Using weighted finite state morphology with VISL CG-3—Some experiments with free open source Finnish resources [*]

Tommi A Pirinen
Ollscoil Chathair Bhaile Átha Cliath
CNGL—School of Computing
Dublin City University, Dublin 9
tommi.pirinen@computing.dcu.ie

June 18, 2015

Abstract

Traditionally, the coupling of finite state morphology and constraint grammar has been strictly rule-based, making binary distinctions between allowed and disallowed readings, however, in the recent years much of the research in the finite state morphologies has adapted the contemporary paradigm of statistically weighted analysis. This is reflected in current versions of free and open source morphology of Finnish, omorfi, in the finite state morphology part. In this paper we examine two strategies of making use of the weights as a part of VISL CG-3 pipeline. We evaluate the results intrinsically on small sample of analyses we have disambiguated by hand ourselves, and extrinsically on the effect it has on the rule-based machine translation of that text using the freely available open source translator, apertium-fin-eng.

1 Introduction

In the recent years, use of statistical information in computational linguistics has gained much interest, with systems like hunpos [1], moses [2] etc. being the main points of interest of most research in the field. In finite state morphology as well as constraint grammars, extensions to handle probabilities are recent and scarcely documented [3, 4]. In this paper we experiment with an existing weighted finite state morphology of Finnish [5][1] with VISL CG-3. For CG we have adapted Fred Karlsson's Finnish CG rules to omorfi's tag set, however, the rules were written for completely different analyser, which results in relatively low quality and high level of ambiguity at the current level. We estimate

[*] This article has a free/open publication licence; this particular version is Tommi A Pirinen's draft version. This version uses my own documentclass instead of official publication's, if any. This version is optimised for screen reading. This version uses only free fonts if possible. Published in NEJLT proceedings and ACL anthology of CG workshop in Nodalida 2015, Vilnius, Lithuania. http://www.aclweb.org/anthology/W/W15/#1800

[1] https://github.com/flammie/omorfi/

that salvaging these rules for the current version of analysis would require a substantial re-writing effort. In the meanwhile, there are a lot of easy targets that correctly trained statistical analyser can already deal with without extra effort. E.g., one large difference we assume between the analyser these CG rules were written for and omorfi's are that omorfi contains a huge number of proper nouns, dialectal and sub-standard forms, and rare language, animal etc. names, that are left ambiguous. It is obvious for a human reader that these words are very unlikely and given most corpora we expect them to be highly penalised as well.

The main goal of this experiment is to create a functional pipeline out of weighted finite-state analysis and current version of the constraint grammar. There are obvious conflicts between the statistically driven ranked hypotheses approach and the strictly deleting approach of the current constraint grammar, which may limit usefulness of our current method of combining these two information sources.

The rest of the paper is structured as follows: In section 2 we explain our starting point and current pipelines for morphological analysis, disambiguation and machine translation. In section 3 we explain various approaches we tried to include and combine weight data from the weighted finite-state analysers into VISL CG-3 and finally into machine translation. In section 4 we describe our experiments and how we measured the workability of our approach. In section 5 we show the results of the experiment. In section 6 we perform error analysis, compare our work with other existing approaches and lay out the future work. Finally in section 7 we summarise the conclusion of the experiments.

2 Background

Our starting point for this experiment is such that we had a modern, weighted finite-state morphology [6, ?] implementation of Finnish morphology in omorfi [?]. This morphology has rudimentary support for probabilistic weighting of surface forms or analyses using corpora-based unigram training approach. However, with the lack of high quality free and open source corpora compatible with omorfi analyses means that it is distributed with very basic linguist-written weights on the analysis side. For the main purpose of this experimentation we deemed this sufficient, to get the weights working through the pipeline at all.

On the other hand we had a free and open source, mature and large CG grammar by Fred Karlsson, that needed conversion to omorfi compatible tagging format, as well as some changes from CG-1 syntax to VISL CG-3.[2]

The fact that the CG rules from Karlsson were built using very different analyser than ours also played a large role in our decision to combine the weighted approach to with pure constraint grammar approach: the rule-writers of the original grammar had not seen large portion of the ambiguities introduced by larger, more varied lexicon of omorfi, including things like dialects, large inventories of proper nouns and unlikely but attested readings like plural cases of singular personal pronouns. For example, in the story we use for reference in our translation experiments, the sentence initial common words like “Mutta” (but)

[2] even though CG-1 and VISL CG-3 are possibly are mostly compatible, we found that some things may have started working better when changing to more conventional VISL CG-3 constructions

and “Koira” (dog) are also proper nouns, but also proper nouns like “Mari” have been added a common noun reading (slang for marihuana). Obviously these are not dealt with in the original ruleset as they have not appeared as ambiguities to the writers of th rules.

3 Methods

To first convert the original CG-1 ruleset to omorfi format analyses, we went through the rules by hand from beginning to end. This resulted in a ruleset where only a subset of rules matched to any constructs in the analysed texts. To further improve the quality and fix a lot of conversion errors we made use of the new VISL CG-3 features `no-inline-sets`. With help of this feature we got most of the ambiguous word-forms at least to match some of the rules, which hopefully means conversion has not too many tag formatting mismatch errors at the very least. The resulting ruleset with weight-based rule integrated is available from omorfi git repository.

To feed omorfi analyses into VISL CG-3 we have extended the python interface of omorfi to output CG stream format analyses, with omor style `[FEATURE=VALUE]` tags mapped into more conventional CG style tags, mostly of form `VALUE`. There are number of deviations to this of course, most notable being the `WEIGHT=` feature, which is turned into VISL CG-3 numeric tag. Other special conversions include things like usage, dialect and such lexical information, which are all included in angle bracket tags following VISL CG-3 conventions. Omorfi python interface also performs some case mangling (uppercasing, lowercasing, title-casing and removing title case) that seems to be similar as CG-1 rules seem to expect to appear in some angle-bracketed tags, so we have tried to map these to the readings in the original ruleset, with limited success.

The probabilities in omorfi are provided by the underlying HFST [7] system as a floating point number based on the finite-state implementation of a tropical semiring. This weight can be based on negative logarithms of probabilities of the word-forms, lemmas, analyses etc., as well as linguist-defined arbitrary values. For the purposes of this experiment we only used the linguist-defined values that are neatly in range of 0.1—32. This simplifies the scaling of the weights introduced by VISL CG-3 processing as we only have to scale against known range instead of e.g. combinations of negative logarithms’ maxima. As noted earlier in section 2, we use the default setting which is based on linguist-approximated tag likelihoods. Since VISL CG-3 does not support floating point numbers, e.g. 0.1, we output weight in a numeric tag multiplied by a 100 before rounding them down and turning into a tag of the form `<W=`*`weight`*`>`, where weight is the multiplied weight. This is sufficient for the coarse weights that default analyser produces, and in line what e.g. `cg-conv` does when it treats stream formats containing decimal data to be converted into numeric tags.

The basic support for numeric tag processign in VISL CG-3 is done by the `SELECT (<W=MIN>)` statement. If applied as a sole rule to result of omorfi to VISL CG-3 conversion it exactly like traditional weighted finite state morphology producing 1-best analysis. When combined into existing ruleset, we add this into a last, separate `SECTION`, in order to integrate some weight handling to CG iterations.

One long-term goal of this experimentation was to use VISL CG-3 also as a part of morphological analysis pipeline that produces n-best lists in same manner as weighted finite-state analyser does. To make this work, we take the output of VISL CG-3's cg-proc in trace mode before converting it back to an n-best list. There are multiple possible strategies to use readigns for deleted analyses as weights again. With this experiment, we have simply gone with adding the line number of the rule, this reflects the fact that later rules in the file are more risky and less ambiguous. Ideally however, we would like to annotate the rules using rule name labels, such as "usually", "dangerous" to denote e.g. multipliers for such rules. Furthermore, it is likely that it is not exactly the line number, but rather the section number, that is relevant for the rule likelihood, due to way linguists and rulewriters will organise rules within sections into blocks of related rules where ordering within and between blocks may not be important.

4 Experimental Setup

For analysis we use the python API to omorfi version 20150326, to turn the analyses into the format understood by VISL CG 3. We use a version Fred Karlsson's Finnish CG found in apertium's repository.[3], with the tag set manually converted to match omorfi's,[4] however, given the amount of ambiguous names of tags and sets and lists in the grammar, there may be some conversion errors left. The system is tested with VISL CG-3 version 0.9.9.10730, compiled from Gentoo packaging.[5]

To test the functionality of our combination of weighted finite-state analyser and VISL CG-3, we analyse a short text that we have manually disambiguated and measure the quality of analyses. The source of the text is found in the apertium's SVN repository.[6] For the purpose of this experiment, we have manually tokenised the text before processing it with omorfi. In addition to analysis we use the results of analyses in apertium's Finnish-English machine translator, and measure the translation quality. This way we can ensure that the gold annotation has not been selected to best fit our results but is actually the semantically most fitting one. The gold annotations can also be found in the omorfi git repository.

To perform evaluations we used simple python script that performs string comparisons of the gold file lines between the lines starting with `"<` ignoring empty lines and the ADDed CLB tags. The machine translation analysis was performed against current apertium-fin-eng ruleset and the reference translation in their svn, with standard machine translation metrics as measured by NIST's `mteval-13a.pl`, which is the standard BLEU metric of machine translation [?].

5 Evaluation

We first evaluated the analysers against the gold standard in table 1. We use simple metrics of Recall and Precision, defined as $\mathsf{Recall} = \frac{\text{Correct}}{\text{Gold}}$, where Cor-

[3] http://sourceforge.net/p/apertium/svn/HEAD/tree/nursery/apertium-fin-eng/apertium-fin-eng.fin-eng.rlx

[4] https://github.com/flammie/omorfi/tree/master/src/vislcg3

[5] https://github.com/flammie/flammie-overlay/tree/master/sci-misc/vislcg3

[6] http://sourceforge.net/p/apertium/svn/HEAD/tree/nursery/apertium-fin-eng/texts/tarina.fin.text

Rules	Precision	Recall
Weights	60	99
Rules	78	91
Combination	80	90

Table 1: Precision and recall of different combinations of weighted morphology and rules.

Rules	BLEU	PWER
Weights	6.86	87.11
Rules	5.76	88.67
Combination	5.60	88.89

Table 2: Precision and recall of different combinations of weighted morphology and rules.

rect is number of correct readings and Gold is number of gold readings, and $\mathsf{Precision} = \frac{\text{Correct}}{\text{All}}$, where All is number of all readings given by the disambiguation scheme. The row Weights stands for CG with only select weighted best applied, the row Rules stands for only converted CG ruleset applied, and row Combination uses both.

The precision with combination is as expected greater than converted rules, which in turn is greater than only weight-based rules that currently have not much large disambiguating effect. The recall conversely is largest as the weights let most readings through.

The resulting analyses is then converted to format expected by apertium for machine translation and evaluated for machine translation quality in table 2.

As can be seen from the scores the translator is still quite far from usable quality and thus comparison may not be very interesting. However we can see that the scores are systematically better with system's deemed worse by precision and better by recall.

6 Discussion

First of all, we note that the quality differences with adding weights has diminished from the version prior to conference and current version. This is largely due to newer version released in the workshop containing features that greatly improved the tag matching of the converted ruleset. Following this result we can say that the weights are most useful when the rules are not as high coverage, i.e. early stages of development or, as in this case, conversion process.

Nevertheless, the overall effect of combining weights has still improvements to exactly the shortcomings noted in the introduction as problems of the mismatching morphologies. In error evaluation, the cases that are affected by rules are mostly in derivation and productive compounding, but also some marginal cases that are not covered by rules.

For future work we are aiming to use the n-best list version of the result in a real-world application pipeline.

7 Conclusion

We have implemented a VISL CG-3 output on top of existing weighted finite-state analysis of Finnish language and tested that it works combined with VISL CG-3. We have successfully included this combination as a part of apertium machine translation pipeline. We note that weighted finite-state analysis can be easily combined with VISL CG 3 and results in an increased accuracy.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement PIAP-GA-2012-324414 (Abu-MaTran).

References

[1] Péter Halácsy, András Kornai, and Csaba Oravecz. Hunpos: an open source trigram tagger. In Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, pages 209–212. Association for Computational Linguistics, 2007.

[2] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pages 177–180. Association for Computational Linguistics, 2007.

[3] Krister Lindén and Tommi Pirinen. Weighting finite-state morphological analyzers using HFST tools. In Bruce Watson, Derrick Courie, Loek Cleophas, and Pierre Rautenbach, editors, FSMNLP 2009, July 2009.

[4] Eckhard Bick. Introducing probabilistic information in constraint grammar parsing. In Proceedings of Corpus Linguistics 2009, 2009.

[5] Tommi A Pirinen. Modularisation of Finnish finite-state language description—towards wide collaboration in open source development of morphological analyser. In Proceedings of Nodalida, volume 18 of NEALT proceedings, 2011.

[6] Kenneth R Beesley and Lauri Karttunen. Finite-state morphology: Xerox tools and techniques. CSLI, Stanford, 2003.

[7] Krister Lindén, Erik Axelson, Sam Hardwick, Tommi A Pirinen, and Miikka Silfverberg. Hfst—framework for compiling and applying morphologies. Systems and Frameworks for Computational Morphology, pages 67–85, 2011.