Eric Y. Kow

Email: eric@erickow.com
Homepage: http://erickow.com

LANGUAGES

Programming: Haskell (preferred), Python (currently used), others (see below)

Natural languages: English (native), French (fluent)

PROJECTS

Université Paul Sabatier, Toulouse (Research Engineer, Feb 2013 to present)

(full-time from Dec 2014)

Development of Python libraries for working with discourse corpora such as the RST corpus

Infrastructure for corpus annotation in the STAC project

Rewriting a statistical discourse parsing system and developing test harness infrastructure

PAST PROJECTS

University of Brighton (Research Fellow, Feb 2014 to Nov 2014)

(half-time)

Prototyping for a visual ontology editor for WebProtégé (Java + GWT), based on a diagrammatic logical formalism developed by the Visual Modelling Group.

Packaging, optimisation, and test methodology for an information extraction system identifying named entities in historical documents from 1200-1700 (Prolog, DATR)

Well-Typed (March 2011 to July 2012)

(1 day a week)

Community building and publicity for the Parallel GHC project, an initiative funded by Microsoft Research to further the use of Parallel Haskell in the real world.

Computational Linguistics Ltd (May 2011 to July 2013)

(2 days a week)

Extensions of and optimisations of the surface realiser software I developed during my PhD (Haskell)

Integration of this software into an interactive teaching application (SRI)

University of Brighton (Research Fellow, 2007 to November 2011)

Research in applying probabilistic methods in deep natural language generation (Prodigy project) (Perl, Haskell)

Organisation of the 2008 Referring Expression Generation Challenge and the 2009-2010 Generation Challenges.

Université Henri Poincaré (PhD student, 2004 to 2007)

Surface realisation: ambiguity and determinism.
The surface realisation task consists in producing the natural language sentence(s) associated with an input grammar and meaning. This thesis presents three extensions to a surface realiser: a symbolic technique for filtering the lexical selection to reduce the effects of lexical ambiguity, a filter which enables the user to control the kind of output produced by the realiser and, an automated harness for using the realiser debug large grammars.

INRIA Lorraine (Engineer, 2001 to 2003)

Tools for facilitating software reuse through open standards such as SOAP. (Java)

Toolchain and support for the Dédé corpus annotation project.

PUBLICATIONS

Natural Language Generation for a Smart Biology Textbook. Eva Banik, Eric Kow, Vinay Chaudhri, Nikhil Dinesh, Umangi Oza. INLG, 2012

LG-Eval: A Toolkit for Creating Online Language Evaluation Experiments. Eric Kow and Anja Belz, LREC, 2012

Discrete vs. Continuous Rating Scales for Language Evaluation in NLP. Anja Belz and Eric Kow. ACL, 2011

Unsupervised Alignment of Comparable Data and Text Resources. Anja Belz and Eric Kow. Workshop on Building and Using Comparable Corpora (4th BUCC Workshop). 2011

Comparing Rating Scales and Preference Judgements in Language Evaluation. Anja Belz and Eric Kow. INLG, 2010.

Extracting Parallel Fragments from Comparable Corpora for Data-to-text Generation. Anja Belz and Eric Kow. INLG 2010.

System Building Cost vs. Output Quality in Data-To-Text Generation. Anja Belz, Eric Kow. ENLG, Athens, Greece, 2009. *Best paper award.*

A Symbolic Approach to Near-Deterministic Surface Realisation using Tree Adjoining Grammar. Claire Gardent and Eric Kow. ACL 2007.

Spotting Overgeneration Suspects. Claire Gardent and Eric Kow. ENLG 2007.

GenI: Natural language generation in Haskell. Eric Kow. Haskell'06.

Three reasons to adopt TAG-based surface realisation. Claire Gardent and Eric Kow. TAG+8, 2006.

Generating and selecting grammatical paraphrases. Claire Gardent and Eric Kow. ENLG, Aberdeen, UK, 2005

EDUCATION

PhD Computer Science (Université Henri Poincaré, Nancy, France, 2007-11-14)

Masters Computer Science (Université Henri Poincaré, 2004)

Bachelors Computer Science (University of Pennsylvania, 2001) [minor linguistics]

INTERESTS

Natural language generation

Functional programming

Revision control systems (Darcs)