scores can be tuned to the domain of the text to get better results. In the preliminary evaluation, the rule-set was modified after an initial look at the results. For example, since the Spanish–English evaluation data was transcribed speech data (Europarl), we were able to add an “impeding indicator” to patterns that contained a proper noun followed by a comma, which attaches a slight negative score to such patterns. These are patterns that are likely to be the speaker addressing an interlocutor, such as *Madam President*, *Mister Speaker*, etc. The interlocutor in these examples is likely to not be the antecedent for a third-person possessive determiner anaphor that follows in the context.

4.3.4 Future work

For now, the linguistic markers used by the anaphora resolution module and their corresponding scores need to be manually defined by language experts. The markers provide linguistic cues for anaphora resolution and the scores are arrived upon empirically.

If these scores can be learnt from a corpus, it would make it much simpler to have an anaphora module with decent accuracy. Since the scope of the rules would be largely defined, it would require much less data to learn the scores as compared to training a corpus-based (machine learning) model to perform anaphora resolution from scratch.

Another idea is to learn these scores from related languages, as the linguistic cues for anaphora resolution shouldn’t vary much among related language pairs. For example, the rules and scores can be learnt from Spanish, which has abundant data, and applied to Catalan, which is a low-resource language.

5 Supporting minoritised languages

It is argued by Kornai (2013) that many languages that are not considered endangered do not have a sufficient level of access to language technology to survive (i.e., maintain intergenerational transmission) in the digital age. He presents evidence of a “massive die-off caused by the digital divide,” and suggests that access to language technology is critical for the continued survival of any currently used language.

We consider MT to be a crucial part of this access to language technology. Specifically, MT allows speakers of a low-resource language to access resources in other languages by translating them into their own language. Additionally, MT enables much more efficient translation of content into low-resource languages—for example, a small team of speakers of a low-resource language may use MT to quickly translate Wikipedia pages from a language with large numbers of high-quality Wikipedia pages. This, of course, requires some attention to post-editing the results of MT, but that is often far less work than translating the information by hand.

It must be stated that the Apertium community does not consider MT to be a single solution for making production-ready translations of texts like marketing materials, literature, and legal documents—a perception that we have encountered anecdotally. Any production-ready translation absolutely requires at a minimum an editor who knows the target language well, and preferably also with expertise in translation from the source language. In such environments, MT is simply a solution that reduces the time investment for human translators to produce a quality translation. It is also meant as a tool for people who do not know the source language to make sense of material in that language. In these ways, MT can be a useful tool for speakers of low-resource languages.

Apertium is designed for rule-based MT. In reference to using corpus-based approaches to developing MT systems for low-resource languages, Martín-Mor (2017) states that “most minoritised languages ... do not have a sufficient number of texts in digital format, because of a lack of digital texts, a lack of consensus on the standardisation models, etc. In those cases, Rule-Based Machine Translation (RBMT) is especially useful, since rules can be manually written even when languages are not fully standardised”. In other words, what can be done with corpus-based approaches is limited when the amount of parallel text is limited. That said, Apertium is open to leveraging corpus-based methods as much as possible given the limitations, as outlined in Sect. 3. In reality, many Apertium pairs are technically hybrid MT systems, although the level of incorporation of corpus-based methods can vary from absolutely none to a rather large amount with recent advancements.

5.1 Released translation pairs

As of December 2020, there are 51 translation pairs released, corresponding to 44 languages of 6 language families (Table 3). See Appendix A for the full list. The vast majority of the languages are Indo-European—and many of these are Romance, Slavic, or Germanic languages. Non Indo-European languages are Afro-Asiatic (Arabic and Maltese), Austronesian (Indonesian and Malay), Turkic (Crimean Tatar, Kazakh, Tatar and Turkish), Uralic (North-Sámi) and isolates (Basque). Table 4 shows quality metrics for some of the released pairs.

For the most part, translation systems are constructed between languages of the same family. There are only three released translators between unrelated languages: North Sámi–Norwegian (Bokmål), Basque–English, and Basque–Spanish. Examples of translation systems developed for translation between closely related languages include Malay–Indonesian, Maltese–Arabic, Dutch–Afrikaans (Otte and

Tyers 2011), Crimean Tatar–Turkish (Gökırmak et al. 2019), and Kazakh–Tatar (Salimzyanov et al. 2013). Even inside subfamilies, translation systems for Romance languages typically target another Romance language (and not other Indo-European languages), and the same is true of Germanic into Germanic and even South Slavic into South Slavic, West Slavic into West Slavic, and East Slavic into East Slavic. There is a heavier density of translation pairs between Romance languages (19 for 12 languages), between Slavic languages (6 for 9 languages), and between Scandinavian languages (5 for 5 languages or language varieties). Two languages tend to break the close-proximity rule: English and Esperanto, which have a significant number of connections with languages outside their sub-family.⁹ Despite this, there is no central language in Apertium: there are 9 translators into both English and Spanish, 8 into Catalan, 4 into both Norwegian (Bokmål) and Esperanto, 3 into both French and Portuguese, etc.

The initial objective of Apertium was to create free and open-source resources for the languages of Spain. In light of the increasing breadth of the published pairs and ongoing work leveraging the Apertium platform (see Table 4), particularly thanks to funding from the Google Summer of Code programme, it may be stated that Apertium has since become a major venue for creating resources for minoritised and low-resource languages in Europe and has shown potential as a language technology platform supporting languages all around the world.

Eleven of the forty-four languages with released translators are considered vulnerable or endangered (Moseley 2010): Aragonese, Arpitan, Asturian, Basque, Belarusian, Breton, Crimean Tatar, North Sámi, Occitan, Sardinian, and Welsh. Other languages hold minority status in their states, like Afrikaans, Catalan, Galician, Silesian, and Tatar. Recent work on other under-resourced and/or minoritised languages includes Bashqort (Tyers et al. 2012b), Bengali (Faridee and Tyers 2009), Chukchi,¹⁰ Gagauz (Bayatli et al. 2018a), Guaraní,¹¹ Qaraqalpaq,¹² Karelian (Pirinen 2019), Kurmanji Kurdish,¹³ Sorani Kurdish (Translators without Borders 2016), Lingala,¹⁴ Malayalam,¹⁵ Marathi (Ravishankar and Tyers 2017), Punjabi,¹⁶ Cuzco Quechua,¹⁷ Lule-Saami (Tyers et al. 2009), South-Saami (Antonsen et al. 2017; Tyers et al. 2009), Sakha,¹⁸ Sicilian,¹⁹ Iraqi Türkman,²⁰ and

⁹ We consider Esperanto to be within a specific constructed subfamily of Indo-European languages.

¹⁰ <https://summerofcode.withgoogle.com/archive/2017/projects/4736366453719040/>.

¹¹ <https://summerofcode.withgoogle.com/archive/2018/projects/5434804640153600/>.

¹² <https://www.google-melange.com/archive/gsoc/2014/orgs/apertium/projects/beknazar.html>, <https://summerofcode.withgoogle.com/archive/2019/projects/6137485212516352/>, <https://summerofcode.withgoogle.com/archive/2020/projects/4815970624864256/>.

¹³ <https://summerofcode.withgoogle.com/archive/2016/projects/5069737520267264/>.

¹⁴ <https://summerofcode.withgoogle.com/archive/2019/projects/4582884889853952/>.

¹⁵ <https://www.google-melange.com/archive/gsoc/2014/orgs/apertium/projects/aboobacker.html>.

¹⁶ <https://summerofcode.withgoogle.com/archive/2020/projects/6209442061746176/>.

¹⁷ https://www.google-melange.com/archive/gsoc/2012/orgs/apertium/projects/pato_yap.html.

¹⁸ <https://summerofcode.withgoogle.com/archive/2018/projects/4877442304966656/>.

¹⁹ <https://summerofcode.withgoogle.com/archive/2016/projects/5883995808071680/>.

²⁰ <https://github.com/apertium/apertium-tki>, <https://wiki.apertium.org/wiki/Apertium-tki>.

Table 3 Released translation systems per language family and sub-family

Language family		Languages	Translation systems	
			In-family	Out-of-family
Afro-Asiatic	Semitic	2	1	0
Austronesic	Malayo-Polynesian	2	1	0
Indo-European		34	44	3
	Celtic	2	0	2
	Germanic	8	7	9
	Indo-Iranian	2	1	0
	Romance	12	19	8
	Slavic	9	6	2
	Constructed	1	0	4
Turkic		4	2	0
Uralic	Finno-Ugric	1	0	1
Basque		1	0	2
Total		44	48	3

Uyghur.²¹ In some cases, coordinated efforts are under way to develop resources for entire language families, such as for Turkic languages (Washington et al. 2021).

5.2 Other languages and work ahead

In Table 5, we show the performance of some unreleased machine-translation systems from previous reports or publications.

An improvement in performance could be possible for these systems with time by improving morphological disambiguation, adding more stems into the dictionaries, and adding or refining lexical and structural transfer rules.

In addition to the language pairs which have been mentioned in Table 5, there are many other pairs that are in various stages of development but have not been systematically evaluated yet, such as Basque–English, Cuzco Quechua–Spanish, Guaraní–Spanish, Karelian–Finnish, Kazakh–Kyrgyz, Kazakh–Russian, Lingala–English, Marathi–Hindi, Sorani Kurdish–Kurmanji Kurdish, Turkish–Uzbek, and Uzbek–Qaraqalpaq, among others.

²¹ <https://summerofcode.withgoogle.com/archive/2018/projects/5988796768190464/>, <https://summerofcode.withgoogle.com/archive/2019/projects/5106764196872192/>.

Table 4 Released translation pairs with available evaluation data

Systems	Coverage (%)	WER (%)	PER (%)	BLEU (0–1)
Aragonese–Spanish ^a	94.33	11.61–14.12		0.72–0.79
Belarusian–Russian ^b	84.3	25.72		
Breton–French ^c	87–90	38	22	
Catalan–Aragonese ^d	87.6–93.2	19.37	17.85	
Catalan–Italian ^e	94.7	14.2		
Catalan–Romanian ^f	88.7		29	
Catalan–Sardinian ^g	94.4	20.5	13.9	
Danish–Bokmål Norwegian ^h	88.1–95.9	10.87		
Danish–Nynorsk Norwegian ⁱ	87.3–92.7	13.64–22.64		
French–Arpitan ^j	92.8–95.8	5.7		
French–Occitan ^k	92.3	10.0		
Italian–Catalan ^l	91.2	15.7		
Italian–Sardinian ^m	89.3–96.4	9.9		
North Sámi–Bokmål Norwegian ⁿ	77.52–94.72	39.68–53.31		
Nynorsk–Bokmål Norwegian ^o	92.6–99.2	10.71		
Portuguese–Catalan ^p	91.4	14.0		
Romanian–Catalan ^q	86.8		46	
Russian–Belarusian ^r	83.6	23.93		
Spanish–Aragonese ^s	95.22	16.83–19.37		0.65–0.71
Serbo-Croatian–Macedonian ^t	74.5–90.96	48.33	48.33	0.36
Swedish–Danish ^u	83.7–88.0	31		
Ukrainian–Russian ^v	80.9–90.0	14.74		
Welsh–English ^w		53.40–64.94	27.22–34.35	0.16–0.32

Coverage is the percentage of tokens which receive at least one analysis from the morphological analyser. WER (Word Error Rate), PER (Position-independent Word Error Rate), and BLEU scores are computed against a reference translation. A relatively low WER/PER score or a relatively high BLEU score generally denotes better translation quality

^a Martínez Cortés et al. (2012)

^b http://wiki.apertium.org/wiki/Belarusian_and_Russian/Work_plan.

^c Tyers (2009, 2010)

^d http://wiki.apertium.org/wiki/Aragonese_and_Catalan/Evaluation.

^e http://wiki.apertium.org/wiki/Hectoralos/GSOC_2019_final_report.

^f http://wiki.apertium.org/wiki/Romanian_and_Catalan/GSOC_2018.

^g Fronteddu et al. (2017)

^h http://wiki.apertium.org/wiki/Scandinavian_MT_project.

ⁱ http://wiki.apertium.org/wiki/Scandinavian_MT_project.

^j https://wiki.apertium.org/wiki/Hectoralos/GSOC_2020_rapport_final.

^k http://wiki.apertium.org/wiki/User:Capsot/GSOC_2018_Occitan_French.

^l http://wiki.apertium.org/wiki/Hectoralos/GSOC_2019_final_report.

^m Tyers et al. (2017)

ⁿ Trosterud and Unhammer (2012)

Table 4 (continued)

- ^o http://wiki.apertium.org/wiki/Scandinavian_MT_project.
^p http://wiki.apertium.org/wiki/Hectoralos/GSOC_2019_final_report.
^q http://wiki.apertium.org/wiki/Romanian_and_Catalan/GSOC_2018.
^r http://wiki.apertium.org/wiki/Belarusian_and_Russian/Work_plan.
^s Martínez Cortés et al. (2012)
^t Peradin and Tyers (2012)
^u http://wiki.apertium.org/wiki/Scandinavian_MT_project.
^v http://wiki.apertium.org/wiki/Russian_and_Ukrainian/Work_plan.
^w Tyers and Donnelly (2009)

Table 5 A selection of unreleased translation pairs with published results

Systems	Coverage (%)	WER (%)	PER (%)	BLEU (0–1)
Kazakh–Turkish ^a	83.42	45.77	41.69	0.17
North Sámi–Finnish ^b	76.81	34.24	–	–
North-Saami–South-Saami ^c	87.4	54.84	30.94	–
Tatar–Bashqort ^d	70.19	8.97	–	–

Coverage is the percentage of tokens which receive at least one analysis from the morphological analyser. WER (Word Error Rate), PER (Position-independent Word Error Rate), and BLEU scores are computed against a reference translation. A relatively low WER/PER score or a relatively high BLEU score generally denotes better translation quality

^aBayatli et al. (2018b)

^bJohnson et al. (2017)

^cAntonsen et al. (2017)

^dTyers et al. (2012b)

6 Supplementary tools

This section highlights supplementary tools maintained by Apertium that are useful for developers as well as end-users. Apertium-viewer (Sect. 6.1) is particularly useful for developers interacting with Apertium resources. The Apertium website software (Sect. 6.2) provides access to free and unlimited translation, morphological analysis, and several under-development features like dictionary lookup and a spell-checking interface, each of which can prove very useful for end-users who do not wish to install any software locally.

6.1 Apertium-viewer

Apertium-viewer is a tool that makes it straightforward for users to view and edit the output of the various stages of an Apertium translation. It reads a translation pair's "modes" configuration file, where the specific pipeline for the translator is defined.

It displays how a text changes as it cascades through the modules, from the source to the target language. The user can change the text string at every stage to see how subsequent stages, including the output translation, are affected. This tool can be useful for understanding translation pairs and debugging translations.

6.2 Website software

Apertium offers an open-source web API and customisable website front-end²² (Cherivirala et al. 2018). Apart from translating text, users can provide a URL to a webpage, which will be translated with the formatting preserved. Note that the Apertium API and website software can also be deployed by anyone for any purpose. The software also provides a front-end to morphological transducers, and there are a number of beta features under development.

One of these features is multi-step translation, where a user can use the interface to translate from one language to another for which there isn't an Apertium translation pair via one or more pivot languages.²³

Another feature under development is dictionary lookup, where a user may use the Apertium website as an online dictionary. That is, a word is not simply translated, but all possible translations of the word are provided. The community hopes to include some additional features with dictionary lookup, including automatic reverse lookups (so that a user may gain a better understanding of the results), grammatical information (such as the gender of nouns or the conjugation paradigms of verbs), and information about MWEs.

A suggestions feature allows users to suggest corrections to translations. This is especially helpful as developers can incorporate these corrections back into the systems.

One last feature under development is a spell-checking interface. This feature provides users with a simple interface to check the spelling of words in a text, and to be offered suggestions for misspelled words. It is noteworthy that there are no known spell checkers available for some of the languages with dictionaries in Apertium, such as Arpitan.

7 Conclusion

We have presented the latest updates to Apertium, a free and open-source platform for machine translation, with a focus on MT for low-resource languages. These updates include approaches to hybridisation of Apertium modules with corpus-based approaches, new modules that are available for the Apertium RBMT pipeline, and newly released languages pairs.

The new modules in the pipeline are all optional since they may be useful for some specific languages pairs, but would not significantly improve others. With an increasing

²² Currently in use on <https://apertium.org/>; source code available at <https://github.com/apertium/apertium-apy/> and <https://github.com/apertium/apertium-html-tools/>, respectively.

²³ This feature is enabled and available at <https://beta.apertium.org/>.

number of released language pairs, Apertium becomes a preferred vehicle for translation to and from low-resource languages, which are not as easily implemented using widely advocated neural approaches to MT due to sparsity of available text, and are also not considered economical for corporate work. In addition, the different sub-components of translation pairs can be and are used independently to produce other types of resources for these languages, such as electronic dictionaries, a tool for searches in electronic corpora (e.g., Saykhunov et al. 2019), spell checkers,²⁴ and tools supporting language learning, maintenance, and revitalisation (e.g., Katinskaia et al. 2018; Ivanova et al. 2019).

Appendix A: List of released languages and translation pairs

The released languages are:²⁵

Afrikaans*	Galician	Russian*
Arabic*	Hindi*	North Sámi*
Aragonese*	Icelandic	Sardinian*
Arpitan*	Indonesian*	Serbo-Croatian*
Asturian	Italian	Silesian*
Basque	Kazakh*	Slovenian*
Belarusian*	Macedonian	Spanish
Breton	Malaysian*	Swedish
Bulgarian	Maltese*	Tatar*
Catalan	Norwegian Bokmål	Crimean Tatar*
Danish	Norwegian Nynorsk	Turkish*
Dutch*	Occitan	Ukrainian*
English	Polish*	Urdu*
Esperanto	Portuguese	Welsh
French	Romanian	

²⁴ Including, for example, http://grammar.corpus.tatar/index_en.php?of=search/spellchecker.php.

²⁵ An asterisk (*) indicates that the language has been released since the previous publication (Forcada et al. 2011). Norwegian Bokmål and Norwegian Nynorsk are considered two different languages in Apertium since there is a translator from one to the other, which is not the case between different varieties of Catalan, Occitan and Portuguese that are supported in Apertium.

The released language pairs (with indication of the translation directions and novelty) are:

Afrikaans \rightleftharpoons Dutch*	Galician \rightleftharpoons Portuguese
Aragonese \rightleftharpoons Catalan*	Galician \rightleftharpoons Spanish
Aragonese \rightleftharpoons Spanish*	Hindi \rightleftharpoons Urdu*
Basque \rightarrow English*	Icelandic \rightarrow English
Basque \rightarrow Spanish	Icelandic \rightleftharpoons Swedish*
Belarusian \rightleftharpoons Russian*	Indonesian \rightleftharpoons Malaysian*
Breton \rightarrow French	Italian \rightarrow Sardinian*
Bulgarian \rightleftharpoons Macedonian	Kazakh \rightleftharpoons Tatar*
Catalan \rightleftharpoons English	Macedonian \rightarrow English
Catalan \rightarrow Esperanto	Maltese \rightarrow Arabic*
Catalan \rightleftharpoons French	Norwegian \rightleftharpoons Swedish*
Catalan \rightleftharpoons Italian	Norwegian Bokmål \rightleftharpoons Nynorsk
Catalan \rightleftharpoons Occitan	Occitan \rightleftharpoons Spanish
Catalan \rightleftharpoons Portuguese	Polish \rightarrow Silesian*
Catalan \rightleftharpoons Romanian*	Portuguese \rightleftharpoons Spanish
Catalan \rightarrow Sardinian*	Romanian \rightarrow Spanish
Catalan \rightleftharpoons Spanish	Russian \rightleftharpoons Ukrainian*
Danish \rightleftharpoons Norwegian*	North Sámi \rightarrow Norwegian*
Danish \rightleftharpoons Swedish	Serbo-Croatian \rightarrow English*
English \rightleftharpoons Esperanto	Serbo-Croatian \rightarrow Macedonian*
English \rightleftharpoons Galician	Serbo-Croatian \rightleftharpoons Slovenian*
English \rightleftharpoons Spanish	Spanish \rightarrow Asturian
French \rightarrow Arpitan*	Spanish \rightarrow Esperanto
French \rightarrow Esperanto	Crimean Tatar \rightarrow Turkish*
French \rightarrow Occitan*	Welsh \rightarrow English
French \rightleftharpoons Spanish	

It should be noted that many of the pairs already released in the last publication (Forcada et al. 2011) have been updated. For example, the Catalan–French pair previously had a bilingual dictionary of 10,554 entries, while in December 2020 it has 71,537.

Acknowledgements Several of the advances described in this paper were supported by Google Summer of Code funding, for which the authors are very grateful. The authors also recognise the Hungerford Faculty Support Fund at Swarthmore College and the University of Chicago’s Department of Linguistics Graduate Research Aid Initiative in Linguistics (GRAIL) for their generous support in covering open-access publication costs.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission

directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.


References

- Antonsen L, Trosterud T, Tyers FM (2017) A North Saami to South Saami machine translation prototype. *Lecture Notes Artif Intell* 4:11–27. <https://doi.org/10.3384/nejlt.2000-1533.1642>
- Baldwin B (1997) CogNIAC: high precision coreference with limited knowledge and linguistic resources. In: *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, Association for Computational Linguistics, pp 38–45, 10.3115/1598819.1598825, <http://portal.acm.org/citation.cfm?doid=1598819.1598825>
- Bayatli S, Karanfil G, Gökırmak M, Tyers FM (2018a) Finite-state morphological analysis for Gagauz. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Miyazaki, Japan, <https://www.aclweb.org/anthology/L18-1411>
- Bayatli S, Kurnaz S, Salimzianov I, Washington JN, Tyers FM (2018b) Rule-based machine translation from Kazakh to Turkish. In: *European Association for Machine Translation (EAMT)*, pp 49–58
- Bayatli S, Kurnaz S, Ali A, Washington JN, Tyers FM (2020) Unsupervised weighting of transfer rules in rule-based machine translation using maximum-entropy approach. *J Inf Sci Eng* 36(2):309–322. [https://doi.org/10.6688/JISE.202003_36\(2\).0010](https://doi.org/10.6688/JISE.202003_36(2).0010)
- Bick E, Didriksen T (2015) CG-3 - beyond classical constraint grammar. In: *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015(May)*, pp 11–13 (2015) Vilnius. Linköping University Electronic Press, Linköpings universitet, Lithuania, pp 31–39
- Cherivirala S, Chiplunkar S, Washington J, Unhammer K (2018) Apertium's web toolchain for low-resource language technology. In: *Proceedings of the AMTA 2018 Workshop on Technologies for MT of Low Resource Languages (LoResMT 2018)*, pp 53–62, <https://www.aclweb.org/anthology/W18-2207/>
- Constant M, Eryiğit G, Monti J, van der Plas L, Ramisch C, Rosner M, Todirascu A (2017) Multiword expression processing: a survey. *Comput Linguist* 43(4):837–892. https://doi.org/10.1162/COLI_a_00302
- Cutting D, Kupiec J, Pedersen J, Sibun P (1992) A practical part-of-speech tagger. In: *Proceeding of the 3rd conference on applied natural language processing*. Trento, pp 133–140
- Faridee AZM, Tyers FM (2009) Development of a morphological analyser for Bengali. In: Pérez-Ortiz J, Sánchez-Martínez F, Tyers F (eds) *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, Universidad de Alicante. Departamento de Lenguajes y Sistemas Informáticos, Alicante, Spain, pp 43–50
- Forcada ML, Ginestí-Rosell M, Nordfalk J, O'Regan J, Ortiz-Rojas S, Pérez-Ortiz JA, Sánchez-Martínez F, Ramírez-Sánchez G, Tyers FM (2011) Apertium: a free/open-source platform for rule-based machine translation. *Mach Transl* 25(2):127–144
- Fronteddu G, Alòs i Font H, Tyers FM (2017) Una eina per a una llengua en procés d'estandardització: el traductor automàtic català–sard. *Linguamàtica* 9(3):3–20
- Gökırmak M, Tyers FM, Washington JN (2019) A free/open-source rule-based machine translation system for Crimean Tatar to Turkish. In: *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages (LoResMT 2019, at Machine Translation Summit XVII)*, pp 24–31, <https://www.aclweb.org/anthology/W19-68#page=30>
- Ivanova S, Katinskaia A, Yangarber R (2019) Tools for supporting language learning for Sakha. In: *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, Linköping University Electronic Press, Turku, Finland, pp 155–163, <https://www.aclweb.org/anthology/W19-6117>
- Johnson R, Pirinen TA, Puolakainen T, Tyers F, Trosterud T, Unhammer K (2017) North-Sámi to Finnish rule-based machine translation system. In: *Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22–24 May 2017, Gothenburg, Sweden*, Linköping University Electronic Press, 131, pp 115–122
- Karlsson F, Voutilainen A, Heikkilä J, Anttila A (1995) *Constraint Grammar: a language-independent system for parsing unrestricted text*, vol 4. Walter de Gruyter

- Katinskaia A, Nouri J, Yangarber R (2018) Revita: a language-learning platform at the intersection of ITS and CALL. In: Proceedings of LREC: 11th International Conference on Language Resources and Evaluation, Miyazaki, Japan
- Kornai A (2013) Digital language death. *PLoS ONE* 8(10):e77056. <https://doi.org/10.1371/journal.pone.0077056>
- Lappin S, Leass HJ (1994) An algorithm for pronominal anaphora resolution. *Comput Linguist* 20(4):535–561
- Lee H, Chang A, Peirsman Y, Chambers N, Surdeanu M, Jurafsky D (2013) Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Comput Linguist* 39(4):885–916
- Lindén K, Axelsson E, Hardwick S, Silfverberg M, Pirinen T (2011) HFST—framework for compiling and applying morphologies. In: Proceedings of Second International Workshop on Systems and Frameworks for Computational Morphology, pp 67–85
- Loáiciga S, Wehrli É (2015) Rule-based pronominal anaphora treatment for machine translation. In: Proceedings of the Second Workshop on Discourse in Machine Translation, Association for Computational Linguistics, pp 86–93, <https://doi.org/10.18653/v1/W15-2512>, <http://aclweb.org/anthology/W15-2512>
- Martín-Mor A (2017) Technologies for endangered languages: the languages of Sardinia as a case in point. *mTm J* 9:365–386
- Martínez FS (2008) Using unsupervised corpus-based methods to build rule-based machine translation systems. phdthesis, Universidad de Alicante, <https://www.dlsi.ua.es/~fsanchez/pub/thesis/thesis-sin.pdf>
- Martínez Cortés JP, O'Regan J, Tyers F (2012) Free/open source shallow-transfer based machine translation for Spanish and Aragonese. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), European Language Resources Association (ELRA)
- Marting M, Unhammer KB (2014) FST trimming: ending dictionary redundancy in Apertium. In: Proceedings of the 9th conference on language resources and evaluation, pp 19–24
- Mitkov R (1999) Multilingual anaphora resolution. *Mach Transl* 14:281–299
- Moseley C (ed) (2010) Atlas of the World's Languages in Danger, 3rd edn. UNESCO, <http://www.unesco.org/languages-atlas/index.php>
- Otte P, Tyers FM, (2011) Rapid rule-based machine translation between Dutch and Afrikaans. In: EAMT, (2011) proceedings of the 15th conference of the European Association for Machine Translation, 30–31 May 2011. Belgium, Leuven
- Peradin H, Tyers FM (2012) A rule-based machine translation system from Serbo-Croatian to Macedonian. In: Third international workshop on free/open-source rule-based machine translation (Free RBMT 2012), pp 55–63
- Pirinen TA (2019) Workflows for kickstarting RBMT in virtually no-resource situation. In: Proceedings of the 2nd workshop on technologies for MT of low resource languages
- Ravishankar V, Tyers FM (2017) Finite-state morphological analysis for Marathi. In: Proceedings of the 13th international conference on finite state methods and natural language processing
- Salimzyanov I, Washington JN, Tyers FM (2013) A free/open-source Kazakh-Tatar machine translation system. In: Sima'an K, Forcada M, Grasmick D, Depraetere H, Way A (eds) Proceedings of the XIV machine translation summit. European Association for Machine Translation, pp 175–182, <http://www.mt-archive.info/10/MTS-2013-Salimzyanov.pdf>
- Sánchez-Cartagena VM, Pérez-Ortiz JA, Sánchez-Martínez F (2015) A generalised alignment template formalism and its application to the inference of shallow-transfer machine translation rules from scarce bilingual corpora. *Comput Speech Lang* 32(1):46–90 <https://doi.org/10.1016/j.csl.2014.10.003>
- Sánchez-Martínez F, Forcada ML (2007) Automatic induction of shallow-transfer rules for open-source machine translation. In: Way A, Gawronska B (eds) Proceedings of the 11th conference on theoretical and methodological issues in machine translation (TMI 2007), Skövde University Studies in Informatics, 2007(1):181–190
- Sánchez-Martínez F, Forcada ML (2009) Inferring shallow-transfer machine translation rules from small parallel corpora. *J Artif Intell Res* 34:605–635
- Sánchez-Martínez F, Ney H (2006) Using alignment templates to infer shallow-transfer machine translation rules. In: Salakoski T, Ginter F, Pyysalo S, Pahikkala T (eds) Advances in natural language processing, vol 4139, Springer, Berlin Heidelberg, pp 756–767, https://doi.org/10.1007/11816508_75, http://link.springer.com/10.1007/11816508_75

- Sánchez-Martínez F, Pérez-Ortiz JA, Forcada ML (2006) Speeding up target language driven part-of-speech tagger training for machine translation. In: Proceedings of the 5th Mexican International conference on artificial intelligence, MICAI 2006, pp 844–854
- Sánchez-Martínez F, Armentano-Oller C, Pérez-Ortiz JA, Forcada ML (2007) Training part-of-speech taggers to build machine translation systems for less-resourced language pairs. In: *Procesamiento del Lenguaje Natural*, (XXIII Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural, Sevilla, Spain, 10–12/09/2007, pp 257–264
- Saykhunov M, Khusainov R, Ibragimov T (2019) Složnosti pri sozdanii tekstovogo korpusa ob'jomom bolee 400 mln tokenov. In: *Finno-ugorskij mir v polietničnom prostranstve Rossii: kulturnoje nasledije i novyje vyzovy: sbornik statej po materialam VI Vserossijskoj naučnoj konferencii finno-ugrovedov*, UdmFITS UrO RAN, pp 548–554
- Swanson DG, Washington JN, Tyers FM, Forcada ML (2021) A tree-based structural transfer module for the Apertium machine translation platform. *Machine Translation* (in press)
- Translators without Borders (2016) Translators without Borders develops world's first crisis-specific machine translation for Kurdish. Slator <https://slator.com/press-releases/translators-without-borders-develops-the-worlds-first-crisis-specific-machine-translation-system-for-kurdish-languages/>
- Trosterud T, Unhammer KB (2012) Evaluating North Sámi to Norwegian assimilation RBMT. In: *Third international workshop on free/open-source rule-based machine translation (FreeRBMT 2012)*, pp 13–25
- Trouilleux F (2002) A rule-based pronoun resolution system for French. *4th Discourse Anaphora and Anaphor Resolution Colloquium* p 7
- Tyers FM (2009) Rule-based augmentation of training data in Breton–French statistical machine translation. In: *Proceedings of the 13th annual conference of the european association of machine translation, EAMT09*, pp 213–218
- Tyers FM (2010) Rule-based Breton to French machine translation. In: *Proceedings of the 14th annual conference of the European association of machine translation, EAMT10*, pp 174–181
- Tyers FM, Donnelly K (2009) apertium-cy—a collaboratively-developed free RBMT system for Welsh to English. *Prague Bull Math Linguist* 91:57–66
- Tyers FM, Wiecheteck L, Trosterud T (2009) Developing prototypes for machine translation between two Sámi languages. In: *Proceedings of the 13th Annual Conference of the European Association of Machine Translation, EAMT09*, pp 120–128
- Tyers FM, Sánchez-Martínez F, Forcada ML, et al. (2012a) Flexible finite-state lexical selection for rule-based machine translation. In: *Proceedings of the 16th EAMT Conference, European association for machine translation*
- Tyers FM, Washington JN, Salimzyanov I, Batalov R (2012b) A prototype machine translation system for Tatar and Bashkir based on free/open-source components. In: *First workshop on language resources and technologies for Turkic languages*, p 11
- Tyers FM, Sánchez-Martínez F, Forcada ML (2014) Unsupervised training of maximum-entropy models for lexical selection in rule-based machine translation. In: *Proceedings of the 18th annual conference of the european association for machine translation*, pp 145–153
- Tyers FM, Fronteddu G, Alòs i Font H, Martín-Mor A (2017) Rule-based machine translation for the Italian–Sardinian language pair. *Prague Bull Math Linguist* pp 221–232 <https://doi.org/10.1515/pralin-2017-0022>
- Washington JN, Salimzyanov I, Tyers FM, Gökırmak M, Ivanova S, Kuyrukçu O (to appear) Free/open-source technologies for Turkic languages developed in the Apertium project. In: *Proceedings of the seventh international conference on computer processing of Turkic languages (TurkLang 2019)*
- Zeldes A, Zhang S (2016) When annotation schemes change rules help: A configurable approach to coreference resolution beyond ontonotes. In: *Proceedings of the NAACL2016 workshop on coreference resolution beyond ontonotes (CORBON)*, pp 92–101
- Zhang Y, Clark S (2011) Syntactic processing using the generalized perceptron and beam search. *Comput Linguistic* 37(1):105–151 https://doi.org/10.1162/coli_a_00037

Authors and Affiliations

Tanmai Khanna¹  · Jonathan N. Washington²  · Francis M. Tyers^{3,8}  ·
Sevilay Bayatlı⁴  · Daniel G. Swanson²  · Tommi A. Pirinen⁵  · Irene Tang⁶  ·
Hèctor Alòs i Font⁷ 

Tanmai Khanna
tanmai.khanna@research.iiit.ac.in

Francis M. Tyers
ftyers@iu.edu

Sevilay Bayatlı
sewaletaha@beykent.edu.tr

Daniel G. Swanson
dswanso1@gmail.com

Tommi A. Pirinen
tommi.pirinen@uit.no

Irene Tang
itang1@uchicago.edu

Hèctor Alòs i Font
hectoralos@gmail.com

¹ Language Technologies Research Centre, IIIT Hyderabad, Hyderabad, Telangana 500032, India

² Swarthmore College, Swarthmore, PA 19081, USA

³ Indiana University, Bloomington, IN 47401, USA

⁴ Beykent Üniversitesi, İstanbul, Turkey

⁵ UiT—Norgga árkálaš universitehta, 9000 Tromsø, Norway

⁶ University of Chicago, Chicago, IL 60637, USA

⁷ Centre de Recerca en Sociolingüística i Comunicació, Universitat de Barcelona, Barcelona, Spain

⁸ School of Linguistics, Higher School of Economics, Moscow, Russia