The BNC Parsed with RASP4UIMA

Ø. Andersen, J. Nioche, E.J. Briscoe and J. Carroll

University of Cambridge, Digital Pebble Ltd, University of Cambridge, University of Sussex
oeistein.andersen@cl.cam.ac.uk, julien@digitalpebble.com, ted.briscoe@cl.cam.ac.uk, j.a.carroll@sussex.ac.uk

Abstract

We have integrated the RASP system with the UIMA framework (RASP4UIMA) and used this to parse the XML-encoded version of the British National Corpus (BNC). All original annotation is preserved, and parsing information, mainly in the form of grammatical relations, is added in an XML format. A few specific adaptations of the system to give better results with the BNC are discussed briefly. The RASP4UIMA system is publicly available and can be used to parse other corpora or document collections, and the final parsed version of the BNC will be deposited with the Oxford Text Archive.

1. The British National Corpus

The British National Corpus (BNC), a 100-million-word balanced sample of written (90%) and spoken (10%) English produced in the UK in the period 1960–93, was first released in 1995 and has since seen a variety of uses in lexicography and linguistics, natural language processing and artificial intelligence. With the third edition (Burnard, 2007), the corpus moved from a custom SGML format to standard XML and Unicode, which makes it compatible with available tools and more readily exploitable.

2. Parsing and Metadata

The BNC contains word and sentence boundaries as well as part-of-speech tags (Leech, Garside & Bryant, 1994), but no parsing information and thus no facility to search for or otherwise make use of grammatical relations between words, which have proven useful in many applications. Various groups of people have parsed the corpus throughout the years using different tools and approaches. However, most, if not all, have simply removed all 'extraneous' mark-up from the corpus before parsing, which is not entirely satisfactory since we lose, *e.g.,* the distinction between titles and running text, formatting information, named entities and multi-word expressions, not to mention metadata including genre and provenance of texts and spoken data. (In addition, white-space modifications for tokenisation purposes will, if employed, cause further divergence from the original.) It seems to us that the only adequate solution is to keep the original mark-up intact and add new elements and attributes to indicate parsing information.

The RASP system (Briscoe, Carroll & Watson, 2006) is a domain-independent, robust parsing system for English which is free for research purposes. It was, in common with other extant publically-available parsers, designed for plain-text input and has only limited ability to handle XML-style mark-up natively. It would be possible, of course, to enhance RASP to handle arbitrary XML, but we chose instead the more flexible option of integrating its different parts into an existing analysis framework able to handle XML.

UIMA, the Unstructured Information Management Architecture (Ferrucci & Lally, 2004), originated at IBM Research from a need to process initially unstructured data, mainly natural-language documents, with a sequence of complementary tools. A well-defined architecture allows 'mixing and matching' of components without worrying about interfacing issues: each part adds new structured information in a way which makes it immediately available as input for the remainder of the processing chain. UIMA accepts modules written in Java and C++ and is currently being developed as a project in the Apache incubator (Apache, 2007).

3. RASP4UIMA

We have made UIMA interfaces to the five individual components of the RASP system under the name RASP4UIMA, the first version of which is publicly available (DigitalPebble, 2007). RASP's sentence splitter, tokeniser, part-of-speech tagger, lemmatiser and parser are hence available as separate *analysis engines* to all types of documents which can be handled within the UIMA framework. Fig. 1 provides a schematic overview of how each module contributes towards the final result.

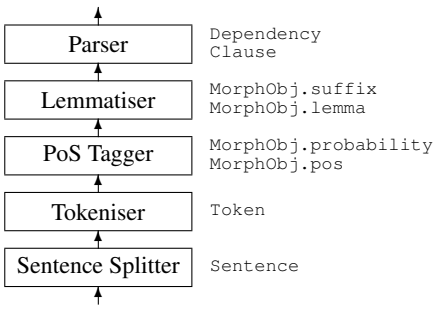

Figure 1: Analysis engines and annotations added.

Starting from unannotated text, the sentence splitter generates `Sentence` annotations; the tokeniser `Tokens`; the part-of-speech tagger `MorphObjs` (potentially more than one for ambiguous tokens) with `.pos` and `.probability` features, to which the lemmatiser adds `.lemma` and `.suffix`; and the parser `Dependency` relations and/or the full parse tree as `Clause` annotations. Each module uses annotations typically generated by the previous modules, but which may alternatively be obtained from elsewhere; *e.g.,* we used sentence boundaries and tokenisation information already present in the BNC.

We have also made an initial version of a *collection reader* and *consumer* (writer) to deal with the particulars of the BNC; the reader reads a document that adheres to the BNC-XML format, using the original mark-up to generate sentence and token annotations which can subsequently be used by the tagger, lemmatiser and parser (*see* Section 4 for details on BNC-specific details), whereas the consumer constructs a file containing all the original data as well as information obtained from processing.

As we shall see, multi-word expressions are currently parsed as individual words. RASP handles many such expressions internally, but alternative approaches might assign such expressions a single part-of-speech tag or propose this as one alternative during parsing (*see*, *e.g.,* Lewin, 2007 for experimental analysis of the utility of these different approaches). The RASP4UIMA framework is flexible enough to support these alternatives, though we have not implemented them in the first version.

RASP4UIMA actually contains a mechanism specifically designed to handle the case of mismatch between tokens and word units: part-of-speech tags are attached not directly to an atomic `token`, but instead to a higher-level `wordForm` which may correspond to any number of tokens. This representation of the linguistic information is inspired by the Morpho-syntactic Annotation Framework (MAF) described in Clément & de la Clergerie (2005) which deals with morpho-syntactic annotation of specific segments of textual documents. MAF currently has the status of an ISO draft international standard (ISO/CD 24611).

4. Processing of the BNC

The BNC contains mark-up identifying sentences and tokens quite accurately, so it seems resonable to take advantage of this information already present in the corpus rather than starting anew, which also alleviates the problem of how to combine pre-existing and additional annotations into an XML file at the end of the processing chain.

(1) `<trunc>`*Any*`</trunc>` *anyone who dissolved* `<mw>`*more than*`</mw>` ½ `<gap desc="formula"/>` *in rivers/lakes is n't gon na forget his pilgrimage , y'know .*

Example 1 shows an example sentence to which we are going to refer throughout this section. It has been artificially constructed from parts of actual sentences in the BNC to illustrate specific issues related to tokenisation, truncation, *etc*. The full XML representation can be found in Fig. 4; only the most essential mark-up is retained in Example 1, where tokens are indicated by separating white space instead of mark-up.

4.1. Collection Reader

Our BNC-specific collection reader uses a BNC-XML file as input to generate a UIMA representation of the textual content. Each sentence (`s` element) in the BNC results in a `Sentence` annotation, and words and punctuation marks (`w` and `c` elements, respectively) typically map to `Tokens`. No exception is made for multi-word expressions, whose constituents are handled as ordinary tokens. We have occasionally found it necessary to depart from the tokenisation in the BNC, however, as not doing so would cause obvious problems.

First, a few thousand `w` and `c` elements are empty. These spurious elements have been removed prior to parsing and are expected to be removed from a new official edition of the BNC as well.

Secondly, whereas most contracted forms like *isn't* and *cannot* have been split into two or several words as appropriate in the BNC, others like *let's* and *y'know* have not, nor have words separated by a solidus like *his/her*. In such cases, what is marked up as one word in the BNC has nevertheless been treated as two or more tokens by the parser, as happens for sequences *rivers/lakes* and *y'know* in the example.

Thirdly, in the BNC-XML, some parts of the original text or transcribed speech have been removed and replaced by a `gap` element, for reasons of anonymisation, inaudibility/illegibility, lack of appropriate textual representation, and so forth. In over 55% of the cases, a name, address or telephone number has been removed, whereas tables and figures, illustrations and photographs, foreign material and somewhat complicated formulæ account for another 35%. Such constituents typically play a syntactic rôle in the sentence and should not be ignored altogether; we have tagged all `gaps` as `&FO`, which effectively means that they will be handled as noun phrases by the parser.

Finally, parsing spoken data presents specific challenges. The tagger lexicon has been extended to cover interjections and contractions not typically found in written text, but several speech-specific issues remain unaddressed. One particular problem is related to truncated words and false starts: sometimes, the speaker stops in the middle of a word, changes his mind and says something else, often as a replacement for the word he was about to utter as well as previous ones. Somewhat simplistically, we ignore truncated words, *i.e.,* words inside `trunc` elements. This does not always work out quite as nicely as in the example, but attempting to tag truncated words is unlikely to work well, so this seems like a reasonable approach given that the mark-up does not really allow us to reconstruct the complete/corrected utterance with false starts removed or relegated to parentheticals.

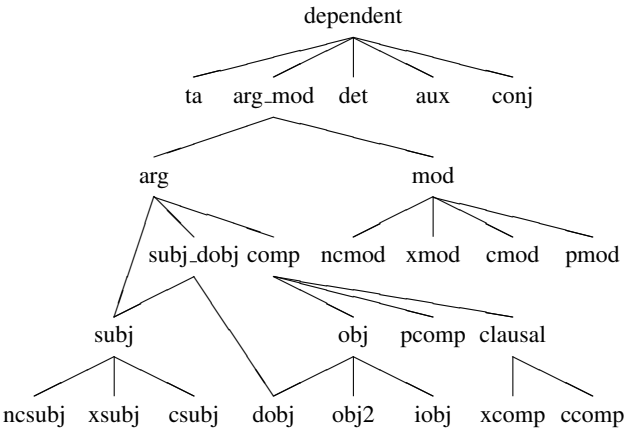

Figure 2: The GR hierarchy

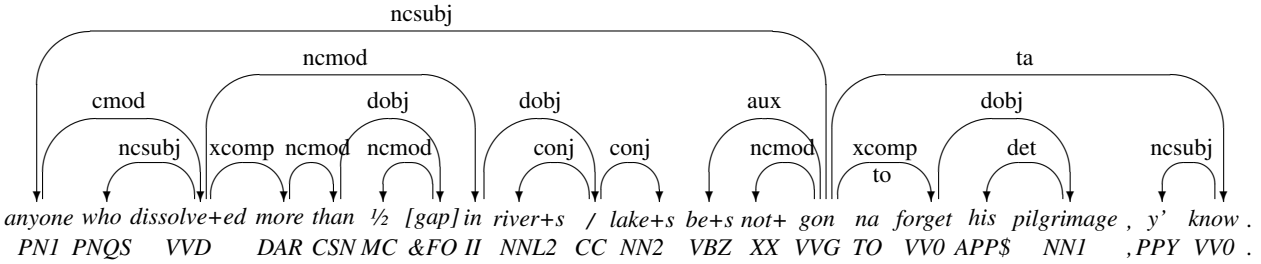

Figure 3: The tagger has added part-of-speech tags; the words have been lemmatised; and the parser has added dependency relations (GRs). The arrows representing GRs are drawn pointing from head to dependent; the label above the arrow indicates the type of relation, and the one below an optional subtype.

4.2. PoS Tagging, Lemmatisation and Parsing

Once the UIMA representation has been generated, the tagger, lemmatiser and parser can be evoked normally, using the generic analysis engines. The current version of RASP is fully UTF-8 compliant, which means that tokens can be passed on directly without worrying about non-ASCII punctuation marks, accented letters, *etc.* Example 2 shows the textual representation sent to the tagger.

(2) *anyone who dissolved more than ½ [gap] in rivers / lakes is n't gon na forget his pilgrimage , y' know .*

The information added to the sentence can be seen in Fig. 3. The parsing data is encoded not as trees, but as grammatical relations (GRs) between head and modifier. GRs result from transformation of a derivation tree constructed by the parser. The different relations are illustrated as a subsumption hierarchy in Fig. 2. They capture those aspects of predicate–argument structure that the system is able to recover and is the most stable and grammar-independent representation available. (*See* Briscoe, 2006 for a more detailed description of the GR scheme.)

As there is no annotated test data for the BNC, we do not know how accurate the RASP analyses are. However, as we use an unlexicalised model, we expect performance to be similar to that on other out-of-training-domain test data (*see* Briscoe & Carroll, 2006 for details).

4.3. Collection Consumer

A collection consumer written for the BNC uses the original BNC file as well as the newly generated annotations to create a new file containing the information from both sources as illustrated in Fig. 4 which shows how the example sentence would end up.

The words have been numbered (attribute `id`) and annotated with part-of-speech tags (`rpos`) and lemma/suffix (`lem` and `affix`). The BNC already contains part-of-speech tags from a slightly less detailed tagset (`c5`) and a coarse word-class category (`pos`) as well as lemmatised forms derived using slightly different rules (`hw`); these cannot be used directly for parsing with RASP, but are kept in the corpus and may be useful to measure, *e.g.,* tagger agreement.

5. Conclusion

We have presented RASP4UIMA, an integration of the RASP parser within the UIMA framework, which is particularly useful for parsing already annotated data and also makes it easier to combine RASP with other systems for analysis of textual data. We have used this system to parse the new XML version of the BNC and documented specific adaptations to the system that give better results than would otherwise have been obtained.

We have demonstrated that it is possible to parse a significant corpus containing rich metadata without loss of information, but at the same time to exploit and augment this metadata in a manner which supports optimal parsing results with a specific extant system. Furthermore, the approach we take is generic and should be reapplicable to any corpus with metadata and data encoded as XML.

This parsed version of the BNC will be available through the Oxford Text Archive for others to use, which not only provides the possibility to work with a parsed corpus without first parsing it oneself, but also avoids duplication of effort and makes it more likely that specific issues will be discovered and can be taken into account when the same or other large corpora are going to be parsed in the future.

Acknowledgements

This paper reports on research supported by the University of Cambridge ESOL Examinations.

6. References

The APACHE Software Foundation (2007): Apache UIMA. <http://incubator.apache.org/uima/>.

Edward J. BRISCOE. (2006): *An Introduction to Tag Sequence Grammars and the RASP System Parser*, University of Cambridge, Computer Laboratory Technical Report 662.

Edward J. BRISCOE and John CARROLL (2006): 'Evaluating the accuracy of an unlexicalized statistical parser on the PARC DepBank'. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney.

Edward J. BRISCOE, John CARROLL and Rebecca WATSON (2006): 'The Second Release of the RASP System'. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, Sydney.

Lou BURNARD (2007): *Reference guide for the British National Corpus (XML edition)*. <http://www.natcorp.ox.ac.uk/XMLedition/URG/>.

Lionel CLÉMENT and Éric VILLEMONTE DE LA CLERGERIE (2005): 'MAF: a Morphosyntactic Annotation

Framework'. In *Proceedings of the 2nd Language & Technology Conference (LT'05)* (pp. 90–94), Poznań.

DIGITALPEBBLE (2007): RASP4UIMA 1.0 beta. <http://www.digitalpebble.com/rasp4uima/>.

David FERRUCCI and Adam LALLY (2004): 'UIMA: an architectural approach to unstructured information processing in the corporate research environment'. In *Natural Language Engineering* Vol. 10, No 3/4 (pp. 327–348).

Geoffrey LEECH, Roger GARSIDE and Michael BRYANT (1994): 'CLAWS4: The tagging of the British National Corpus'. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94)* (pp. 622–628), Kyōto.

Ian LEWIN (2007) 'BaseNPs that contain gene names: domain specificity and genericity'. In *BioNLP 2007: Biological, translational, and clinical language processing* (pp. 163–170), Prague.

```
<s n="1">
  <trunc>
    <w c5="UNC" hw="any" pos="UNC">Any </w>
  </trunc>
  <w n="1" c5="PNI" hw="anyone" pos="PRON" rpos="PN1" lem="anyone">anyone </w>
  <w n="2" c5="PNQ" hw="who" pos="PRON" rpos="PNQS" lem="who">who </w>
  <w n="3" c5="AJ0-VVN" hw="dissolved" pos="ADJ"
     rpos="VVD" lem="dissolve" affix="+ed">dissolved </w>
  <mw c5="AV0">
    <w n="4" c5="AV0" hw="more" pos="ADV" rpos="DAR" lem="more">more </w>
    <w n="5" c5="CJS" hw="than" pos="CONJ" rpos="CSN" lem="than">than </w>
  </mw>
  <w n="6" c5="UNC" hw="" pos="UNC" rpos="MC" lem=""> </w>
  <gap n="7" desc="formula" rpos="amp;FO" lem="[gap]"/>
  <w n="8" c5="PRP" hw="in" pos="PREP" rpos="II" lem="in"> in </w>
  <w n="9 10 11" c5="NN1" hw="rivers/lakes" pos="SUBST"
     rpos="NNL2 CC NN2" lem="river / lake" affix="+s  +s">rivers/lakes </w>
  <w n="12" c5="VBZ" hw="be" pos="VERB" rpos="VBZ" lem="be" affix="+s">is</w>
  <w n="13" c5="XX0" hw="not" pos="ADV" rpos="XX" lem="not" affix="+">n't </w>
  <w n="14" c5="VVG" hw="gon" pos="VERB" rpos="VVN" lem="gon">gon</w>
  <w n="15" c5="TO0" hw="na" pos="PREP" rpos="TO" lem="na">na </w>
  <w n="16" c5="VVI" hw="forget" pos="VERB" rpos="VV0" lem="forget">forget </w>
  <w n="17" c5="DPS" hw="his/her" pos="PRON" rpos="APP$" lem="his">his </w>
  <w n="18" c5="NN1" hw="pilgrimage" pos="SUBST"
     rpos="NN1" lem="pilgrimage">pilgrimage</w>
  <c n="19" c5="PUN" rpos="," lem=",">, </c>
  <w n="20 21" c5="VVB-NN1" hw="y'know" pos="VERB"
     rpos="PPY VV0" lem="y' know">y'know</w>
  <c n="22" c5="PUN" rpos="." lem=".">.</c>
  <grlist parse="1" score="-40.848">
    <gr type="ncsubj" head="14" dep="1"/>
    <gr type="cmod" subtype="_" head="1" dep="3"/>
    <gr type="ncsubj" head="3" dep="2"/>
    <gr type="ncmod" subtype="_" head="3" dep="8"/>
    <gr type="xcomp" subtype="_" head="3" dep="4"/>
    <gr type="ncmod" subtype="_" head="4" dep="5"/>
    <gr type="dobj" head="5" dep="7"/>
    <gr type="ncmod" subtype="_" head="7" dep="6"/>
    <gr type="dobj" head="8" dep="10"/>
    <gr type="conj" head="10" dep="9"/>
    <gr type="conj" head="10" dep="11"/>
    <gr type="aux" head="14" dep="12"/>
    <gr type="ncmod" subtype="_" head="14" dep="13"/>
    <gr type="ta" subtype="end" head="14" dep="21"/>
    <gr type="xcomp" subtype="to" head="14" dep="16"/>
    <gr type="passive" head="14"/>
    <gr type="dobj" head="16" dep="18"/>
    <gr type="det" head="18" dep="17"/>
    <gr type="ncsubj" head="21" dep="20"/>
  </grlist>
</s>
```

Figure 4: The example sentence after parsing with the current version of RASP4UIMA. Elements and attributes in italics have been added; the rest is taken directly from the BNC. Note that the attributes `n`, `rpos`, `lem` and `affix` are space-separated lists when tokens have been split.