REMU – Reliable Multilingual Digital Communication: Methods and Applications

Aarne Ranta, **Koen Claessen**, **Gerardo Schneider**,
Ramona Enache, **Normunds Gruzitis**, **John J. Camilleri**, **Inari Listenmaa**, **Prasanth Kolachina**

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

The REMU project addresses the problem of reliable multilingual communication in a digital setting. It is a cooperation of research groups from language technology, software testing and formal methods. REMU has three tracks: a translation track, in which human language translation is approached with methods from compiler technology, to guarantee meaning preservation; a testing track, in which software testing techniques are applied to grammars of natural language, to guarantee their quality; and a formal methods track, where logical reasoning is applied to documents written in natural language.

1. Introduction

To understand the goals of the REMU project, let us consider a possible use case: a service in which two participants can prepare a rental contract for a house. The house might be owned by an Italian person, located in Germany, and rented by a Swede. Three languages would thereby be involved, to make sure that the owner and the tenant understand each other accurately and that the contract is compliant with German regulations. It should be customised to the details relevant to the house and also to the wishes of the owner and the tenant. Ideally, all of the partners should be able to pose questions such as, can the contract be transferred to a third person, and get answers via an inference engine without reading the whole contract, let alone involving an expert lawyer.

Today, such services are scarce because they require manual work. Translations to different languages must be made manually, because automatic tools such are not reliable enough. The inference required for question answering is equally manual, because the information contained in contracts is not formal enough to be reasoned mechanically; methods like string-based search are not accurate enough. The aim of this project is to solve both problems, with a solution that is common to a great extent.

The solution is to use controlled natural language (CNL), which is a subset of natural language with a formally specifiable structure. Our use of CNL is inspired by compiler technology, where abstract syntax is a formal structure underlying programming languages. When compilers analyse programs and reason about them, they work on the abstract syntax. This idea is adapted to natural languages in the Grammatical Framework (GF) (Ranta 2011), which moreover allows the mapping between abstract syntax and multiple simultaneous languages. For the mentioned use case, the following workflows are possible by using GF and abstract syntax trees (AST):

Static translation of contracts:
contract in German/Italian/Swedish/...
→ contract AST
→ contract in German/Italian/Swedish/...

Corrections and updates of the contract:
changes in German/Italian/Swedish/...
→ changes in AST
→ changes in German/Italian/Swedish/...

Queries about the contract:
question in German/Italian/Swedish/...
→ question in AST
→ answer in AST
→ answer in German/Italian/Swedish/...

In addition to grammars, REMU uses formal methods, such as automated reasoning and software verification. Reasoning is used for question answering, but also for consistency checking of documents. Software verification is in this project applied to a new field: to computational grammars, which are very complex programs often created in collaborative and distributed ways. Their quality is crucial for the reliability of the overall system.

An example of how formal methods can be applied in this context is ambiguity analysis of computational grammars. In the contract example above, it is crucial that all parties understand each other unambiguously, that is, the text of each contract can only be understood in one way. It is a fact of life that the use of natural language leaves open different ways of interpreting the same text. Therefore, it is important to analyse the grammars involved to firstly be made aware of ambiguities, and secondly, avoid their presence in formal documents such as contracts.

2. Background

Most tools for automatic translation today target consumers of information and promise browsing quality. Consumers use the tools at their own risk, and no-one is responsible for the translations — neither the original author, nor the translation system provider. REMU's focus is on automatic translation tools for producers. These tools should be quickly adaptable to the frequently changing information and render it accurately in the targeted languages. What makes this possible is that the producers know what they

need to say: for instance, that they only need to publish e-commerce offers or rental contracts. Therefore they can use translation tools that work on limited domains, and can therefore be made reliable. Consumers translation tools, in contrast, must work on an open domain. This means that they must be able to cope with any documents thrown at them - but the users are happy with browsing quality.

REMU builds partly on the European **MOLTO** project (Multilingual On-Line Translation, FP7-ICT-247914, 2010–2013). MOLTO's goal was to make it easy to produce translation systems for new domains and languages, via software tools and libraries. The methods were tested on several domains and on up to 17 simultaneous languages. The main innovation in REMU is the introduction of formal methods in the loop, which at the same time permits scaling up the applications in a reliable way by semi-automatic means using statistics and machine learning.

3. Research Tracks

3.1 Translation

One line of research in REMU is developing high-precision "production-quality" translation systems for open-domain. Traditional GF translation systems have small, domain-specific interlinguas and grammars. However, the architecture equally allows for adding large, domain-independent interlinguas and grammars. The novelty of the architecture is the combination of *both* coverage and quality in one and the same system. In this project, the interlingua, based on **GF Resource Grammar Library** (RGL, (Ranta, 2009)) defines a large-scale generic grammar, while more robust grammars or domain-specific grammars or both are defined as embedded controlled languages. One direction of our study is the use of GF interlingua in statistical models used in syntactic machine translation (Zollmann and Venugopal, 2006; Yamada and Knight, 2001).

An additional direction of research relevant to the goals of the project is in the evaluation of translation systems. The linguistic resources in the GF Resource Grammar Library combined with the large-scale interlingua allow the possibility of quality-estimation (Specia et al., 2010) for general-purpose MT systems using these resources.

As we move on to more open-domain translation, it is important to handle non-compositional constructs in the translation grammar. For this purpose, we have created a new module for *constructions*, situated between syntax and lexicon (Goldberg, 1995). We parsed bilingual aligned texts with a wide-coverage GF grammar and extracted phrases that were not syntactically equivalent as candidates, and added relevant findings to the new GF module (Enache et al., 2014).

The large multilingual GF dictionary is a key component in the automatic translation. In the case of verbs, it provides not only translation equivalents of a particular verb sense but also language-specific syntactic valence — prepositions or cases that are required for the verb arguments.

As one way to check the semi-automatically specified valence information, as well as to add missing verb entries, we have extracted semantico-syntactic verb valence patterns from FrameNet-annotated corpora, currently for English and Swedish (Dannélls and Gruzitis, 2014b). FrameNet is based on the theory of frame semantics (Fillmore et al., 2003), where a frame represents a cognitive scenario that is characterised by core and non-core frame elements (FE). While it is generally difficult to distinguish between verb arguments and adjuncts in a purely grammatical analysis of a sentence, core and non-core FEs correspond, according to FrameNet, to arguments and adjuncts respectively, helping to make entries in the dictionary more consistent.

From the extracted patterns, we have also generated a currently bilingual but potentially multilingual semantic grammar on its own, providing a FrameNet-based abstraction layer to GF RGL (Dannélls and Gruzitis, 2014a).

3.2 Testing

Two main cases for grammar testing are ambiguity and adequacy testing. Testing for ambiguities is essential since the CNL described by the grammar is an interface to a logical system, whose functionality can be compromised by language ambiguities. In our case, the situation is even more complicated due to the multilingual context, since we also need to resolve cases of ambiguous translation between natural languages, within the CNL (Ranta and Angelov, 2010).

Adequacy testing entails checking that the semantics from the abstract syntax is preserved in all concrete syntaxes. A prototype of this method was used in (Ranta et al., 2011), a tourist phrasebook grammar available in 15 languages, where each new language added was tested by comparing its constructions with their equivalents in English and the abstract syntax.

3.3 Formal Methods

Formal methods are a means to verify grammars, but they can also be applied to the documents created using grammars. We are specifically interested in the formal analysis of *contracts* — normative texts describing the rights and obligations of various parties. Such documents may include rental agreements, employment contracts and terms of use.

The first part of this task involves modelling documents using some formalism. For this we are developing a formalism based on existing work in the area of e-contracts (Diaz et al., 2013), (Marjanovic and Milosevic, 2001). Such models can then be used in analysis techniques for reasoning about logical inconsistencies, causality of events and dependencies. This is done by adapting existing techniques from static analysis and model checking.

As it is not possible to perform automatic reasoning directly on documents written in natural language, we have defined an abstract but expressive CNL as a target language for reasoning (Camilleri et al., 2014). By using GF we can easily add linearisations of the abstract structure in many further languages, without having to adapt the analysis techniques for each new translation added.

Acknowledgements

The authors would like to acknowledge the Swedish Research Council for financial support under grant nr. 2012-5746 (Reliable Multilingual Digital Communication: Methods and Applications), as well as the collaborations with Dana Dannélls, Krasimir Angelov and Thomas Hallgren.

References

John J. Camilleri, Gabriele Paganelli, and Gerardo Schneider. 2014. A CNL for Contract-Oriented Diagrams. In *Proceedings of CNL 2014, Galway*, LNCS.

Dana Dannélls and Normunds Gruzitis. 2014a. Controlled natural language generation from a multilingual FrameNet-based grammar. In *Proceedings of CNL 2014, Galway*, LNCS.

Dana Dannélls and Normunds Gruzitis. 2014b. Extracting a bilingual semantic grammar from FrameNet-annotated corpora. In *Proceedings of LREC 2014*, pages 2466–2473.

Gregorio Diaz, Maria Emilia Cambronero, Enrique Martinez, and Gerardo Schneider. 2013. Specification and Verification of Normative Texts using C-O Diagrams. *IEEE Transactions on Software Engineering*.

Ramona Enache, Inari Listenmaa, and Prasanth Kolachina. 2014. Handling non-compositionality in multilingual cnls. In *Proceedings of CNL 2014, Galway*, LNCS.

Charles J. Fillmore, Christopher R. Johnson, and Miriam R. L. Petruck. 2003. Background to Framenet. *International Journal of Lexicography*, 16(3):235–250.

Adele E Goldberg. 1995. *Construction grammar*. Wiley Online Library.

Olivera Marjanovic and Zoran Milosevic. 2001. Towards formal modeling of e-contracts. In *Proceedings of the 5th IEEE International Conference on Enterprise Distributed Object Computing*, EDOC '01, pages 59–68, Washington, DC, USA. IEEE Computer Society.

Aarne Ranta and Krasimir Angelov. 2010. Implementing Controlled Languages in GF. In *Proceedings of CNL 2009, Marettimo*, LNCS.

Aarne Ranta, Ramona Enache, and Grgoire Dtrez. 2011. Controlled Language for Everyday Use: the MOLTO Phrasebook. *Proceeding of CNL 2010, Zurich*.

Aarne Ranta. 2009. The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine Translation*, 24(1):39–50.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France, July. Association for Computational Linguistics.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June. Association for Computational Linguistics.