Declarative Models for Hybrid Machine Translation

Proposal for VR "Projektbidrag", Appendix A

Professor Aarne Ranta (PI)
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Purpose and aims

The project will develop theory and technology for precise and scalable machine translation. The models are based on grammatical structure, which is shared between languages on different levels of abstraction. Thus for instance the structure of passive constructions ("Guernica was painted by Picasso") is the same in many different languages, even though languages may realize it in different ways.

Describing which structures are available in which languages and how they are realized is the backbone of the proposed models. But to make this into an accurate language model, we also need to know the distribution of structures in different languages. For instance, even though passives are possible in Finnish, they are not very frequent, in particular with agents. Thus the normal translation of "Guernica was painted by Picasso" is "Guernican maalasi Picasso", which is an active construction with topicalized (fronted) object. One goal of the project is to make such structural changes follow from purely declarative language models, without hand-hacked transformation rules.

The models to be built are based on GF, Grammatical Framework, which is a functional programming language designed for implementing multilingual grammars. GF has an extensive grammar library, which currently covers 24 languages, largely using shared abstract structures such as "the passive construction". GF is the main technology of the on-going EU project MOLTO (Multilingual On-Line Translation), where it is used both by itself and in combination with statistical translation techniques, amounting to hybrid translation systems. The main goal of the current proposal is to strengthen the theoretical basis of such hybrid systems, maximally exploiting the multilingual abstractions available in GF and providing them with solid probabilistic interpretations.

Survey of the field

Machine Translation (MT) is one of the oldest tasks ever attempted with computers. The earliest attempts saw it as a cryptographic problem, similar to ones successfully solved during the Second World War. Thus, in many projects, Russian was seen as encrypted English, and the noisy channel method of Shannon (1948) was applied to decryption. None of the first attempts was successful, and

the famous ALPAC report (Carroll et al., 1966) cut most of the U.S. funding to MT. As a result, more basic research was done in Computational Linguistics (CL), which in particular approached MT with more sophisticated models based on linguistic knowledge, instead of cryptography based on unanalysed strings. However, the problem of automatic translation of natural languages remained unsolved, as the sophisticated linguistic systems were not able to scale up beyond small fragments of language.

In the late 1980's, new variants of Shannon's methods were introduced in a project at IBM, this time with clear advantages over linguistic methods. This initiated a new era of Statistical Machine Translation (SMT), of which Google's translation system is the most well-known current example. The success was largely due to two developments outside MT itself: first, the computation machinery was now able to deal with orders of magnitude more data; second, such data was now available electronically in quantities sufficient for building reliable statistical models for translation. Google translate, in particular, has had access to billions of words in hundreds of languages, and now provides translation for all pairs of languages in a set of over 50 languages.

Of course, also the mathematical foundations of SMT have developed during the last two decades, and the models are more sophisticated than in the beginning. However, it has also turned out that in many cases, further progress is not possible without the reintroduction of linguistic knowledge. **Sparse data** is the root of many of these problems. In raw text data, if no linguistic knowledge is applied, all inflection forms of a word must be treated as separate words. Thus English, which has something like 200,000 dictionary words, has almost a million distinct word forms. This means that more data is needed to cover all English words than would be needed if morphological analysis was available. The problem is amplified in SMT, because the models that are used need not just separate words but **n-grams**, that is, sequences of n words. SMT uses n-grams as a model for syntax - to specify which combinations of words are possible. The larger the *n* that is used, the better the model.

For English, which has relatively few inflection forms and vast amounts of text data available, good n-gram models are available for up to n=5 or even more. But most other languages are in a much worse situation. Often there is just a fraction of text available compared to English. At the same time, the number of words can be much higher. For instance, German verbs have 61 forms where English verbs have just 5. It is unlikely that all of the forms ever appear in any given data, but likely (by Zipf's law) that new data will expose some yet unseen forms. As a result, systems like Google translate often return German words without translation in the middle of English output.

Sparse data can be bad for word forms in many languages, but where it is really fatal is n-grams. In a language like German, the problem is made even more difficult because of variations in word order. Thus a given syntactic phrase, which in English usually appears as always the same n-gram of words, can in German be permuted in different orders. For instance, *the snow is white* is, in the main clause word order, *der Schnee ist weiss*. But there are two other common orders, shown in the conditional

wenn der Schnee weiss ist, ist der Schnee weiss ("if the snow is white, the snow is white"). In particular, words that belong together in a construction may end up arbitrarily far apart from each other, and hence they might not fit in a the same n-gram for any given *n*. A typical example in German are compound verbs such as *umbringen* ("to kill", *um* "around" + *bringen* "to bring"). This is why *er bringt seinen besten Freund um* ("he kills his best friend") gets translated *he brings his best friend to* in Google translate.

Today, more than 20 years after the new advent of statistical machine translation, the attention is again turning to linguistic knowledge; Church 2011 is an excellent summary of the issues. For instance, the sparse data problem with unknown words is easily overcome if morphological rules are applied to the data: what used to be 61 unrelated distinct words now becomes a pair of one word with 61 form descriptions, which in turn are shared with thousands of words. Similarly, the problem with word order can be solved by applying syntactic rules that apply to arbitrarily long sequences of words. Such approaches, combining SMT with linguistic knowledge, are known as **hybrid models of translation**.

The increasing popularity of hybrid models is easy to observe in the main conferences in the field. It is also witnessed by a recent textbook of SMT (Koehn 2010), which concludes with chapters on hybrid models as the future of the field. This creates a new demand for computationally precise and efficient linguistic models. Some such knowledge can be found in the "old" approaches, many dating back to the 1980's or even before. But the old knowledge-based systems of computational linguistics also have their problems: they are often complex and monolithic, and therefore not reusable in new constellations such as hybrid systems. It is also fair to say that CL research prior to the SMT era was not tuned towards highly multilingual systems. Much of the research concentrated on the large European languages, often just on English, working towards depth in semantics and Artificial Intelligence tasks rather than modular and scalable linguistic resources.

What is more, CL of the old school was notoriously protected by proprietary licenses, and large resources such as the morphological analysers of Xerox (Beesley and Karttunen 2003) and the Core Language Engine for syntax and semantics (Alshawi & al. 1993) were never made available for free use. One of the virtues of the current SMT approaches is its open-source practice, which has made the principal tools and data freely available for both research and industry, thereby boosting the development.

Our own main contribution in the field is GF, Grammatical Framework (Ranta 2011). It is a system partly in the spirit of the "old school", aiming at linguistically precise and deep descriptions. But at the same time, GF is a software engineering approach, attempting to make computational grammars scalable, efficient, and reusable. These two goals are in fact neatly unifiable, due to the theoretical basis of GF in type theory, functional programming, and compiler construction. Thus GF treats translation as a task similar to compilation, where an **abstract syntax** works as the hub between the source and target

languages. Based on this idea, GF implements the notion of **multilingual grammars**, where many languages share a common abstract syntax. The main illustration of this method is the **GF Resource Grammar Library** (RGL), which currently contains the main grammatical rules (morphology and syntax) of 24 languages, ranging from English and Swedish to Finnish, Nepali, and Thai. The RGL is an open-source project to which over 40 programmers around the world have contributed.

The main application of GF is currently the MOLTO project (Multilingual On-Line Translation), where GF is used for building multilingual translation systems for up to 15 simultaneous languages. These systems are mostly quality-oriented, in the sense that they have a limited coverage which is increased only when the semantics is well enough understood so that the translation is reliable. Thus one can say that the systems are restricted to what is known as **controlled language**. However, MOLTO also explores the possibility of hybrid models using combinations of GF and SMT. In this work, a novel view of hybrid models has emerged, which is the topic of the current proposal.

An important division in computational systems like MT is between **declarative** and **procedural** ones. Simply expressed, a declarative system is based on a theoretical description of its subject matter, and actual processing is derived by general principles. A typical example is **grammars**, which are declarative models of languages. In compilers, grammars are used for specifying a programming language on a high level, and the processing tasks of parsing and generating the language are derived from the grammar by general methods such as LR(k) parser generation. This is in contrast to a parser that is "hand-hacked" as a program that consists of explicit analysis rules for the language. Such hand-hacked parsers are avoided in modern compilers because they are messy, error-prone, and hard to maintain.

Also in natural language processing, grammars have the promise of providing declarative language models. Even though natural languages have more complex grammars than programming languages, there are general methods for parser generation; GF is one example of this (Ljunglöf 2004, Angelov 2012). However, in many *applications* of grammars, declarative models are just a small component, to which complex procedural rules are added. Translation has notoriously been such an area, so that statistical models have, perhaps surprisingly, had an advantage over traditional rule-based systems: an SMT system can be built on top of a declarative data model by using general algorithms for alignment and decoding. Standard software is available to support this, which is one reason why the systems are easy to replicate for new languages.

In GF, translation is purely declarative, since the grammars are by nature multilingual. What we want to add in the proposed project is a statistical component, which will also be declarative. Thus the notion of a "hybrid system", which may sound like a mix of heterogeneous things, will be defined in an orderly fashion as combining declarative multilingual grammars with declarative statistical models.

Project description

Translation in GF follows the phases of a modern compiler (Appel 1998):

1. Parsing: analyse the source language string *f* to get an abstract syntax tree *t0*.
2. Semantic analysis: enrich the tree *t0* with semantic information, obtaining *t1*.
3. Optimization: improve the tree *t1* with respect to the target language, obtaining *t2*.
4. Linearization: convert the tree *t2* into a target language string *e*.

In the simplest GF-based translation models, phases 2 and 3 are identities: in a multilingual grammar, the tree obtained by parsing the source can as such be linearized to the source language, and the result can be good enough. This corresponds to simple cases of compilation, such as converting infix expressions of Java (2 + 3 * 4) to postfix expressions of Java Virtual Machine (2 3 4 * +). But compilers typically need phase 2 as well, for instance, to decide whether "+" stands for the addition of integers, floating point numbers, or strings. Phase 3 includes non-compositional tree transformations such as peephole optimization and register allocation.

Most of the previous GF applications, including the MOLTO project, have followed the simplest compilation model, where phases 2 and 3 are trivial. This has worked surprisingly well in natural languages, which, due to the abstraction mechanisms of GF, are surprisingly similar on the level of abstract structure - more similar than different computer languages. However, the simple model requires there to be a uniform abstract syntax to represent meaning common to all involved languages. Such an abstract syntax can be built for specific domains - for instance, as in MOLTO, for mathematical exercises, museum object descriptions, and tourist phrasebooks. But if the domains are widened to cover, for instance, newspaper text, new problems arise.

One problem is **ambiguity**. It cannot be completely avoided even on small domains, as shown in the MOLTO phrasebook (Ranta & al. 2012). But it can be kept in reasonable limits, so that, in an interactive system, the user can easily choose the proper alternative. In grammars with wider coverage, however, ambiguities accumulate and can result in thousands of analyses for a relatively short sentence (Angelov 2012). Automatic disambiguation, analogous to the resolution of overloaded "+" in compilers, is therefore needed. For instance, when translating the English word *drug* to Swedish, a choice has to be made between medical and narcotic drugs, to choose the proper word.

Another problem is **fluency**. In small grammars, again, the rules can be carefully devised to generate good language from the same tree as obtained in parsing. The abstract syntax then expresses semantic concepts proper to the domain. For instance, in the museum object descriptions of MOLTO, the abstract syntax has the predicate *Painted(x,y),* which in English is expressed by a passive construction

(*x was painted by y*) but in Finnish by an active construction with fronted object (*y:n maalasi x*). In a wider context, such transformations accumulate, and they must be treated in a more global way. Reaching the best fluency in the target language is a problem very similar to optimization (phase 3 above) in compilers.

Both ambiguity and fluency are taken into account in statistical translation systems. This is a side effect of their holistic view of translation: they don't separate between different phases of analysis, but just select the most likely output for the given input according to the model constructed from previously seen data. For instance, "x + 3.14" could be interpreted as floating point addition just because the digram "3.14 fadd" is more frequent than "3.14 iadd" in a JVM code corpus. Clark and Curran (2007) actually show how a grammar-based parsing system can be dramatically improved by statistical disambiguation. Similarly, dysfluent sentences such as Finnish passive constructions with agents could be prevented by statistical models, because they don't appear in actual Finnish data.

Purely statistical disambiguation and optimization are, however, not reliable, because they don't guarantee the sameness of meaning. Thus, if we want to produce high quality in machine translation, we cannot solely rely on them. This is where the grammar-based semantical side of the system can help. The abstract syntax part of GF is a **logical framework**, which implements the **type theory** of Martin-Löf (1984). This means that the abstract syntax is capable of expressing any semantic content, both logical and computational. In particular, the abstract syntax of GF implements the notion of **definitional equality**, which is a decidable, computational model of the sameness of meaning. If we look back at the optimization phase (3) of compilation, we can add the condition that the optimized tree *t2* must be definitionally equal to the original tree *t1*. This is what of course must be satisfied by optimizations in compilers, as well.

To make the optimization both preserve meaning and improve fluency, it can be defined as search in the restricted space of trees definitionally equal to the original input. Within this space, the search can be performed in the same way as in statistical SMT: by maximizing the probability of the resulting tree. In fact, the model can be seen as an instance of the basic equation of SMT, where the translation $\hat{e}$ of an input f is defined

$$\hat{e} = argmax\ P(f|e)P(e)$$

In standard SMT, the probability $P(e)$ is estimated by putting together n-grams of words of the target language. In the hybrid GF-SMT approach, this is replaced by the probability of trees in the target language, rather than n-grams of words. This guarantees that the output is always grammatical, whatever the length of the sentence, permutations of words, and so on. The further constraint of definitional equality guarantees that the meaning is preserved.

It is also interesting to give an interpretation to the term *P(f|e)* in the equation: the probability that *f* is the source of *e*. Intuitively, this probability might have a maximum when *f* = *e*, and decrease when the distance from *f* to *e* increases. Keeping the trees as close as possible means that translation should change the structure as little as possible. Recall that, since the abstract tree structure in GF is shared, it is always meaningful to speak about the same tree in different languages. Minimization of tree distance means that the translation should preserve as much of the structure of the original as possible. Now, of course, we are looking at the probabilities *P(f|e)* and *P(e)* at the same time, so that the formula says, intuitively: find the best possible translation as close as possible to the original.

Even though the resulting optimization problem is hard (in general, undecidable), it gives some guidance to the search procedure finding the translation: starting from the input tree, one can generate definitionally equal trees in the order of increasing distance (which is easy), picking the one that is the most fluent in the target language and cutting the search when the increased distance makes fluency irrelevant.

Work plan

Developing a GF-based declarative model for hybrid translation can be divided to four tasks:
1. **Notions of tree probability**. The baseline is to use context-free probabilities, that is, the probability of a tree is the product of the probabilities of its nodes. These probabilities are easy to estimate from language data such as treebanks. But they do not cover the dependencies between different nodes, where a notion of **conditional probability** is needed. A promising direction for this is probabilistic models with **dependent types,** which GF inherits from type theory; in particular the branching topological models of type theory (Martin-Löf 1988). The problem is also related to **conditional random fields** (Lafferty & al. 2001).

2. **Language models**. A language model in the sense of GF is a grammar with associated tree probabilities. It is crucial to extend and improve the GF RGL, so that it can deal with language in the large. As a new task compared with previous development, tree probabilities will to be estimated from available data. One advantage of the shared tree structure in RGL is that models can be ported from one language to another. Whether this gives a good approximations of a "native" model is an interesting question that will be evaluated. Also the large scale definition of semantic equalities is a task that hasn't been properly addressed before; Ranta 1994 defines some basic ideas involved. Also **multilingual lexica**, based on resources such as linked WordNets, belong here.

3. **Software methods**. The actual translation with GF hybrid models needs algorithms and software that are efficient and scalable, and can be applied to any declarative model of right type. Both disambiguation and fluency optimization are potentially undecidable tasks, where good approximations

must be developed using techniques such as beam search.

4. **Case studies**. The main goal of the project is to create generic methods and tools. However, they must be constantly tested and demonstrated by real use cases. A natural candidate for such a use case is translation between English and Swedish, where GF already has comprehensive language models (Angelov 2012, Ahlberg 2012). At a later phase, we plan to add Finnish, for which similar resources are under construction; this will provide a case with a very different kind of a language, which is notoriously difficult for machine translation.

Timeline and human resources

This is the timeline for a four-year project. Frequent open-source releases of software and data are planned throughout the project, as well as publications in major conferences (such as ACL, EAMT, COLING) and journals (such as Machine Translation, Computational Linguistics).

Year 1. First case study: bidirectional translation between English and Swedish, and its comparison to other systems such as Google translate. In parallel, ground work on tree probabilities, software, and language models.
Year 2. Case study extended to Finnish. In parallel, ground work on probabilities, software, and language models.
Year 3. Large-scale work on language models, including semantics (definitional equalities) and methods for semantic lexicon acquisition.
Year 4. Case study extended to more languages depending on available resources from the open-source RGL project. Final evaluation and delivery of the software.

Aarne Ranta (PI) will contribute to the project with 20% of the budget. Krasimir Angelov is budgeted 75% as post-doc and later as assistant professor ("forskarassistent"). PhD student time is budgeted for Ramona Enache 50% for 18 months. Our home department provides relevant contacts in type theory (Thierry Coquand, Peter Dybjer, Bengt Nordström, Ulf Norell) and statistical modelling (Devdatt Dubhashi).

The project is a part of the activities of CLT, Centre for Language Technology, which is one of the five focus areas of research of the University of Gothenburg. CLT collaboration gives support to the project in the form of a research engineer (Thomas Hallgren), an assistant professor (Peter Ljunglöf), as well as cooperation with Språkbanken (Bank of Swedish) in particular on language resources and grammars (Lars Borin, Elisabet Engdahl).

The PI has an extensive international network, in particular within the MOLTO project which will run till

the end of May 2013. Collaborations relevant for the current proposal include work with Per Martin-Löf (University of Stockholm) on type theory and probabilistic models, and with Lluís Màrquez and Cristina España-Bonet (UPC, Barcelona Tech) on statistical and hybrid machine translation.

Significance

The project will create a new model for machine translation, which combines the advantages of statistical and grammar-based methods in a rigorous and declarative way. In particular, we expect the productivity and scalability of grammar-based methods to evolve into the same level as SMT methods. This will help improve the quality of machine translation, in particular for languages with complex grammars (e.g. Finnish), where pure SMT has known problems.

Preliminary results

In the MOLTO project, a hybrid GF-SMT system has been built for translating patents between English and French (Enache & al. 2012). This system shows that SMT can be improved by grammars. But it leaves room for further improvements. The study is just being extended to German, where we expect word order correction to show new improvements due to grammars.

GF-based analysis of open-domain text has been started for English by Angelov (2012), inspired by Clark and Curran (2007). Similar techniques are applied to Swedish by Ahlberg (2012).

References

Malin Ahlberg, *Towards a Wide-Coverage Grammar of Swedish Using GF*, MSc Thesis, University of Gothenburg, 2012.

Andrew Appel, *Modern Compiler Implementation in ML*, Cambridge University Press, 1998.

John R. Pierce, John B. Carroll, et al., *Language and Machines — Computers in Translation and Linguistics*. ALPAC report, National Academy of Sciences, National Research Council, Washington, DC, 1966.

Hiyan Alshawi, *The Core Language Engine*, MIT Press, 1992.

Krasimir Angelov, *The Mechanics of the Grammatical Framewor*k, PhD thesis, Chalmers University of Technology, 2012

Ken Beesley and Lauri Karttunen, *Finite State Morphology*, CSLI, Stanford, 2003.

Kenneth Church, A Pendulum Swung Too Far, *Linguistic Issues in Language Technology*, 2011.

Clark and Curran, Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models, *Computational Linguistics* 33(4):493–552, 2007.

Grégoire Détrez, Ramona Enache, and Aarne Ranta. Controlled Language for Everyday Use: the MOLTO Phrasebook. To appear in Fuchs & Rosner (eds), CNL 2010 proceedings, Springer LNCS/LNAI vol. 7175, 2012.

Ramona Enache, Cristina España-Bonet, Aarne Ranta, and Lluís Màrquez, 2012.

John Lafferty, Andrew McCallum, and Fernando Pereira, Conditional Random Fields: Probabilistic

Models for Segmenting and Labeling Sequence Data. *Proc. 18th International Conf. on Machine Learnin*g. Morgan Kaufmann. pp. 282–289, 2001.

Philipp Koehn, *Statistical Machine Translation*, Cambridge University Press, 2010.

Peter Ljunglöf, *The Expressivity and Complexity of Grammatical Framework*, PhD thesis, University of Gothenburg, 2004.

Per Martin-Löf, Mathematics of Infinit*y*, in: P. Martin-Löf and G.E. Mints (eds.) *COLOG-88 Computer Logic, Lecture Notes in Computer Science*, vol. 417, Springer, Berlin 1989.

Per Martin-Löf, *Intuitionistic Type Theor*y, Bibliopolis, Naples, 1984.

Aarne Ranta, *Type-Theoretical Grammar*, Oxford University Press, 1994.

Aarne Ranta, *Grammatical Framework: Programming with Multilingual Grammar*s, CSLI, Stanford, 2011.

Claude Shannon, A Mathematical Theory of Communication, *Bell System Technical Journal,* Vol. 27, pp. 379–423, 623–656 ,1948.