Tools to Address the Interdependence between Tokenisation and Standoff Annotation

Claire Grover and **Michael Matthews** and **Richard Tobin**
School of Informatics
University of Edinburgh
{C.Grover, M.Matthews, R.Tobin}@ed.ac.uk

Abstract

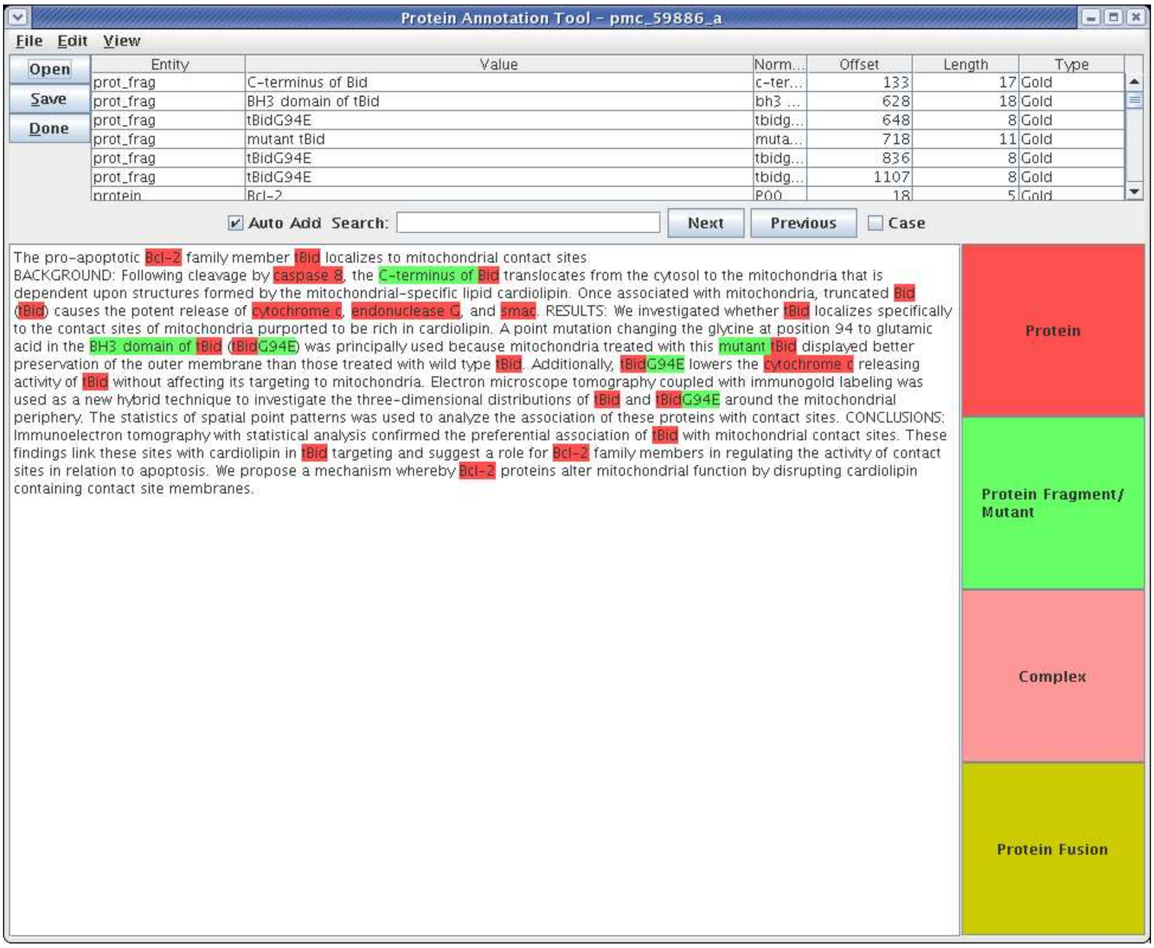
In this paper we discuss technical issues arising from the interdependence between tokenisation and XML-based annotation tools, in particular those which use standoff annotation in the form of pointers to word tokens. It is common practice for an XML-based annotation tool to use word tokens as the target units for annotating such things as named entities because it provides appropriate units for stand-off annotation. Furthermore, these units can be easily selected, swept out or snapped to by the annotators and certain classes of annotation mistakes can be prevented by building a tool that does not permit selection of a substring which does not entirely span one or more XML elements. There is a downside to this method of annotation, however, in that it assumes that for any given data set, in whatever domain, the optimal tokenisation is known before any annotation is performed. If mistakes are made in the initial tokenisation and the word boundaries conflict with the annotators' desired actions, then either the annotation is inaccurate or expensive retokenisation and reannotation will be required. Here we describe the methods we have developed to address this problem. We also describe experiments which explore the effects of different granularities of tokenisation on NER tagger performance.

1 Introduction

A primary consideration when designing an annotation tool for annotation tasks such as Named Entity (NE) annotation is to provide an interface that makes it easy for the annotator to select contiguous stretches of text for labelling (Carletta et al., 2003; Carletta et al., in press). This can be accomplished by enabling actions such as click and snapping to the ends of word tokens. Not only do such features make the task easier for annotators, they also help to reduce certain kinds of annotator error which can occur with interfaces which require the annotator to sweep out an area of text: without the safeguard of requiring annotations to span entire tokens, it is easy to sweep too little or too much text and create an annotation which takes in too few or too many characters. Thus the tokenisation of the text should be such that it achieves an optimal balance between increasing annotation speed and reducing annotation error rate. In Section 2 we describe a recently implemented XML-based annotation tool which we have used to create an NE-annotated corpus in the biomedical domain. This tool uses standoff annotation in a similar way to the NXT annotation tool (Carletta et al., 2003; Carletta et al., in press), though the annotations are recorded in the same file, rather than in a separate file.

To perform annotation with this tool, it is necessary to first tokenise the text and identify sentence and word tokens. We have found however that conflicts can arise between the segmentation that the tokeniser creates and the segmentation that the annotator needs, especially in scientific text where many details of correct tokenisation are not apparent in advance to a non-expert in the domain. We discuss this problem in Section 3 and illustrate it with examples from two domains, biomedicine and astrophysics.

In order to meet requirements from both the annotation tool and the tokenisation needs of the annotators, we have extended our tool to allow



Entity	Value	Norm...	Offset	Length	Type
prot_frag	C-terminus of Bid	c-ter...	133	17	Gold
prot_frag	BH3 domain of tBid	bh3 ...	628	18	Gold
prot_frag	tBidG94E	tbidg...	648	8	Gold
prot_frag	mutant tBid	muta...	718	11	Gold
prot_frag	tBidG94E	tbidg...	836	8	Gold
prot_frag	tBidG94E	tbidg...	1107	8	Gold
protein	Bcl-2	P00	18	5	Gold

☑ Auto Add Search: ☐ Case

The pro-apoptotic Bcl-2 family member tBid localizes to mitochondrial contact sites
BACKGROUND: Following cleavage by caspase 8, the C-terminus of Bid translocates from the cytosol to the mitochondria that is dependent upon structures formed by the mitochondrial-specific lipid cardiolipin. Once associated with mitochondria, truncated Bid (tBid) causes the potent release of cytochrome c, endonuclease G, and smac. RESULTS: We investigated whether tBid localizes specifically to the contact sites of mitochondria purported to be rich in cardiolipin. A point mutation changing the glycine at position 94 to glutamic acid in the BH3 domain of tBid (tBidG94E) was principally used because mitochondria treated with this mutant tBid displayed better preservation of the outer membrane than those treated with wild type tBid. Additionally, tBidG94E lowers the cytochrome c releasing activity of tBid without affecting its targeting to mitochondria. Electron microscope tomography coupled with immunogold labeling was used as a new hybrid technique to investigate the three-dimensional distributions of tBid and tBidG94E around the mitochondrial periphery. The statistics of spatial point patterns was used to analyze the association of these proteins with contact sites. CONCLUSIONS: Immunoelectron tomography with statistical analysis confirmed the preferential association of tBid with mitochondrial contact sites. These findings link these sites with cardiolipin in tBid targeting and suggest a role for Bcl-2 family members in regulating the activity of contact sites in relation to apoptosis. We propose a mechanism whereby Bcl-2 proteins alter mitochondrial function by disrupting cardiolipin containing contact site membranes.

Figure 1: Screenshot of the Annotation Tool

the annotator to override the initial tokenisation where necessary and we have developed a method of recording the result of overriding in the XML mark-up. This allows us to keep a record of the optimal annotation and ensures that it will not be necessary to take the expensive step of having data reannotated in the event that the tokenisation needs to be redone. As improved tokenisation procedures become available we can retokenise both the annotated material and the remaining unannotated data using a program which we have developed for this task. We describe the extension to the annotation tool, the XML representation of conflict and the retokenisation program in Section 4.

2 An XML-based Standoff Annotation Tool

In a number of recent projects we have explored the use of machine learning techniques for Named Entity Recognition (NER) and have worked with data from a number of different domains, including data from biomedicine (Finkel et al., in press; Dingare et al., 2004), law reports (Grover et al., 2004), social science (Nissim et al., 2004), and astronomy and astrophysics (Becker et al., 2005; Hachey et al., 2005). We have worked with a number of XML-based annotation tools, including the the NXT annotation tool (Carletta et al., 2003; Carletta et al., in press). Since we are interested only in written text and focus on annotation for Information Extraction (IE), much of the complexity offered by the NXT tool is not required and we have therefore recently implemented our own IE-specific tool. This has much in common with NXT, in particular annotations are encoded as standoff with pointers to the indices of the word tokens. A screenshot of the tool being used for NE annotation of biomedical text is shown in Figure 1. Figure 2 contains a fragment of the XML underlying the annotation for the excerpt

> “glutamic acid in the BH3 domain of tBid (tBidG94E) was principally used because ....”.

```
<body>
  .... <w id=‘w609’>glutamic</w> <w id=‘w618’>acid</w> <w id=‘w623’>in</w> <w id=‘w626’>the</w>
  <w id=‘w630’>BH3</w> <w id=‘w634’>domain</w> <w id=‘w641’>of</w> <w id=‘w644’>tBid</w>
  <w id=‘w649’>(</w><w id=‘w650’>tBidG94E</w><w id=‘w658’>)</w> <w id=‘w660’>was</w>
  <w id=‘w664’>principally</w> <w id=‘w676’>used</w> <w id=‘w681’>because</w> ....
</body>
<ents>
  <ent id=‘e7’ type=‘prot_frag’ sw=‘w630’ ew=‘w644’>BH3 domain of tBid</ent>
  <ent id=‘e8’ type=‘protein’ sw=‘w644’ ew=‘w644’>tBid</ent>
  <ent id=‘e9’ type=‘prot_frag’ sw=‘w650’ ew=‘w650’>tBidG94E</ent>
  <ent id=‘e10’ type=‘protein’ sw=‘w650’ ew=‘w650’ eo=‘–4’>tBid</ent>
</ents>
```

Figure 2: XML Encoding of the Annotation.

Note that the standoff annotation is stored at the bottom of the annotated file, not in a separate file. This is principally to simplify file handling issues which might arise if the annotations were stored separately. Word tokens are wrapped in w elements and are assigned unique ids in the id attribute. The tokenisation is created using significantly improved upgrades of the XML tools described in Thompson et al. (1997) and Grover et al. (2000)[1]. The ents element contains all the entities that the annotator has marked and the link between the ent elements and the words is encoded with the sw and ew attributes (*start word* and *end word*) which point at word ids. For example, the protein fragment entity with id e7 starts at the first character of the word with id w630 and ends at the last character of the word with id w644.

Our annotation tool and the format for storing annotations that we have chosen are just one instance of a wide range of possible tools and formats for the NE annotation task. There are a number of decision points involved in the development of such tools, some of which come down to a matter of preference and some of which are consequences of other choices. Examples of annotation methods which are not primarily based on XML are GATE (Cunningham et al., 2002) and the annotation graph model of Bird and Liberman (2001). The GATE system organises annotations in graphs where the start and end nodes have pointers into the source document character offsets. This is an adaptation of the TIPSTER architecture (Grishman, 1997). (The UIMA system from IBM (Ferrucci and Lally, 2004) also stores annotations in a TIPSTER-like format.) The annotation graph model encodes annotations as a directed graph with fielded records on the arcs and optional time references on the nodes. This is broadly compatible with our standoff XML representation and with the TIPSTER architecture. Our decision to use an annotation tool which has an underlying XML representation is partly for compatibility with our NLP processing methodology where a document is passed through a pipeline of XML-based components. A second motivation is the wish to ensure quality of annotation by imposing the constraint that annotations span complete XML elements. As explained above and described in more detail in Section 4 the consequence of this approach has been that we have had to develop a method for recording cases where the tokenisation is inconsistent with an annotator's desired action so that subsequent retokenisation does not require reannotation.

3 Tokenisation Issues

The most widely known examples of the NER task are the MUC competitions (Chinchor, 1998) and the CoNLL 2002 and 2003 shared task (Sang, 2002; Sang and De Meulder, 2003). In both cases the domain is newspaper text and the entities are general ones such as person, location, organisation etc. For this kind of data there are unlikely to be conflicts between tokenisation and entity mark-up and a vanilla tokenisation that splits at whitespace and punctuation is adequate. When dealing with scientific text and entities which refer to technical concepts, on the other hand, much more care needs to be taken with tokenisation.

In the SEER project we collected a corpus of abstracts of radio astronomical papers taken from the NASA Astrophysics Data System archive, a digital library for physics, astrophysics, and instru-

[1] Soon to be available under GPL as LT-XML2 and LT-TTT2 from `http://www.ltg.ed.ac.uk/`

mentation[2]. We annotated the data for the following four entity types:

Instrument-name Names of telescopes and other measurement instruments, e.g. *Superconducting Tunnel Junction (STJ) camera, Plateau de Bure Interferometer, Chandra, XMM-Newton Reflection Grating Spectrometer (RGS), Hubble Space Telescope.*

Source-name Names of celestial objects, e.g. *NGC 7603, 3C 273, BRI 1335-0417, SDSSp J104433.04-012502.2, PC0953+ 4749.*

Source-type Types of objects, e.g. *Type II Supernovae (SNe II), radio-loud quasar, type 2 QSO, starburst galaxies, low-luminosity AGNs.*

Spectral-feature Features that can be pointed to on a spectrum, e.g. *Mg II emission, broad emission lines, radio continuum emission at 1.47 GHz, CO ladder from (2-1) up to (7-6), non-LTE line.*

In the Text Mining programme (TXM) we have collected a corpus of abstracts and full texts of biomedical papers taken from PubMed Central, the U.S. National Institutes of Health (NIH) free digital archive of biomedical and life sciences journal literature[3]. We have begun to annotate the data for the following four entity types:

Protein Proteins, both full names and acronyms, e.g. *p70 S6 protein kinase, Kap-1, p130(Cas).*

Protein Fragment/Mutant Subparts or mutants of proteins e.g. $Bub1^{1-331}$, a domain of Bub1, nup53-δ405-430.

Protein Complex Complexes made up of two or more proteins e.g. Kap95p/Kap60, DOCK2-ELMO1, RENT complex. Note that nesting of protein entities inside complexes may occur.

Fusion Protein Fusions of two proteins or protein fragments e.g. β-catenin-Lef1, GFP-tubulin, GFP-EB1. Note that nesting of protein entities inside fusions may occur.

In both the astronomy and biomedical domains, there is a high density of technical and formulaic language (e.g. from astronomy: $(N(H_2) \simeq 10_{24} cm^{-2})$, 17.8 h_{70}^{-1} kpc, for $\Omega_m = 0.3$, $\Lambda = 0.7$, 30 μJy/beam). This technical nature means that the vanilla tokenisation style that was previously adequate for MUC-style NE annotation in generic newspaper text is no longer guaranteed to be a good basis for standoff NE annotation because there will inevitably be conflicts between the way the tokenisation segments the text and the strings that the annotators want to select. In the remainder of this section we illustrate this point with examples from both domains.

3.1 Tokenisation of Astronomy Texts

In our tokenisation of the astronomy data, we initially assumed a vanilla MUC-style tokenisation which gives strong weight to whitespace as a token delimiter. This resulted in 'words' such *Si[I]>0.4* and *I([OIII])* being treated as single tokens. Retokenisation was required because the annotators wanted to highlight *Si[I]* and *[OIII]* as entities of type Spectral-feature. We also initially adopted the practice of treating hyphenated words as single tokens so that examples such as *AGN-dominated* in the Source-type entity *AGN-dominated NELGs* were treated as one token. In this case the annotator wanted to mark *AGN* as an embedded Source-type entity but was unable to do so. A similar problem occurred with the Spectral-feature *BAL* embedded in the Source-type entity *mini-BAL quasar*.

Examples such as these required us to retokenise the astronomy corpus. We then performed a one-off, ad hoc merger of the annotations that had already been created with the newly tokenised version and then asked the annotators to revisit the examples that they had previously been unable to annotate correctly.

3.2 Tokenisation of Biomedical Texts

Our starting point for tokenisation of biomedical text was to use the finer grained tokenisation that we had developed for the astronomy data in preference to a vanilla MUC-style tokenisation. For the most part this resulted in a useful tokenisation; for example, rules to split at hyphens and slashes resulted in a proper tokenisation of protein complexes such as *Kap95p/Kap60* and *DOCK2-ELMO1* which allowed for the correct annotation of both the complexes and the proteins embedded within them. However, a slash did not always cause a token split and in cases such as *ERK 1/2* the *1/2* was treated as one token which prevented the annotator from marking up *ERK 1* as a protein. A catch-all rule for non-ASCII

[2] http://adsabs.harvard.edu/preprint_service.html

[3] http://www.pubmedcentral.nih.gov/

```
<body>
  .... <w id='w609'>glutamic</w> <w id='w618'>acid</w> <w id='w623'>in</w> <w id='w626'>the</w>
  <w id='w630'>BH3</w> <w id='w634'>domain</w> <w id='w641'>of</w> <w id='w644'>tBid</w>
  <w id='w649'>(</w><w id='w650'>tBid</w><w id='w654'>G94E</w><w id='w658'>)</w>
  <w id='w660'>was</w> <w id='w664'>principally</w> <w id='w676'>used</w>
  <w id='w681'>because</w> ....
</body>
<ents>
  <ent id='e7' type='prot_frag' sw='w630' ew='w644'>BH3 domain of tBid</ent>
  <ent id='e8' type='protein' sw='w644' ew='w644'>tBid</ent>
  <ent id='e9' type='prot_frag' sw='w650' ew='w654'>tBidG94E</ent>
  <ent id='e10' type='protein' sw='w650' ew='w650'>tBid</ent>
</ents>
```

Figure 3: Annotated File after Retokenisation.

characters meant that sequences containing Greek characters became single tokens when sometimes they should have been split. For example, in the string *PKCγK380R* the annotator wanted to mark *PKC* as a protein. Material in parentheses when not preceded by white space was not split off so that in examples such as *coilin(C214)* and *Cdt1(193-447)* the annotators were not able to mark up just the material before the left parenthesis. Sequences of numeric and (possibly mixed-case) alphabetic characters were treated as single tokens, e.g., *tBidG94E* (see Figure 2), *GAL4AD*, *p53TAD*—in these cases the annotators wanted to mark up an initial subpart (*tBid*, *GAL4*, *p53*).

4 Representing Conflict in XML and Retokenisation

Some of the tokenisation problems highlighted in the previous section arose because the NLP specialist implementing the tokenisation rules was not an expert in either of the two domains. Many initial problems could have been avoided by a phase of consultation with the astronomy and biomedical domain experts. However, because they are not NLP experts, it would have been time-consuming to explain the NLP issues to them.

Another approach could have been to use extremely fine-grained tokenisation perhaps splitting tokens on every change in character type.

Another way in which many of the problems could have been avoided might have been to use extremely fine-grained tokenisation perhaps splitting tokens on every change in character type. This would provide a strong degree of harmony between tokenisation and annotation but would be inadvisable for two reasons: firstly, segmentation into many small tokens would be likely to slow annotation down as well as give rise to more accidental mis-annotations because the annotators would need to drag across more tokens; secondly, while larger numbers of smaller tokens may be useful for annotation, they are not necessarily appropriate for many subsequent layers of linguistic processing (see Section 5).

The practical reality is that the answer to the question of what is the 'right' tokenisation is far from obvious and that what is right for one level of processing may be wrong for another. We anticipate that we might tune the tokenisation component a number of times before it becomes fixed in its final state and we need a framework that permits us this degree of freedom to experiment without jeopardising the annotation work that has already been completed.

Our response to the conflict between tokenisation and annotation is to extend our XML-based standoff annotation tool so that it can be used by the annotators to record the places where the current tokenisation does not allow them to select a string that they want to annotate. In these cases they can override the default behaviour of the annotation tool and select exactly the string they are interested in. When this happens, the standoff annotation points to the word where the entity starts and the word where it ends as usual, but it also records start and end character offsets which show exactly which characters the annotator included as part of the entity. The protein entity e10 in the example in Figure 2 illustrates this technique: the start and end word attributes sw and ew indicate that the entity encompasses the single token *tBidG94E* but the attribute eo (end offset) indicates that the annotator selected only the string *tBid*. Note that the annotator also correctly annotated the entire string *tBidG94E* as a protein fragment. The start and end character offset notation provides a subset of the range descriptions defined in the XPointer draft specification[4].

With this method of storing the annotators' decisions, it is now possible to update the tokenisation component and retokenise the data at any point during the annotation cycle without risk of losing completed annotation and without needing to ask annotators to revisit previous work. We have developed a program which takes as input the original annotated document plus a newly tokenised but unannotated version of it and which causes the correct annotation to be recorded in the retokenised version. Where the retokenisation accords with the annotators' needs there will be a decrease in the incidence of start and end offset attributes. Figure 3 shows the output of retokenisation on our example. The current version of the TXM project corpus contains 38,403 sentences which have been annotated for the four protein named entities described above (50,049 entity annotations). With the initial tokenisation (*Tok1*) there are 1,106,279 tokens and for 719 of the entities the annotators have used start and/or end offsets to override the tokenisation. We have defined a second, finer-grained tokenisation (*Tok2*) and used our retokenisation program to retokenise the corpus. This second version of the corpus contains 1,185,845 tokens and the number of entity annotations which conflict with the new tokenisation is reduced to 99. Some of these remaining cases reflect annotator errors while some are a consequence of the retokenisation still not being fine-grained enough. When using the annotations for training or testing, we still need a strategy for dealing with the annotations that are not consistent with our final automatic tokenisation routine (in our case, the 99 entities). We can systematically ignore the annotations or adjust them to the nearest token boundary. The important point is we we have recorded the mismatch between the tokenisation and the desired annotation and we have options for dealing with the discrepancy.

5 Tokenisation for Multiple Components

So far we have discussed the problem of finding the correct level of granularity of tokenisation purely in terms of obtaining the optimal basis for NER annotation. However, the reason for obtaining annotated data is to provide training material for NLP components which will be put together in a processing pipeline to perform information extraction. Given that statistically trained components such as part-of-speech (POS) taggers and NER taggers use word tokens as the fundamental unit over which they operate, their needs must be taken into consideration when deciding on an appropriate granularity for tokenisation. The implicit assumption here is that there can only be one layer of tokenisation available to all components and that this is the same layer as is used at the annotation stage. Thus, if annotation requires the tokenisation to be relatively fine-grained, this will have implications for POS and NER tagging. For example, a POS tagger trained on a more conventionally tokenised dataset might have no problem assigning a propernoun tag to *Met-tRNA/eIF2·* in

> ... and facilitates loading of the Met-tRNA/eIF2· GTP ternary complex ...

however, it would find it harder to assign tags to members of the 10 token sequence *M et - t RNA / e IF 2 ·*.

Similarly, a statistical NER tagger typically uses information about left and right context looking at a number of tokens (typically one or two) on either side. With a very fine-grained tokenisation, this representation of context will potentially be less informative as it might contain less actual context. For example, in the excerpt

> ... using a Tet-on LMP1 HNE2 cell line ...

assuming a fine-grained tokenisation, the pair of tokens *LMP* and *1* make up a protein entity. The left context would be the sequence *using a Tet - on* and the right context would be *HNE 2 cell line*. Depending on the size of window used to capture context this may or may not provide useful information.

To demonstrate the effect that a finer-grained tokenisation can have on POS and NER tagging, we performed a series of experiments on the NER annotated data provided for the Coling BioNLP evaluation (Kim et al., 2004), which was derived from the GENIA corpus (Kim et al., 2003). (The BioNLP data is annotated with five entities, protein, DNA, RNA, cell_type and cell_line.) We trained the C&C maximum entropy tagger (Curran and Clark, 2003) using default settings to obtain

[4] `http://www.w3.org/TR/xptr-xpointer/`

	Orig	*Tok1*	*Tok2*
training # sentences	18,546		
eval # sentences	3,856		
training # tokens	492,465	540,046	578,661
eval # tokens	101,028	110, 352	117, 950
Precision	65.14%	62.36%	61.39%
Recall	67.35%	64.24%	63.24%
F1	66.23%	63.27%	62.32%

Table 1: NER Results for Different Tokenisations of the BioNLP corpus

NER models for the original tokenisation (*Orig*), a retokenisation using the first TXM tokeniser (*Tok1*) and a retokenisation using the finer-grained second TXM tokeniser (*Tok2*) (see Section 4). In all experiments we discarded the original POS tags and performed POS tagging using the C&C tagger trained on the MedPost data (Smith et al., 2004). Table 1 shows precision, recall and f-score for the NER tagger trained and tested on these three tokenisations and it can be seen that performance drops as tokenisation becomes more fine-grained.

The results of these experiments indicate that care needs to be taken to achieve a sensible balance between the needs of the annotation and the needs of NLP modules. We do not believe, however, that the results demonstrate that the less fine-grained original tokenisation is necessarily the best. The experiments are a measure of the combined performance of the POS tagger and the NER tagger and the tokenisation expectations of the POS tagger must also have an impact. We used a POS tagger trained on material whose own tokenisation most closely resembles the tokenisation of *Orig* (hyphenated words are not split in the MedPost training data) and it is likely that the low results for *Tok1* and *Tok2* are partly due to the tokenisation mismatch between training and testing material for the POS tagger. In addition, the NER tagger was used with default settings for all runs where the left and right context is at most two tokens. We might expect an improvement in performance for *Tok1* and *Tok2* if the NER tagger was run with larger context windows. The overall message here, therefore, is that the needs of all processors must be taken into account when searching for an optimal tokenisation and developers should beware of bolting together components which have different expectations of the tokenisation—ideally each should be tuned to the same tokenisation.

There is a further reason why the original tokenisation of the BioNLP data works so well. During our experiments with the original data we observed that splitting at hyphens was normally not done (e.g. *monocyte-specific* is one token) but wherever an entity was part of a hyphenated word then it was split (e.g. *IL-2 -independent* where *IL-2* is marked as a protein.) The context of a following word which begins with a hyphen is thus a very clear indicator of entityhood. Although this will improve scores where the training and testing data are marked in the same way, it gives an unrealistic estimate of actual performance on unseen data where we would not expect the hyphenation strategy of an automatic tokeniser to be dependent on prior knowledge of where the entities are. To demonstrate that the *Orig* NER model does not perform well on differently tokenised data, we tested it on the *Tok1* tokenised evaluation set and obtained an f-score of 55.64%.

6 Conclusion

In this paper we have discussed the fact that tokenisation, especially of scientific text, is not necessarily a component that can be got right first time. In the context of annotation tools, especially where the tool makes reference to the tokenisation layer as with XML standoff, there is an interdependence between tokenisation and annotation. It is not practical to have annotators revisit their work every time the tokenisation component changes and so we have developed a tool that allows annotators to override tokenisation where necessary. The annotators' actions are recorded in the XML format in such a way that we can retokenise the corpus and still faithfully reproduce the original annotation. We have provided very specific motivation for our approach from our annotation of the astronomy and biomedical domains but we hope that this method might be taken up as a standard elsewhere as it would provide benefits when sharing corpora—a corpus annotated in this way can be used by a third party and possibly retokenised by them to suit their needs. We also looked at the interdependence between the tokenisation used for annotation and the tokenisation requirements of POS taggers and NER taggers. We showed that it is important to provide a consistent tokenisation throughout and that experimentation is required before the optimal balance can be found. Our retokenisation tools support just this kind of experimentation

Acknowledgements

The work reported here was supported by the ITI Life Sciences Text Mining programme (www.itilifesciences.com), and in part by Edinburgh-Stanford Link Grant (R36759) as part of the SEER project. All Intellectual Property arising from the Text Mining programme is the property of ITI Scotland Ltd.

References

Markus Becker, Ben Hachey, Beatrice Alex, and Claire Grover. 2005. Optimising selective sampling for bootstrapping named entity recognition. In *Proceedings of the ICML-2005 Workshop on Learning with Multiple Views*. Bonn, Germany.

Steven Bird and Mark Liberman. 2001. A formal framework for linguistic annotation. *Speech Communication*, 33(1,2):23–60.

Jean Carletta, Stefan Evert, Ulrich Heid, Jonathan Kilgour, Judy Robertson, and Holger Voormann. 2003. The NITE XML toolkit: flexible annotation for multi-modal language data. *Behavior Research Methods, Instruments, and Computers*, 35(3):353–363.

Jean Carletta, Stefan Evert, Ulrich Heid, and Jonathan Kilgour. in press. The NITE XML toolkit: data model and query. *Language Resources and Evaluation*.

Nancy A. Chinchor. 1998. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Fairfax, Virginia.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the Association for Computational Linguistics*.

James R. Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-2003*, pages 164–167.

Shipra Dingare, Jenny Finkel, Malvina Nissim, Christopher Manning, and Claire Grover. 2004. A system for identifying named entities in biomedical text: How results from two evaluations reflect on both the system and the evaluations. In *Proceedings of the 2004 BioLink meeting: Linking Literature, Information and Knowledge for Biology, at ISMB 2004*.

David Ferrucci and Adam Lally. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

Jenny Finkel, Shipra Dingare, Christopher Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. in press. Exploring the boundaries: Gene and protein identification in biomedical text. *BMC Bioinformatics*, 6 (Suppl 1).

Ralph Grishman. 1997. TIPSTER Architecture Design Document Version 2.3. *Technical report, DARPA*, `http://www.itl.nist.gov/div894/894.02/related_projects/tipster/`.

Claire Grover, Colin Matheson, Andrei Mikheev, and Marc Moens. 2000. LT TTT—a flexible tokenisation tool. In *LREC 2000—Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pages 1147–1154.

Claire Grover, Ben Hachey, and Ian Hughson. 2004. The HOLJ corpus: Supporting summarisation of legal texts. In *Proceedings of the 5th International Workshop on Linguistically Interpreted Corpora (LINC-04)*. Geneva, Switzerland.

Ben Hachey, Beatrice Alex, and Markus Becker. 2005. Investigating the effects of selective sampling on the annotation task. In *Proceedings of the 9th Conference on Computational Natural Language Learning*. Ann Arbor, Michigan, USA.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl.1):180–182.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the Bio-Entity Recognition Task at JNLPBA. In *Proceedings of the International Joint Workshop on NLP in Biomedicine and its Applications*, pages 70–75.

Malvina Nissim, Colin Matheson, and James Reid. 2004. Recognising geographical entities in Scottish historical documents. In *Proceedings of the Workshop on Geographic Information Retrieval at SIGIR 2004*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the 2003 Conference on Computational Natural Language Learning*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 2002 Conference on Computational Natural Language Learning*.

L. Smith, T. Rindflesch, and W. J. Wilbur. 2004. MedPost: a part-of-speech tagger for biomedical text. *Bioinformatics*, 20(14):2320–2321.

Henry Thompson, Richard Tobin, David McKelvie, and Chris Brew. 1997. LT XML. software API and toolkit for XML processing. `http://www.ltg.ed.ac.uk/software/`.