Text Processing Tools and Services from iLexIR Ltd

Ted Briscoe, Paula Buttery, John Carroll
Ben Medlock, Rebecca Watson

iLexIR Ltd, Cambridge, UK
www.ilexir.com, ejb@ilexir.co.uk

Abstract

We describe the text processing tools that iLexIR has developed in collaboration with the Universities of Cambridge and Sussex. iLexIR has sole commercial rights to an extensive toolkit for English text processing applications and undertakes additional software development as well as tool tuning and porting for SMEs marketing applications with a text processing component.

To date, we have worked with clients to develop sentiment classification systems, mobile phone based question-answering services, and text mining tools for use in ESOL examination design and biomedical information extraction. Our toolkit has been extensively deployed for non-commercial research and proven its utility in projects on ontology and lexicon construction, anonymisation, anaphora resolution, word sense disambiguation, and many forms of text classification at the document, passage and sentence levels.

Index Terms: text analytics, text mining, text classification, question-answering, information extraction, ontology construction

1. Introduction

The RASP (robust accurate statistical parsing) toolkit is being developed by research groups based at the universities of Cambridge and Sussex (Briscoe & Carroll, 2002; Briscoe, Carroll & Watson, 2006). iLexIR was incorporated in 2003 as the sole commercial agent and owner of the intellectual property rights in RASP. We have deployed this toolkit, in conjunction with a range of open-source tools such as machine learning classifiers (e.g. Mallet, mallet.cs.umass.edu), information retrieval engines (e.g. Lucene, www.lucene.sourceforge.net) and XML-based document metadata handling systems (e.g. UIMA, uima-framework.sourceforge.net), to solve a diverse range of real-world text processing tasks. As a consequence of this activity, the RASP system is now also available embedded in UIMA (Andersen *et al.*, submitted; www.digitalpebble.com/resources.html), and iLexIR also offers tight integration of the toolkit with its own timed aggregate perceptron classifier, an innovative machine learning classifier with the accuracy comparable to support vector machines but training time closer to a naive bayes classifier (Medlock, forthcoming).

The resulting suite of tools, and expertise in using them, allows us to tackle almost any English text processing problem rapidly and effectively, yielding systems with state-of-the-art performance. In this paper, we describe the functionality of the toolkit and briefly discuss and reference some of the applications we have developed using the toolkit.

2. The RASP Toolkit

RASP is implemented as a series of modules written in C and Common Lisp, which are pipelined, working as a series of Unix-style filters. RASP runs on Unix-based platforms and is compatible with most C compilers and Common Lisp implementations. The public release includes Lisp and C executables for common 32- and 64-bit architectures, shell scripts for running and parameterising the system, documentation, and so forth. Potential commercial users may download the freely-distributed system under the non-commercial licence to conduct their own evaluation of its suitability for their application – see www.informatics.susx.ac.uk/research/nlp/rasp/ for licence and download details. An overview of the system is given in Figure 1.

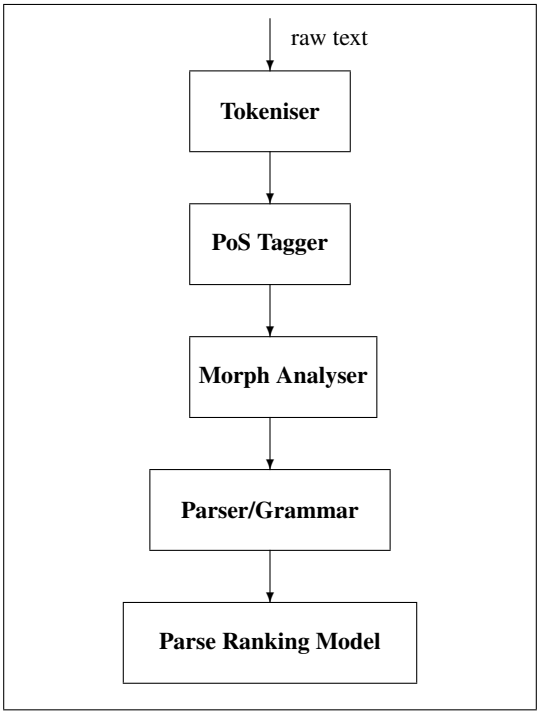

Figure 1: RASP Pipeline

2.1. Sentence Boundary Detection and Tokenisation

The system is designed to take unannotated text or transcribed (and punctuated) speech as input, and not simply to run on pre-tokenised input. Sentence boundary detection and tokenisation modules, implemented as a set of deterministic finite-state rules

in Flex (an open source re-implementation of the original Unix Lex utility) and compiled into C, convert raw ASCII (or Unicode in UTF-8) data into a sequence of sentences in which, for example punctuation tokens are separated from words by spaces, and so forth. Users are able to modify the rules used and recompile the modules. All RASP modules now accept XML mark up (with certain hard-coded assumptions) so that data can be pre-annotated – for example to identify named entities – before being passed to the tokeniser, allowing for more domain-dependent, potentially multiword tokenisation and classification prior to parsing if desired (e.g. Vlachos et al., 2006), as well as, for example, handling of text with sentence boundaries already determined, and retention of any document metadata encoded as XML.

2.2. PoS and Punctuation Tagging

The tokenised text is tagged with one of 150 part-of-speech (PoS) and punctuation labels (derived from the CLAWS tagset). This is done using a first-order ('bigram') hidden markov model (HMM) tagger implemented in C (Elworthy, 1994) and trained on the manually-corrected tagged versions of the Susanne, LOB and BNC corpora. The tagger has been augmented with an unknown word model which performs well under most circumstances as well as an extended lexicon better able to assign appropriate tags to rare words. The new tagger has an accuracy of just over 97% on the DepBank part of section 23 of the Wall Street Journal (WSJ), suggesting that this modification has resulted in competitive performance on text types at some remove from the original training data. The tagger implements the Forward-Backward algorithm as well as the Viterbi algorithm, so users can opt for tag thresholding rather than forced-choice tagging (giving $>$99% tag recall on DepBank, at some cost to overall system speed).

2.3. Morphological Analysis

The morphological analyser is also implemented in Flex, with about 1400 finite-state rules incorporating a great deal of lexically exceptional data. These rules are compiled into an efficient C program encoding a deterministic finite state transducer. The analyser takes a word form and CLAWS tag and returns a lemma plus any inflectional affixes. The type and token error rate of the current system is less than 0.07% (Minnen, Carroll and Pearce, 2001). The primary system internal value of morphological analysis is to enable later modules to use lexical information associated with lemmas, and to facilitate further acquisition of such information from lemmas in parses.

2.4. PoS and Punctuation Sequence Parsing

The manually-developed wide-coverage tag sequence grammar utilised in this version of the parser consists of around 700 unification-based phrase structure rules. The preterminals to this grammar are the PoS and punctuation tags. The terminals are featural descriptions of the preterminals, and the nonterminals project information up the tree using an X-bar scheme with 41 attributes with a maximum of 33 atomic values. The current version of the grammar finds at least one parse rooted in S for about 85% of the Susanne corpus (used for grammar development), and most of the remainder consists of phrasal fragments marked as independent text sentences in passages of dialogue. The coverage of our WSJ/DepBank test data is 84%. In cases where there is no parse rooted in S, the parser returns a connected sequence of partial parses covering the input. The criteria are partial parse probability and a preference for longer but non-lexical combinations (Kiefer *et al.*, 1999).

2.5. Probabilistic Generalised LR Parser

A non-deterministic LALR(1) table is constructed automatically from a CF 'backbone' compiled from the feature-based grammar. The parser builds a packed parse forest using this table to guide the actions it performs. Probabilities are associated with subanalyses in the forest via those associated with specific actions in cells of the LR table (Inui *et al.*, 1997). The n-best (i.e. most probable) parses can be efficiently extracted by unpacking subanalyses, following pointers to contained subanalyses, and choosing alternatives in order of probabilistic ranking. The probabilities of actions in the LR table are computed using bootstrapping methods which utilise an unlabelled bracketing of the Susanne Treebank (Watson *et al.*, 2007). This makes the system more easily retrainable after changes in the grammar and opens up the possibility of quicker tuning to in-domain data. In addition, the structural ranking induced by the parser can be reranked using (in-domain) lexical data which provides conditional probability distributions for the SUBCATegorisation attributes of the major lexical categories.

2.6. Grammatical Relations Output

The resulting set of ranked parses can be displayed, or passed on for further processing, in a variety of formats which retain varying degrees of information from the full derivations. The most common output format is a set of named grammatical relations (GRs), illustrated as a subsumption hierarchy in Figure 2. Factoring rooted, directed graphs of GRs into a set of bilexical dependencies makes it possible to compute the transderivational support for a particular relation and thus compute a weighting which takes account both of the probability of derivations yielding a specific relation and of the proportion of such derivations in the forest produced by the parser. A weighted set of GRs from the parse forest is computed efficiently using a variant of the inside-outside algorithm (Watson *et al.*, 2005).

2.7. Evaluation

The system has been evaluated using our reannotation of the PARC dependency bank (DepBank; King *et al.*, 2003) – consisting of 560 sentences chosen randomly from section 23 of the WSJ – with GRs compatible with our system. Relations take the following form: (**relation subtype head dependent initial**) where **relation** specifies the type of relationship between the **head** and **dependent**. The remaining **subtype** and **initial** slots encode additional specifications of the relation type for some relations and the initial or underlying logical relation of the grammatical subject in constructions such as passive. We determine for each sentence the relations in the test set which are correct at each level of the relational hierarchy. A relation is correct if the head and dependent slots are equal and if the other slots are equal (if specified). If a relation is incorrect at a given level in the hierarchy it may still match for a subsuming relation (if the remaining slots all match). Thus, the evaluation scheme calculates unlabelled dependency accuracy at the most general level in the hierarchy. The micro-averaged precision, recall and F_1 score are calculated from the counts for all relations in the hierarchy. Table 1 gives the microaveraged F_1 score for RASP, the Collins Model 2 parser, the Parc XLE parser, and the CCG parser. Only the CCG parser which is trained on in-domain data and includes many lexical parameters derived from

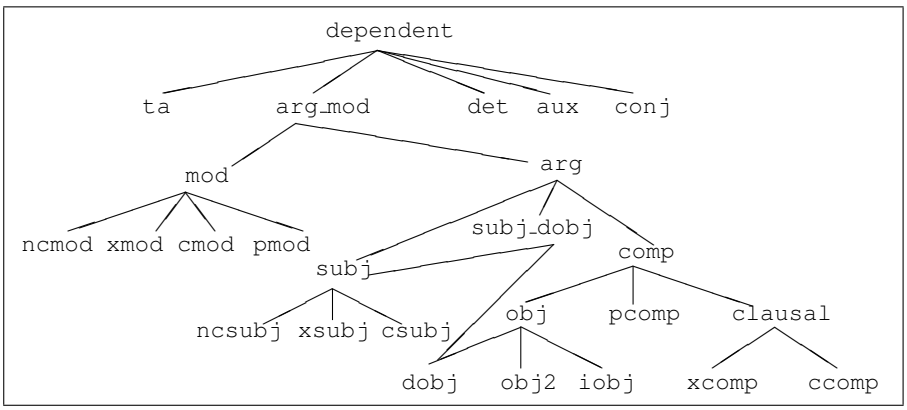

Figure 2: The GR hierarchy

the WSJ treebank outperforms the unlexicalised RASP parse ranking system (see Briscoe & Carroll, 2006 and Clark & Curran, 2007 for detailed discussion of the evaluation and results).

System	Precision	Recall	F_1
Collins	78.3	71.2	74.6
XLE	79.4	79.8	79.6
RASP	81.5	78.1	79.7
CCG	82.4	81.3	81.9

Table 1: Overall Microaveraged Scores

3. Text Classification with RASP

Standard text classification adopts the 'bag of words' (BoW) model in which a document is treated as an unstructured multiset of terms and information about word position or syntactic structure is ignored. This approach works well for document topic classification but less well for sentiment or genre classification, or for (sub)sentential classification tasks such as named entity recognition, anonymisation, or (non)-speculative assertion identification (e.g. Medlock, 2006).

The RASP toolkit makes available a range of features beyond BoW, based on morphological analysis (lemmas, stems), part of speech tags, and word cooccurrences mediated by grammatical relations rather than by adjacency or windowing. These additional feature types can be made available to machine learning classifiers, and feature instances from these types that are effective for a given classification task can be selected during the training phase by the classifier for run-time application.

The timed aggregate perceptron (TAP) classifier (Medlock, forthcoming) is a highly scalable linear classifier which has been shown to outperform SVMs and Bayesian logistic regression (BLR) on topic and other text classification tasks. The TAP classifier achieved better classification accuracy than either popular alternative, but trained in near linear time. This means that a classifier trained on the entire Reuters Rcv1 corpus of around 800K news stories (Lewis *et al.*, 2004) divided into 103 classes could be built in around 3.5hrs CPU time (as opposed to around 20hrs for the SVM or 50hrs for BLR). This is a significant advantage for real world applications where reductions in training time allow vital experimentation into enhancing feature generation and selection as well as frequent retraining as data is accumulated.

The TAP classifier has been tightly integrated with the RASP toolkit so that it is easy to undertake experiments to find the optimal set of feature types and instances for a particular classification task, whether this be at the document, passage, sentence or (sub)sentence level. However, in many real world applications it is not possible to train a classifier in a fully supervised fashion because data is only partially or noisily labelled. A significant element of the research undertaken with the toolkit has been to explore the use of bootstrapping and other semi-supervised techniques to circumvent the need for large quantities of well-annotated training data. In areas such as anonymisation (Medlock, 2006) and biomedical named entity recognition (Vlachos *et al.*, 2006) we have been successful in bootstrapping accurate classifiers from text automatically annotated with RASP.

4. Text Information Retrieval/Extraction with RASP

To date, RASP has been applied to around one billion words of English text drawn from genres as diverse as biomedical scientific papers through to second language learners' examination scripts. The additional annotations produced by RASP, possibly in conjunction with text classifiers, can form the basis for enhanced information retrieval at the document, passage or sentence level, based on going beyond keyword (BoW based) search for documents to search for named entitities in specific relations or contexts.

The fact that RASP and our text classifiers' produce XML annotations on text that may already be annotated with metadata allows us to efficiently exploit the new generation of XML aware open-source indexing engines such as Lucene, Indri (www.lemurproject.org/indri) and Xapian (www.xapian.org). These provide flexible search interfaces that allow Boolean combination of constraints based on XML path specifications and, thus, support seamless extension from information retrieval to information extraction, only limited by the extent of annotation in the indexed text. If a free text question interface is used, the approach can be extended to parsing the query, extracting the GRs in the query, and using the resulting annotation to find matches in the annotated document database. As an illustration of one application, Figure 3 shows a screenshot of the FlyBase curator interface to an article annotated with gene names.

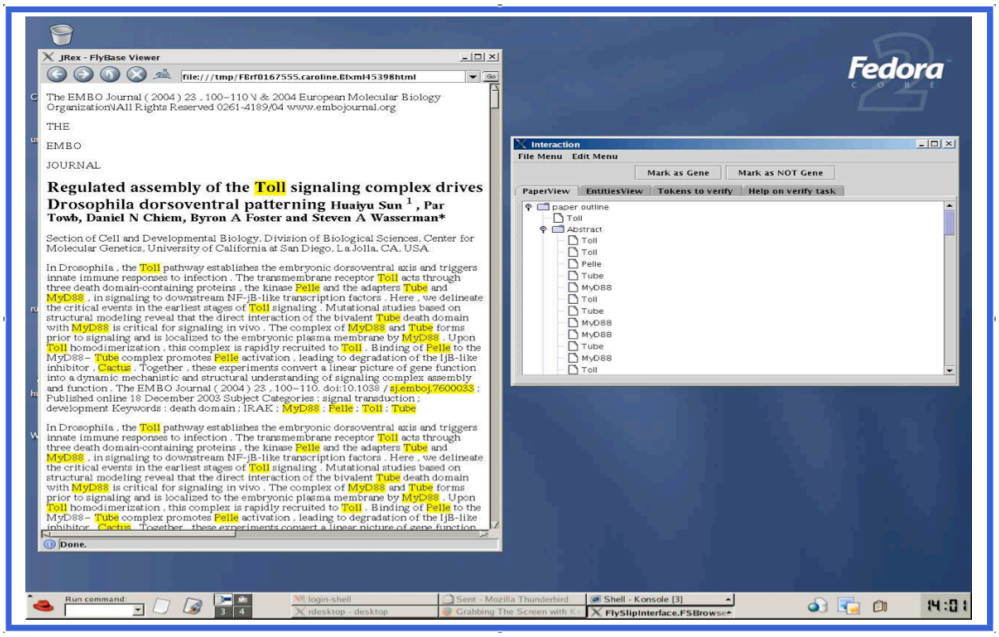

Figure 3: The FlyBase Curation Tool

5. Conclusions

Through integration of state-of-the-art tools from natural language processing, information retrieval and machine learning, we have been able to build a flexible toolkit with which it is possible to efficiently develop an optimal solution to most text processing tasks. The toolkit together with the know-how gained from research with its precursors has allowed us to rapidly develop components for commercial applications involving text mining, classification and question-answering. As a small research-led company, we expect to continue to develop the toolkit informed by the latest research in all three fields, whilst adding functionality in-house which will enhance robustness and scalability and decrease the resources required for tuning and adaptation to new applications.

6. References

Andersen, O., J. Nioche, E.J. Briscoe and J. Carroll (submitted) 'The BNC parsed with RASP4UIMA', *Proceedings of the 6th Int. Conf. on Language Resources and Evaluation (LREC),* Marrakech, Morocco.

Briscoe, E.J. and J. Carroll (2002) 'Robust accurate statistical annotation of general text', *Proceedings of the 3rd LREC,* Las Palmas, Gran Canaria, pp. 1499–1504.

Briscoe, E.J. and J. Carroll (2006) 'Evaluating the Accuracy of an Unlexicalized Statistical Parser on the PARC DepBank', *Proceedings of the 44th Assoc. Computational Linguistics(ACL)-Coling, Main Conf. Poster Sesssion,* Sydney, Australia, pp. 41–48.

Briscoe, E.J., J. Carroll and R. Watson (2006) 'The Second Release of the RASP System', *Proceedings of the 44th ACL-Coling, Interactive Presentation Session,* Sydney, Australia, pp. 77–80.

Clark, S. and J. Curran (2007) 'Formalism independent parser evaluation with CCG and DepBank', *Proceedings of the 45th ACL,* Prague, Czech Republic, pp. 248–255.

Elworthy, D. (1994) 'Does Baum-Welch re-estimation help taggers?', *Proceedings of the 4th ACL Conf. on Applied NLP,* Stuttgart, Germany, pp. 53–58.

Inui, K., V. Sornlertlamvanich, H. Tanaka and T. Tokunaga (1997) 'A new formalization of probabilistic GLR parsing', *Proceedings of the 5th Int. Workshop on Parsing Technologies (IWPT),* MIT, pp. 123–134.

Kiefer, B., H-U. Krieger, J. Carroll and R. Malouf (1999) 'A bag of useful techniques for efficient and robust parsing', *Proceedings of the 37th ACL,* University of Maryland, pp. 473–480.

King, T.H., R. Crouch, S. Riezler, M. Dalrymple and R. Kapl an (2003) 'The PARC700 Dependency Bank', *Proceedings of the 4th Int. Workshop on Linguistically Interpreted Corpora (LINC-03),* Budapest.

Lewis, D., Y. Yang, T. Rose, F. Li (2004) 'Rcv1: a new benchmark collection for text categorization research', *J. Machine Learning Research, vol.5,* 361–397.

Medlock, B. (2006) 'An introduction to NLP-based textual anonymisation', *Proceedings of the 5th LREC,* Genoa, Italy.

Medlock, B. (forthcoming) *Scalability for text categorization and the timed aggregate perceptron,* m.s..

Minnen, G., J. Carroll and D. Pearce (2001) 'Applied morphological processing of English', *Natural Language Engineering, vol.7.3,* 225–250.

Watson, R., J. Carroll and E.J. Briscoe (2005) 'Efficient extraction of grammatical relations', *Proceedings of the 9th IWPT,* Vancouver, Ca..

Watson, R., E.J. Briscoe and J. Carroll (2007) 'Semi-supervised Training of a Statistical Parser from Unlabeled Partially-bracketed Data', *Proceedings of the 10th IWPT,* Prague, Czech Republic.

Vlachos, A., Gasperin, C., Lewin, I., Briscoe, E. J. (2006) 'Bootstrapping the recognition and anaphoric linking of named entities in Drosophila articles', *Proceedings of the Pacific Symposium in Biocomputing,* Hawaii.