Squibs

Tommi A. Pirinen

Development and Use of Computational Morphology of Finnish in the Open Source and Open Science Era: Notes on Experiences with Omorfi Development

Abstract

This article describes a contemporary system for the computational modelling of the morphology of Finnish word-forms called Omorfi. The purpose of this article is to present new developments and an open development model of the morphological analysis of Finnish to the linguistic audience. The article shows Omorfi as a full-fledged, stable system for real-world usage in linguistic research and computational linguistics applications. Omorfi is free and open-source project, and crowd-sourcing and successful use of the community-driven development model is one of the key aspects of the system we want to present. We evaluated our analyser to give a rough idea of its usefulness and applications in linguistic work: around 95 % of the word-forms are known to the system and the analyses match the FinnTreeBank 3.1 standard analyses at 93 % faithfulness per token count.

1. Introduction

Computational morphological models and management of lexicographical data are a central component for most of the computational applications of linguistic analysis. Computational morphology of the Finnish language was

first described some 30 years ago (Koskenniemi 1983). The aim of this article is to present Omorfi[1] as a matured scientific project involving contributions from scientific community as well as crowd-sourced lexicographical additions, as a full-fledged project for managing lexicographical database on one hand and its natural language parser on the other hand. We will discuss our approach to lexicography and parser building in collaboration with crowds and experts. On technical side, we highlight some of the new features in the parser, especially from the point of view of linguists and end-users. The new features of the system at large that we bring to focus in this article consist of two items: the inter-operation of statistical and rule-based parsing methods and the open development model.

This article records a state of the state-of-the-art morphological analysis of Finnish. For a system overview in the article to be interesting and usable, we only highlight the long-term design goals of the system instead of transitional and volatile features of a fast-moving computer software that is developed by a base of open-source and language enthusiasts.[2]

The scientific advances within the development of the various features of Omorfi have been documented in scientific publications in various fora. The main advance to previous systems is the introduction of statistical language parsing component (cf. Manning & Schutze 1999), including its combination with a traditional rule-based model. The novelty in this article is not in singular experiments gone into Omorfi but a large-coverage system composed of all the state-of-the-art results in the field of computational morphology in weighted finite-state and relate technologies. This is, to our knowledge, one of the only on-going, mature, high-coverage statistical-rule based finite-state natural language parser, developed and used jointly by scientists, engineers and open source contributors via crowd-sourcing.

One notable practical distinction in our system is its licensing policy. Omorfi analyser is a free and open source product. In contemporary computational linguistics, freeness of systems and data is rightly seen as a cornerstone of properly conducted science, as it fulfils the requirement of repeatability by not setting unnecessary fences for the repetition of the

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

[2] For up-to-date documentation for implementation details and rapidly changing features, the project web site is the place to go: <https://github.com/flammie/omorfi/wiki>.

scientific results. There is a large base of recent research supporting this, specifically for Finnish the latest is by Koskenniemi (2008). For computer-literate end users this means that the tools necessary to perform linguistic analysis with Omorfi can be downloaded to and used on any average PC. There is an installation hosted and maintained by CSC – IT Center for Sciences[3] available for researchers.

2. Prior and related work

Omorfi is based on the tradition of finite-state morphologies, a theoretical framework laid out by Koskenniemi (1983). While our implementation is not directly related and it was written from the scratch, Omorfi was created in the context of University of Helsinki, parallel to a project to update, open-source and maintain the software necessary to build systems akin original two-level morphology (Lindén et al. 2011).[4] Omorfi roots are in a Master's thesis project (Pirinen 2008) based on the newly released open source word list from the Institute for the Languages of Finland at the time.[5] From a typical single-author project of that time, Omorfi has become a large coverage multi-author project with crowd-sourced lexical data sources.

Many of the scientific advances made by research groups in the Language technology department of the University of Helsinki have directly or indirectly affected Omorfi. The research on sub-word *n*-gram models (Lindén & Pirinen 2009a, 2009b) has been transferred to Omorfi compound disambiguation schemes. The methodology for semi-automatic lexical data harvesting, e.g. by Lindén (2008), has been largely influential on the gathering of the huge lexical database in Omorfi. Finally, the work on coupling statistical and rule-based approaches for disambiguation (Pirinen 2015), based on a grammar and a parsing approach by Karlsson et al. (1995), is included in the recent versions of Omorfi.

There have been competing and complementary approaches to computational parsing of Finnish. For example, in machine learning, Durrett and DeNero (2013)[6] show that unsupervised learning from Wiktionary data will create an analyser with recall in prediction of inflected

[3] <http://www.csc.fi/english/research/sciences/linguistics>
[4] <http://hfst.sf.net>
[5] Nykysuomen sanalista <http://kaino.kotus.fi/sanat/nykysuomi>
[6] We thank the anonymous reviewer for bringing this recent research to our attention.

word-forms in the ballpark of 83–87 %. However, their goal was to learn to predict Wiktionary's example inflection table's 28 forms per noun and 53 forms per verb, and they only performed intrinsic evaluation on held-out Wiktionary pages. Our approach to the usage of Wiktionary data is to collect the lexemes and their inflectional patterns already confirmed and written down by human language users[7], and use hand-written rules to inflect, which yields to a recall of virtually 100 % (bar bugs in our code) for the full paradigms. For this reason, it is hard to directly compare these two approaches. On the other hand, statistical language parsing systems have been built on top of Omorfi that go far and beyond the language parsing capabilities of a morphological parser, such as the *Universal dependency parser of Finnish* ("UD Finnish", Pyysalo et al. 2015).

One source of development in related works is the applications, Omorfi has been used in many real-world scientific applications to handle the Finnish language. For example spell-checking (Pirinen 2014), language generation (Toivanen et al. 2012), machine translation (Clifton & Sarkar 2011; Rubino et al. 2015), and statistical language modelling (Haverinen et al. 2013; Bohnet et al. 2013). On top of adding lexical data and statistical models, the vast array of applications has necessitated for Omorfi to take strong software engineering best common practices in use, in order to keep different end-applications usable. This is one of the key developments we wish to highlight in this article. The concept of continuous development by cooperation with computer scientists, linguists and common crowds via crowd-sourcing is as far as we know unique and under-documented for such a long-term free and open-source project as Omorfi is. The development by linguists and language technologists has been studied, e.g. by Maxwell (2008), and we have done our best to adapt and extend it to large open source development setting described in this article.

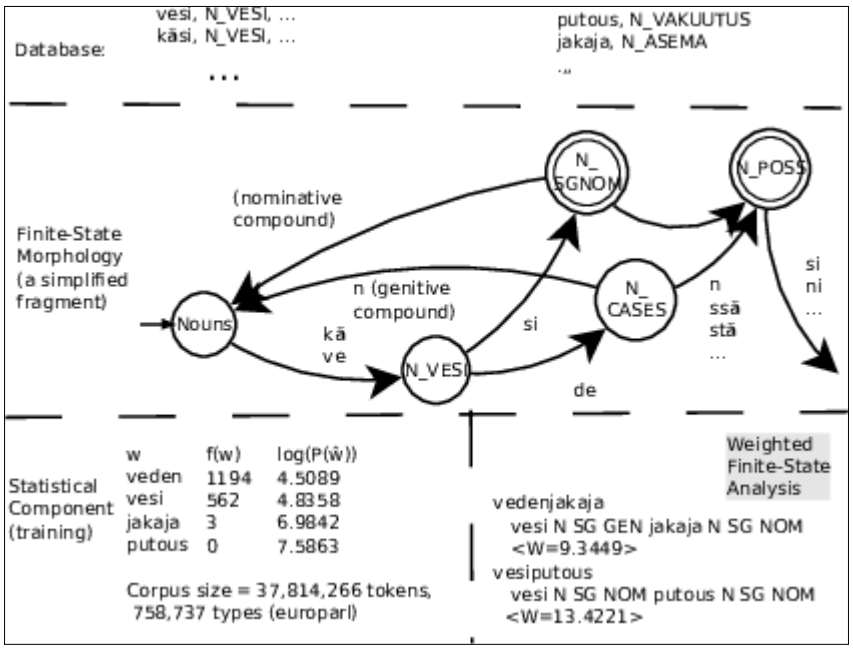
3. Methods

The implementation of our analyser follows the traditional works on *Finite State Morphology* by Beesley and Karttunen (2003). On top of that we have applied recent extensions from the research of finite-state morphology, such as weighted finite-state methods (Allauzen et al. 2008; Lindén et al. 2012). What this means in practice is basic unigram probabilities of word-

[7] In our opinion, trying to machine learn data, that is already available and verified by humans, is not largely useful.

forms composed[8] over the analyser from a corpus. Finally, probabilities are used in conjunction with constraint grammar rules (Karlsson et al. 1995) to disambiguate. This brings the traditional rule-based language analyser towards the statistical language analysers that are widely popular in the handling of morphologically less complex languages. A diagram of the combination is shown in Figure 1. The figure is a simplified version of the real implementation, just to show how few forms of select words interact in the system. The statistical component also omits the existence of known compounds to simplify the presentation. The flow of the system is the following: from database we generate a rule based analyser. The statistical data is counted from the corpora, and applied over the automaton using the formula by Lindén and Pirinen (2009a). The resulting automaton is used to analyse word-forms and the sentence context is used by constraint grammar to further select the best analyses.

Figure 1. Diagram of Omorfi technology showing a few example words (*vesi* 'water,' and *käsi* 'hand') and forms in the database, analyser and statistical training. Not shown in the automaton but used are also words *putous* 'fall' and *jakaja* 'divider' used to demonstrate compound formation and probablity calculations for *vesiputous* 'waterfall' and *vedenjakaja* 'watershed'. In finite-state representation, the double circle marks the end state, and the arrow leading away from the figure is cropped out of the example. The sub-strings in automaton drawing were compacted to single transitions where possible.



[8] Composition as defined in the standard finite-state algebra is well-defined for weighted automata.

The implementation of finite-state morphology in Omorfi is based on the arrangement of stems, stem variations and suffix morphs, without intermediate morphographemic processing. This relies on word classification to include data about stem patterns and vowel harmony for example. The classified dictionary words are stripped of their varying stem parts, and then concatenated with the variations and then stems, followed by all suffixes and optionally extended by compounding. This is done using standard finite-state morphology approach. E.g. in Figure 1, we have dictionary words *vesi* 'water' and *käsi* 'hand' with stem invariants *ve-* and *kä-* resp., and stem variation in *-si ~ -de- ~ ...*, and respectively suffixes 0 (nominative) ~ *-n* (genitive, 'water's') ~ *-ssä* (inessive, 'in water') ~ *-stä* (elative, 'from water') and so forth. This simple concatenation forms altogether some thousands of word-forms per dictionary word, as well as returns back to new words for compounding where applicable.

The baseline statistical methods for morphological models are applied over the finite-state formulation within the same framework, as is shown in the example in Figure 1. The formulation we use is the schoolbook unigram training (cf. Manning & Schutze 1999): get the likelihood $P(w)$ for the surface form w, by counting the amount of word-forms $f(w)$ in a corpus and divide it by the number of word-forms in the whole corpus CS: $P(w) = f(w)/CS$. To get around the problems with the probability of *0* for unseen word-forms, we use additive smoothing (Chen & Goodman 1999), which estimates frequency of each type as *1* larger than it is and the size of corpus as number of types larger $P(\hat{w}) = (f(w) + 1)/(CS + TC)$, where TC is a type count. The acquired likelihoods are combined to the finite-state morphological analyser by producing a weighted finite-state automaton for language model and composing it over the analyser to create a morphological analyser capable of producing both analyses and their likelihoods as shown in the last frame of Figure 1.[9] The probability-weighted analysis can be combined with rule-based probability-aware constraint grammars to produce robust disambiguating analysers (Pirinen 2015).

[9] The availability of accurate probabilistic data in the analyser is dependent on the acquisition of a suitable corpus, the default system builds "toy" weights based on linguistic insight.

4. Data

There are a few freely available open resources for lexicographical data of Finnish. The first one we used is based on lexicographical data of the dictionary from Institute for Languages of Finland, which has been available under free software licence GNU LGPL since 2007. The second source of lexical data we acquired from the internet is a free, open source database named *Joukahainen*[10]. For another source of lexical data we used the popular crowd-sourced *Wiktionary* project. We have used data from *FinnWordNet* (Lindén & Carlson 2010), as well as gathered data from students and various yet unpublished projects of University of Helsinki, and finally a number of contributors within project have added word-forms and attributes specifically for *Omorfi* using semi-automatic and manual approaches. The current dictionary includes 424,259 lexemes, classified in over 17 categories, including semantic features like biological gender, proper noun categories as well as morphosyntactic features like argument structures and defective paradigms.[11.]

5. Experimental set-up and evaluation

In this section we evaluate Omorfi to give an impression of its usefulness in various tasks and potential caveats when using for linguistic research. For evaluation we use only freely available corpora. The sizes of the corpora are detailed in Table 1. They include the following: ebooks of project Gutenberg[12], the data of Finnish Wikipedia[13], and the JRC Acquis corpus[14]. For downloading and pre-processing these corpora we use freely available scripts[15]. The scripts retain most of the punctuation and white-space as-is. The resulting token counts are given in Table 1. Some further tests were made with fully tokenised and analysed FinnTreeBank (Voutilainen et al. 2012) version 3.1. The scripts used for this evaluation are part of Omorfi source code and are usable for anyone.

---

[10] <http://joukahainen.puimula.org/>

[11] Figures change nearly weekly, up-to-date information is available on the project web site.

[12] <http://gutenberg.org>

[13] <http://fi.wikipedia.org>

[14] <http://ipsc.jrc.ec.europa.eu/index.php?id=198>

[15] <https://githb.com/flammie/bash-corpora/>

Table 1. Corpora used for evaluations. Tokens are all strings extracted from corpus and types are unique strings, both include punctuation and some codified expressions like URLs, addresses etc.

Corpus	Tokens	Types
Gutenberg	36,743,872	1,590,642
Wikipedia	55,435,341	3,223,985
JRC Acquis	42,265,615	1,425,532
FTB 3.1	76,369,439	1,648,420

First we measure the proportion of out-of-vocabulary items in the data. This gives us a naive coverage, formally defined as *Coverage = Analysed / Corpus size*. The results are presented in Table 2 for all the corpora we have.

Table 2. Naive coverages when analysing common corpora

Corpus	Gutenberg	Wiki	JRC Acquis	FTB 3.1
Coverage (tokens)	97.2 %	93.3 %	92.2 %	96.8 %
Coverage (types)	90.9 %	87.6 %	82.9 %	87.6 %

Faithfulness is measured as a proportion of equal analyses, formally *Faithfulness = Matched / (Correct + Missing)*. In Table 3 we show the results for the FTB3.1 corpus and analyses, first by proportion of all tokens in data then by unique tokens.

Table 3. The proportion of FTB3.1 analyses Omorfi can analyse with exact match in results.

Corpus	Faithfulness
FTB 3.1 (tokens)	93.3 %
FTB 3.1 (types)	77.0 %

The sizes and processing speeds for the automata built from the data described in section 4 using Debian packaged HFST software version 3.8.3[16]on a Dell XPS 13 laptop are given in Table 4. The speed was averaged over three runs using 1 million first tokens from Europarl.

Table 4. Size of Omorfi analyser as measured by `ls -lh`, speed of analysis using hfst-lookup in words per second averaged over three runs

Feature	Value
Size	22 megabytes
Speed	11,099 words per second

[16] <http://wiki.apertium.org/wiki/Prerequisites_for_Debian>

This result is in line with previous research on speed of optimised finite-state automata in natural language processing by Silfverberg and Lindén (2009).

6. Discussion and future work

We have presented a mature, jointly developed open source natural language analyser using both rule-based and statistical analysis approaches, and crowd-sourced lexicography development. The techniques of statistical language parsing in Omorfi are quite modest at modern standards. While the successful combination of statistical parsing and rule-based disambiguation is shown to be usable for a range of NLP applications, it would be interesting to see how the inclusion of more representative corpora applied with different methods would effect the parsing quality of Omorfi. In particular, it would be interesting to see an end-user application that would necessitate the use of high-quality disambiguated morphological analyses. We expect that the development towards universally recognised and comparable linguistic resources by projects like Universal dependencies will be crucial to the future development of Omorfi to the direction of state-of-the-art language processing.

One of the key components in the recent success of Omorfi is its adaptability and usefulness for various end uses. While it seems from the number of end users that it is in fact possible for independent researchers to use and develop Omorfi, it would be interesting to see more how linguists and lexicographers using Omorfi might improve the description as well as the end application quality.

6.1 Error Analysis

The coverage of the analyser is systematically around 98 %. This is virtually at the upper limits of reasonable results with the given corpora. This can be noticed by analysing the errors or the out-of-vocabulary word-forms left in the current corpora. For Wikipedia, we get codes, like *Lä*, *amp*, English, like *of*, *The*, and so forth. In the Gutenberg corpus, we get, among some missing proper nouns, archaic and dialectal forms like: *nämät* ‘these’, *kauvan* ‘long’, *sitte* ‘then’. While these can be added to the analyser quite easily, the examples will show what is known as *Zipfian distribution* of language data: rare word-forms and phenomena get exponentially rarer, thus the effect of collecting and classifying further lexemes will become insignificantly small (compare to Manning 2011). For

applications requiring higher, potentially 100 % coverage, using guessing techniques, e.g. Mikheev (1997), should be investigated.

The FTB3.1 evaluation (Table 3) is presented here as an example of customising Omorfi for an end user, and the faithfulness evaluations are based on comparison against an unknown closed source commercial tagger of FTB3.1. While we have mostly done our best to match the reference analyses, we have not degraded the analyser quality to match analyses what we view as bugs in the corpus. As an example of mismatched analyses right now: top wrong word-forms *oli* 'was', *olivat* 'were' are analysed as present tense in their annotations. We feel this is incorrect and does not warrant such analysis. In the near future we will use a free and open source, human-verified reference corpora instead, such as UD Finnish (Pyysalo, 2015), to gain stable high-quality analysis.

7. Conclusion

In this article we present a new fully open source Finnish morphological lexicon. We confirm that it is a full-fledged and mature lexical database that can be used as a baseline morphological analyser with large coverage, suitable for linguistic research, as well as in external applications such as spelling correction and machine translation. We have shown some approaches that make available use of modern natural language processing techniques like statistics in conjunction with analysers built from our data and paved a way forward for researchers interested in those topics. We also provide some easy-to-access ways for linguists and researchers to use and extend our database via publicly maintained servers and crowd-sourced web-based services.

References

Allauzen, Cyril; Riley, Michael; Schalkwyk, Johan; Skut, Wojchiech & Mohri, Mehryar (2007) Open-Fst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Twelfth International Conference on Implementation and Application of Automata,* (*CIAA 2007*), Lecture Notes in Computer Science 4783, pp. 11–23. Springer.

Beesley, Kenneth R. & Karttunen, Lauri (2003) *Finite State Morphology*. CSLI publications.

Bohnet, Bernd; Nivre, Joakim; Bouguavsky, Igor; Farkas, Richard; Ginter, Filip & Hajič, Jan (2013) Joint morphological and syntactic analysis for richly inflected languages. *Transactions of ACL* (1): 415–428.

Chen, Stanley F. & Goodman, Joshua (1999) An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13 (4): 359–393.

Clifton, Ann & Sarkar, Anoop (2011) Combining morpheme-based machine translation with post-processing morpheme prediction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Volume 1, HLT '11*, pages 32–42, Stroudsburg, PA: Association for Computational Linguistics.

Durrett, Greg & DeNero, John (2013) Supervised learning of complete morphological paradigms. In *Proceedings of the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL 2013*), pp. 1185–1195. Atlanta, GA: Association for Computational Linguistics.

Haverinen, Katri; Nyblom, Jenna; Viljanen, Timo; Laippala, Veronika; Kohonen, Samuel; Missilä, Anna; Ojala, Stiina; Salakoski, Tapio & Ginter, Filip (2013) Building the essential resources for Finnish: the Turku dependency treebank. *Language Resources and Evaluation* 47: 1–39.

Karlsson, Fred; Voutilainen, Atro; Heikkilä, Juha; & Anttila, Arto (1995) *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*, Natural Language Processing 4. Berlin/New York: De Gruyter Mouton.

Koskenniemi, Kimmo (1983) Two-level Morphology: A General Computational Model for Word-Form Recognition and Production. PhD thesis, University of Helsinki.

—— (2008) How to build an open source morphological parser now. In Joakim Nivre, Mats Dahllöf & Megyesi Beáta (eds.), *Resourceful Language Technology: Festschrift in Honor of Anna Sågvall Hein*, pp. 86–95. Uppsala: Acta Universitatis Upsaliensis.

Lindén, Krister (2008) A probabilistic model for guessing base forms of new words by analogy. In Alexander Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing*, pp. 106–116. Berlin/Heidelberg: Springer.

Lindén, Krister; Axelson, Erik; Drobac, Senka; Hardwick, Sam; Silfverberg, Miikka & Pirinen, Tommi A. (2012) Using HFST for creating computational linguistic applications. In Adam Przepiórkowski, Maciej Piasecki, Krzystof Jassem & Piotr Fuglewicz (eds.), *Proceedings of Computational Linguistics – Applications, 2012*, pp. 3–25. Springer.

Lindén, Krister & Carlson, Lauri (2010) Finnwordnet-wordnet på finska via översättning. *LexicoNordica* 17: 119–140.

Lindén, Krister & Pirinen, Tommi (2009a) Weighted finite-state morphological analysis of Finnish compounds. In Kristiina Jokinen & Eckhard Bick (eds.), *Nodalida 2009*, NEALT Proceedings Series, Volume 4, pp. 89–95. Tartu.

—— (2009b) Weighting finite-state morphological analyzers using HFST tools. In Bruce Watson, D. Courie, Loek Cleophas & P. Rautenbach (eds.), *FSMNLP 2009*. University of Pretoria.

Manning, Chris D. (2011) Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In Alexander Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing, 12th International Conference, CICLing 2011, Proceedings, Part I*. Lecture Notes in Computer Science 6608, pp. 171–189. Springer.

Manning, Chris D. & Schütze, Hinrich (1999) *Foundations of statistical natural language processing*. Cambridge: MIT press.

Maxwell, Mike & David, Anne (2008) Joint grammar development by linguists and computer scientists. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 27–34. Hyderabad: Asian Federation of Natural Language Processing.

Mikheev, Andrei (1997) Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23 (3): 405–423.

Pirinen, Tommi (2008) Suomen kielen äärellistilainen automaattinen morfologinen analyysi avoimen lähdekoodin menetelmin [Automatic Finite State Morphological Analysis of Finnish Language Using Open Source Resources]. Master's thesis. Helsingin yliopisto.

—— (2014) Weighted Finite-State Methods for Spell-Checking and Correction. PhD thesis. University of Helsinki.

—— (2015) Using weighted finite state morphology with visl cg-3—some experiments with free open source Finnish resources. In Eckhard Bick & Kristin Hagen (eds.), *Proceedings of CG Workshop in Nodalida 2015*, pp. 29–33. Linköping University Electronic Press.

Pyysalo, Sampo; Kanerva, Jenna; Missilä, Anna; Laippala, Veronika & Ginter, Filip (2015) Universal dependencies for Finnish. In *Nordic Conference of Computational Linguistics NODALIDA 2015*, pp. 163–172.

Rubino, Raphael; Pirinen, Tommi; Esplà-Gomis, Miquel; Ljubešić, Nikola; Ortiz-Rojas, Sergio; Papavassiliou, Vasilis; Prokopidis, Prokopis; & Toral, Antonio (2015) Abu-matran at wmt 2015 translation task: Morphological segmentation and web crawling. In *Proceedings of the Tenth Workshop on Statistical Machine Translation in EMNLP 2015*, pp. 184–191. <http://www.aclweb.org/anthology/sigmt.html#2015_1> (12 December 2015)

Silfverberg, Miikka & Lindén, Krister (2009) Hfst runtime format—a compacted transducer at allowing for fast lookup. In Bruce Watson, D. Courie, Loek Cleophas & P. Rautenbach (eds.), *FSMNLP 2009*. University of Pretoria.

Toivanen, Jukka; Toivonen, Hannu; Valitutti, Alessandro & Gross, Oskar (2012) Corpus-based generation of content and form in poetry. In Mary Lou Maher, Kristian Hammond, Alison Pease, Rafael Pérez y Pérez, Dan Ventura & Geraint Wiggins (eds.), *Proceedings of the Third International Conference on Computational Creativity*, pp. 175–179. Dublin.

Voutilainen, Atro; Muhonen, Kristiina; Purtonen, Tanja K.; Lindén, Krister (2012) Specifying treebanks, outsourcing parsebanks: Finntreebank 3. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of LREC 2012 8th ELRA Conference on Language Resources and Evaluation*, pp. 1927–1931. Istanbul.

Contact Information:

Tommi A Pirinen
Ollscoil Chathair Bhaile Átha Cliath
IE-D09 W6Y4
Éire
e-mail: Tommi(dot)Pirinen(at)computing(dot)dcu(dot)ie