# Introduction

**Flammie A Pirinen**
Department of Language and Culture / Divvun
NO-9019 UiT–The Arctic University of Norway
*Flammie.pirinen@uit.no*

## 1 Introduction

*Rule-based language technology (RBLT)* refers to a language technology approach grounded in *rules*. As rules we understand grammatical rules written by linguists but in a broader view, also any world knowledge and logical deductions are rules. The rule-based language technology is indeed often seen as a combination of general linguistics and software engineering or computer science. The purpose of the computational component can be seen as a research question in its own right or as a practical engineering problem of how we express the linguistic grammar in a computational form to be used by the computer.

One of the core ideas of linguistics as it is understood by language technology engineers is the ability to *abstract* away from the specifics in order to generalize the thinking. That is, for example, to understand that a relationship between wordforms 'cat' and 'cats,' and 'dog' and 'dogs' is the same, is a very useful piece of information in rule-based language technology, and eventually understanding that perhaps the same relationship applies between 'mouse' and 'mice'. Modelling it accurately is the task we often refer to when talk about language modelling. Modelling of a whole language naturally encompasses an entire range of linguistic and even more general notions such as world knowledge, not just the dictionary of words and morphology. I will detail these later in this chapter.

An idealistic goal of the rule-based language technology is to encode all information in the words and their relations in texts in such a way that the computer can deduce the exact interpretations and make deductions on them. Of course, the main counterargument to this is that languages are ambiguous and encode information only partially, leaving much information to be discovered by using world knowledge or contextual evidence. Therefore, a perfect rule-based system is not always perfect language technology software (but can often produce perfect listing of all possibilities that are plausible).

There are numerous approaches, software, toolboxes and theories to rule-based language technologies, and this book is naturally only going to cover a subset of them. Specifically, one big branch of rule-based technology that is heavily featured in this book is computational morphology, that originated in the context of European and Uralic languages; the approach has been successfully used for languages of most typological branches thereafter and includes tools and technologies that go well beyond morphology and into syntax and semantics of language as well. Certain linguistic aspects and concepts are common to virtually all rule-based language technologies, and I will try to cover them in the rest of this introduction in a generic manner.

## 2 History and current trends into 2020s

In the early days of language technology, a rule-based approach was the norm. It was not until the 1990's when an alternative to rule-based approaches started to gain traction in the language-technology communities to such an extent that the term rule-based language technology became a common term. The alternative that came about was that of *Statistical Language Technology.* In statistically based approach of language technology, less emphasis is given on the grammar and linguistics, and the basis of computational modelling of the language lies in feeding computer enormous quantities of texts (or speech) and using mathematical algorithms to *learn* certain characteristics of the language. From this point of view the two approaches to language technology have also been called *expert-driven,* referring to the rule-based language technology, and *data-driven,* referring to the statistical language technology. From the data-driven language technologies, a branch was popularized in the 2000s, based on *neural network* technology used in artificial intelligence research, which is at the peak of its popularity now in 2022 as we are writing this book. The neural networks are in fact so overwhelmingly popular that rule-based approach at the time of writing almost has a reputation of being outdated and useless. It is one of our aims of this book to show this preconception as false.

Many researchers see these approaches to language technology as mutually exclusive and competing. However, this is not at all necessary and there are several approaches that make use of rule-based components and stochastic methods, forming a kind of technology often referred to as *hybrid language technology.*

The fundamental issue in the current trend of forgetting linguistics and pursuing statistical language technology models is that it is based on research and experimentation with only such languages that have a very long written history with practically endless amounts of written texts that machine can learn from. This is not the case for the majority of world's languages; according to the *Ethnologue (2022),* a language catalogue published by SIL international, there are 7151 living languages in the world in 2022. It is quite unlikely that more than top 100 of these have available linguistic resources for the data-driven learning of the language[1]. When discussing *lesser resourced* and *minority languages,* one must also keep in mind that the situations of text corpora can be very different to the majority languages. There may not be stable standard for written language, or one is very new or is controversial, and the majority of the text data follows different standards. It is vastly more difficult for a machine to learn to understand and produce texts following the standard if it is not represented in the data that is available. With rule-based approaches, the expert constructing the language technology can describe and prescribe the language norms as needed. For this reason, a rule-based approach can be particularly useful for language revitalization and for support of endangered languages.

[1] It is noteworthy, that several of these languages are oral and not written, and many more have only oral tradition with no pre-existing writing system. There exist language technologies for oral languages that are beyond the scope of this book; for mainly spoken languages some are aspiring to have written standard and there are also contents on spoken language processing in this book while that is not the main component of this book.

## 3 Linguistic concepts

Rule-based language technology draws its information from linguistic grammar and data, and much of this data has commonalities across theories and frameworks as well as software toolboxes. One of the most used sources of linguistic information in language technologies is *a dictionary.* All language technology performing rule-based processing of texts (and many working on speech signals as well) need to know information about words and word-forms. The scientific fields pertaining to modelling and understanding of words and word-forms span from lexicography to morphology. The task of modelling the words and their forms might sound like a simple concept, but there is a lot of detail that goes into modelling the lexicon of the language and its computational presentation. One of the most obvious linguistic observations here is that languages do differ in their morphology: for any English noun in English dictionary there exists at most four different word-forms of that word: e.g 'cat,' 'cats,' 'cat's,' and 'cats''. If you are modelling a Finnish lexicon computationally, a word exists in maybe 30 different forms if you look at Wictionary[2], or several thousand forms if you ask Finnish linguists[3]. In any case, modelling these as abstractions that will be useful in generic language technology tools requires much care and understanding of linguistic diversity. Initially one may engineer a system that works with one language only. If a tool is useful, it will eventually be tried with other languages or with machine translation in mind, and at that level any language-specific abstraction will become more apparent.

The other big component of rule-based language modelling is syntax of the language, what is often colloquially called just grammar, as in the context of *grammar checking and correction.* The syntax in this context refers to the rules governing how the words are arranged in a sentence. In many rule-based language technology systems, syntax is used more cautiously and sparingly than morphology, in that many language technology applications can be built with limited modelling of syntax. The syntax is so to say one level of abstraction higher than morphology and lexicon.

Going to further levels of abstraction in language modelling, language technology also uses *semantic* modelling, where also the semantics of words are further abstracted. In machine translation, the abstraction often goes even further up to have a concept of *Interlingua,* a completely language independent representation of the language that can be used to translate between languages or in general to have fully language independent abstract representation of meaning.

There are rule-based language technology approaches using all levels of abstraction and some based on mainly morphology or morphology with little help of syntactic processing. As a comparison one can say that a lot of statistical language technology tries to bypass all this abstraction by relying on just statistics of data, for example distributions of word-forms.

---

[2] https://en.wiktionary.org/wiki/talo#Finnish
[3] https://flammie.github.io/omorfi/genkau3.html

## 4 Language technology software

Language technology gets used in a multitude of end-user software, spanning from research tools to software aimed at big audiences. The research software that is built from language technology is used in linguistic research, but also in other near fields such as literature research. The uses of language technology in end-user software for big audiences are many. Language technology is needed in a high number of everyday applications, such as: spell-checking and correction, text input, search engines, grammar checkers, conversational agents (chatbots), recommendation systems (users who liked this will also like that), automatic subtitles, text-to-speech, machine translation, and summarisation. Practically any application that needs to process language has some language technology component in it.

It is also in the context of software that is based on language technology that one can easily appreciate the differences between the rule-based language technology and the data-driven approach. With rule-based approach, one is in control of every phase of the language processing in a very detailed and specific manner; if spell-checking application underlines a wrong word, we can find out if it is missing in our dictionary component. If a data-driven spell-checker underlines the wrong word, we only have an intuition that maybe it is under-represented in the texts we used to train the model. Similarly, when a rule-based machine translation gives out wrong translation, the expert who made it can often just look at the result to see where his rules, i.e., his mental modelling of the language grammars went wrong, whereas the same problem with data-driven methods will only point to lack of data with the only solution being that we should feed the machine more data and it may or may not get better.

Because I work with minority and under-resourced languages, another point of view I often like to bring out, is the risk and accountability of the language technology software you make. As an example, if you bring a spell-checker that underlines many correct words in red squiggly lines, to a community of language users, who are not confident in the use of their language and the current norms, it can be destructive for the whole culture. In rule-based approach we have a control of the software in the extent that we can limit the risk and take account of the errors of the software, whereas the statistical software can be of bad quality and there is no accountability since it was the data that was bad. On the other end for example for research software I have had experience with many researchers that a language technology software that makes "mistakes" of giving plausible but unlikely ideas of sentences is interesting because it is part of the research.

Ultimately, the choice here, like I have alluded to before, is not necessarily binary and one can incorporate various levels of statistical information to rule-based language technology to make a hybrid model, depending on the needs of the application. There is a popular saying in our sub-field of rule-based language technology "don't guess if you know," (e.g., Tapanainen and Voutilainen, 1994) which also encompasses well the most

common approach I have to hybridisation in this sense and I believe is a general approach within the community.

## 5 Summary

Rule-based language technology is the form of natural language software engineering based on using linguistic information as the main guideline of language understanding and generation. Rule-based language technology's main benefits are that it gives full control of the language, this is opposed to the other view on language technology that is based on statistical processing of the language, where the outcome is largely decided by the kinds of data you feed to it. Rule-based language technology is also not dependent on existing enormous quantities of well-written texts (or recorded speech) of the language. Furthermore, the approaches to language technology are not mutually exclusive, and can be combined for hybrid language technology.

## References

Eberhard, David M., Gary F. Simons, and Charles D. Fennig. 2022:
*Ethnologue: Languages of the world*. twenty-fifth edition.
http://www.ethnologue.com/.

Tapanainen, P., & Voutilainen, A. 1994:
Tagging accurately-Don't guess if you know. In *Fourth Conference on Applied Natural Language Processing*.