



ELSEVIER

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Marky: A tool supporting annotation consistency in multi-user and iterative document annotation projects

Martín Pérez-Pérez^a, Daniel Glez-Peña^a, Florentino Fdez-Riverola^a,
Anália Lourenço^{a,b,*}

^a ESEI – Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, Universidad de Vigo, 32004 Ourense, Spain¹

^b Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal

ARTICLE INFO

Article history:

Received 24 July 2014

Received in revised form

24 October 2014

Accepted 18 November 2014

Keywords:

Document annotation

Collaborative annotation

Iterative annotation

Inter-annotator agreement

Tracking system

ABSTRACT

Background and objectives: Document annotation is a key task in the development of Text Mining methods and applications. High quality annotated corpora are invaluable, but their preparation requires a considerable amount of resources and time. Although the existing annotation tools offer good user interaction interfaces to domain experts, project management and quality control abilities are still limited. Therefore, the current work introduces Marky, a new Web-based document annotation tool equipped to manage multi-user and iterative projects, and to evaluate annotation quality throughout the project life cycle.

Methods: At the core, Marky is a Web application based on the open source CakePHP framework. User interface relies on HTML5 and CSS3 technologies. Rangy library assists in browser-independent implementation of common DOM range and selection tasks, and Ajax and JQuery technologies are used to enhance user–system interaction.

Results: Marky grants solid management of inter- and intra-annotator work. Most notably, its annotation tracking system supports systematic and on-demand agreement analysis and annotation amendment. Each annotator may work over documents as usual, but all the annotations made are saved by the tracking system and may be further compared. So, the project administrator is able to evaluate annotation consistency among annotators and across rounds of annotation, while annotators are able to reject or amend subsets of annotations made in previous rounds. As a side effect, the tracking system minimises resource and time consumption.

Conclusions: Marky is a novel environment for managing multi-user and iterative document annotation projects. Compared to other tools, Marky offers a similar visually intuitive annotation experience while providing unique means to minimise annotation effort and enforce annotation quality, and therefore corpus consistency. Marky is freely available for non-commercial use at <http://sing.ei.uvigo.es/marky>.

© 2014 Elsevier Ireland Ltd. All rights reserved.

* Corresponding author at: ESEI – Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, Universidad de Vigo, 32004 Ourense, Spain. Tel.: +34 988 387013; fax: +34 988 387001.

E-mail address: analialourenco@uvigo.es (A. Lourenço).

¹ <http://sing.ei.uvigo.es/>.

<http://dx.doi.org/10.1016/j.cmpb.2014.11.005>

0169-2607/© 2014 Elsevier Ireland Ltd. All rights reserved.

1. Introduction

Text Mining (TM) has a wide range of applications that require differentiated processing of documents of various nature [1,2]. Ultimately, the goal is to learn how to recognise and contextualise information of interest [3]. Therefore, the annotation of documents by domain experts is invaluable to provide for a ground truth against which to train and evaluate TM methods and algorithms.

Depending on the application area, the creation of such semantically annotated corpora is a resource and time consuming activity [4,5]. Usually, multiple domain experts should review the documents and manual annotations should be compared. The initial set of annotation guidelines often fails to anticipate some of the semantics issues and it is highly unlikely that multiple annotators completely agree on the annotation of a document [6–8]. So, annotation consistency needs to be monitored through the multiple rounds of the project in order to identify relevant differences in annotation patterns and make opportune amendments to the annotation guidelines and schema, and thus, guarantee the quality of the generated corpus. This entails the active monitoring and reinforcement of inter-annotator consistency (i.e. two annotators should annotate the same text fragment equally) and intra-annotator consistency (i.e. if an annotator should annotate multiple occurrences of same text fragment similarly, within the same context).

Annotation tools are expected to manage such multi-user and iterative annotation projects actively and efficiently. So far, most of the document annotation tools available can be differentiated in terms of the task-specific specialisation of the interface, i.e. the modularity and configurability of the annotation environment [9], whereas providing similar limited quality monitoring and control abilities. Just as means of comparison, Table 1 summarises the main characteristics of some of the most recent annotation tools, and the tool presented in this work. The U-Compare of the Apache Unstructured Information Management Architecture (UIMA) [10] and the Teamware of the General Architecture for Text Engineering (GATE) [11] are two meaningful examples of open-source framework-integrated document annotation tools. Argo stands as a workbench for building and running text analysis solutions [12], while MyMiner [13], EGAS [14], PubTator [15], BioNotate [16] and BioAnnote [17] are examples of free independent annotation tools, all of which are commonly used in biomedical applications. Finally, A.nnotate (<http://a.nnotate.com/>), MMAX2 (<http://mmax2.sourceforge.net/>) and annotateit/Annotator (<http://annotateit.org/>) are experienced broad-application systems.

Seeking to overcome some of the current limitations, this paper presents Marky, a free Web-based document annotation tool, which implements the main steps of the annotation project life cycle, with particular emphasis on annotation quality assessment. Notably, its novelty lays on (i) the annotation tracking system, which ensures that all actions occurring in the annotation project are recorded and may be amended and (ii) the annotation quality evaluation

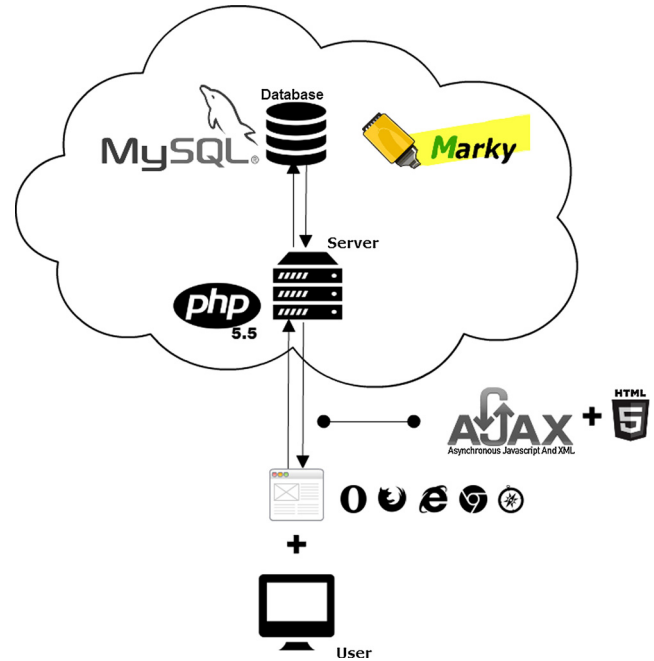


Fig. 1 – Main technologies supporting the architecture of Marky.

tool, which monitors inter-annotator agreement (IAA) and intra-annotator patterns.

In the next sections, the architecture of Marky is described, in terms of the main design requirements and the implemented annotation life cycle. A case study taken from the literature is used to exemplify the abilities of Marky in practical terms. The final discussion stresses the importance of enforcing annotation consistency and assisting the work of annotators, and presents near future developments.

2. Methods

Marky is a general purpose Web-based application for document annotation. This section describes the requirements that motivated some of the aspects of its design, the overall system architecture, and the annotation life cycle implemented by the tool.

2.1. Requirements

From the start, Marky was designed to support general purpose annotation, i.e. domain specifications are considered only in project configuration and do not affect the behaviour of the software. This choice has allowed us to concentrate more on the infrastructure than on the application itself.

Marky is built on top of open technologies and standards to grant extensibility and interoperability with other systems. Internally, project configuration and management is kept simple, but flexible. A project may have associated multiple annotators and include several rounds of annotation. The corpus to be annotated may be loaded through a Web-accessible bibliographic service, from a local folder, or by copying contents manually. Marky handles structured documents,

Table 1 – Comparison of some public and well-known annotation systems with Marky, the tool presented in this work.

Tool	Year	Load annotated documents	Customised types and colours	Multi-user	Search and annotate	Annotation agreement	Annotation's commentary	Rich text support	Annotation relationships	Install	Dependencies	Input	Output	License
A.nnotate	2013	Y	N	Y	Y	N	Y	Y	N	Web	Web browser	TXT, WORD, PDF, HTML	PDF, DOCX, XML	Proprietary
anotateit/ Anotator	2009	Y	N	Y	N	N	Y	Y	N	Web	Web browser	Web page	JSON	MIT
Anotation- Studio	2012	Y	N	Y	N	N	Y	Y	N	SA, Web	PostgreSQL, Nde.js, NPM, MongoDB	WORD, PDF, TXT	-	GNU GPL2
Argo	2012	N	N	Y	N	Y	N	N	Y	Web	Web browser	TXT	TXT, A1, A2	Proprietary
BioNotate	2009	N	N	N	N	N	N	N	N	Web	Apache, Perl	TXT	XML, SO	GNU GPL
BioAnnote	2013	Y	Y	N	Y	N	N	N	N	SA	Java	TXT, WORD, PDF, XML	XML	GPLv3
Egas	2014	Y	Y (max. 14 colours)	Y	N	N	N	N	Y	Web	Web browser	TXT, A1, XML	TXT, XML, ANN	Proprietary
GATE Teamware	2010	Y	Y	Y	Y	N	N	N	N	SA	Java, Perl, MySQL	TXT	HTM, XML, IL, DB	GNU AGPL
Marky	2013	Y	Y	Y	Y	Y	Y	Y	N	SA, Web	MySQL, PHP, Web browser	TXT, HTML, XML	HTML, TXT+TSV	GNU GPL3
MMAX2	2006	Y	N	N	N	N	N	N	Y	SA	Java	XML, Web page	HTML, XSL	Apache License V2.0
My Miner	2012	N	N	N	N	N	N	N	N	Web	Web browser	TXT, PDF	TXT, IL	Proprietary
Pubtator	2013	N	Y (max. 144 colours)	N	N	N	N	N	Y	Web	Web browser	TXT	XML	Proprietary
UIMA	2006	Y	Y	N	Y	N	N	N	N	SA	Java	XML	XML	Apache License V2.0

Y, yes; N, no; NPM, node package manager; SA, stand alone.

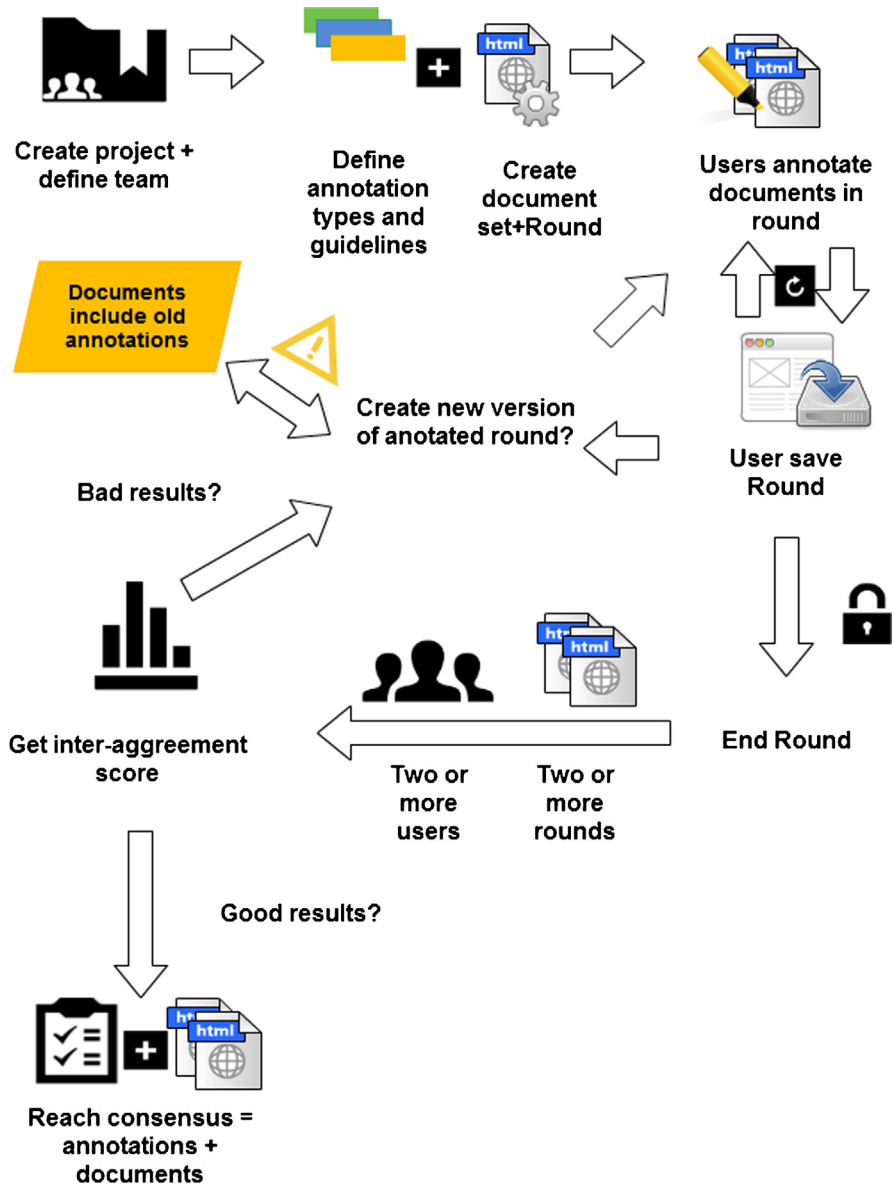


Fig. 2 – The annotation project life cycle implemented by Marky.

namely HTML or XML-based structured documents, Word documents and Word documents, and plain text documents.

Most likely, different projects will have different annotation necessities and thus, the project administrator should configure the annotation guidelines and annotation types according to the application. Both documents and annotations are stored in a relational database. This is a main requirement to implement the annotation tracking system and make quality evaluation more flexible and computationally efficient. In particular, the system supports bulk annotation amendments and the addition or removal of annotation types during annotation rounds.

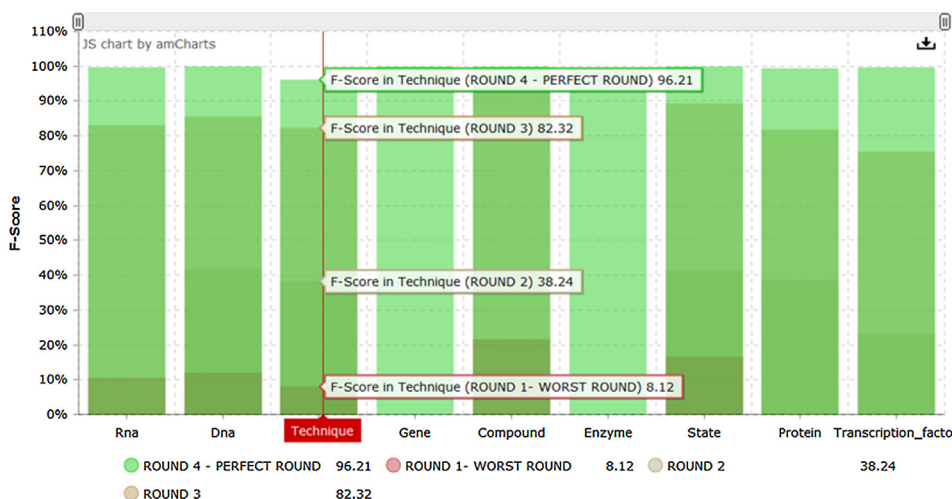
Finally, user interface is based on the “What You See Is What You Get” (WYSIWYG) paradigm. The designed interface allows the annotator to directly tag text fragments

without modifying the layout. The type of an annotation is domain-specific metadata about the annotation itself, which the quality evaluation module uses to identify domain-specific semantics and terminological issues. Users may also add notes on the decisions made and cross-link annotations to external data sources (e.g. database or ontology entries).

2.2. System architecture

Marky was developed using the open source CakePHP Web framework, which follows the MVC (Model-View-Controller) model [18]. At the core of the architecture of Marky are several consolidated Web technologies (Fig. 1): PHP 5.5 and MySQL 5.1.73 are responsible for server side operations; system interface relies on the HTML5 (<http://www.w3.org/TR/html5/>)

F-score across rounds



Details on round annotator agreement



Fig. 5 – Two examples of annotation consistency evaluations implemented in Marky. The plot shows IAA results across rounds and for all the annotation types. The table details the agreement between the two annotators in round one such that the diagonal identifies concordant annotations and the rest of the cells identify type and term mismatches.

corpus to be annotated, the annotation guidelines and the set of annotation types to be considered, and selects the individuals that will participate as annotators. Documents may be automatically retrieved from an online source, e.g. PubMed Central, or manually uploaded by the project administrator.

A first round of annotation is launched as soon as the project is created. Usually, the annotation process is split into training and active annotation. During training the annotators should become familiar with the task and the tool.

After that, annotation consistency becomes the focus of the data workflow and will determine the number of iterations to be conducted.

The annotation tracking system of Marky records the set of annotations produced by the individual annotators at each round. The IAA scores measure the degree of inconsistency affecting the results. For example, issues related to domain expertise, semantic ambiguity and glitches in the annotation guidelines. According to the severity of the issues detected,

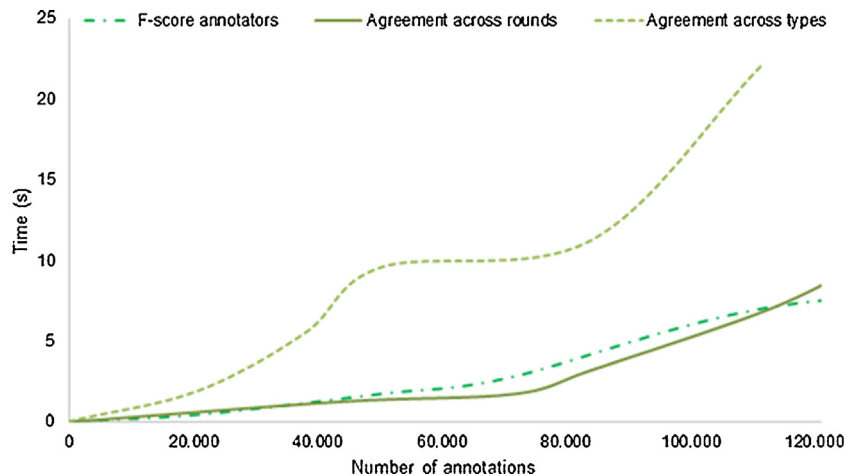


Fig. 6 – Performance of Marky during quality assessment. The calculation of the IAA in a single round and across rounds shows similar performance. The calculation of the agreement across annotation types, which may involve multiple annotators and rounds, is more complex and gets affected by the volume of annotations involved.

the project administrator may decide to initiate a new round of annotation with a refined set of guidelines. Even, it may be the case that the set of annotation types should be altered. The tracking system offers a precious help here by enabling that annotators may reject or amend subsets of their previous annotations, preserving good quality annotations, and avoiding starting the process from scratch. Furthermore, to preserve round annotations, and enable comparison throughout the whole project, Marky keeps record of the set of annotation types and guidelines used at each round of annotation.

Project execution terminates after the administrator considers that the IAA is acceptable. A consensus version of the annotations is then created and the obtained annotated corpus becomes available for download and further application (e.g. the training and testing of Text Mining models).

2.4. Document encoding and annotation format

The management of documents and annotations is supported by a relational database. Document contents are saved in HTML format with UTF-8 encoding. Annotations describe the span of text tagged, the corresponding char offset, the domain-specific type, and any additional notes made by the annotator.

The database manages all the data generated during the annotation rounds. That is, Marky keeps record of the set of annotations made by each individual annotator at every round of annotation. This is crucial to the functioning of the annotation tracking system and the quality assessment module.

Regarding annotation itself, Marky takes advantage of the HTML 5 tag *mark* to implement the function. This tag enables the association of domain-specific information to text spans while offering an intuitive, common Web browsing experience to the annotator (Fig. 3).

In terms of annotation exportation, Marky offers the possibility to store the annotated documents produced by individual annotator, at any of the rounds of annotation, as well as the consensus corpus. Individual annotator results are

exported in inline format, i.e. annotations are stored within the annotated document. The consensus corpus is stored in a stand-off format, i.e. annotations are stored separately from the annotated document.

2.5. Quality assessment

Quality assessment entails inter-annotator agreement per round and annotation consistency across multiple rounds of annotation. The rate of agreement among annotators or among rounds is quantified using the F-score, a common metric in IAA evaluations [19–21]. The standard measures of recall, precision and F-score are calculated as follows:

$$F\text{-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{precision} = \frac{\text{number of identical annotations in set A and set B}}{\text{number of annotations in set A}}$$

$$\text{recall} = \frac{\text{number of identical annotations in set A and set B}}{\text{number of annotations in set B}}$$

Marky enables IAA calculations to be stricter or more relaxed. It is possible to calculate agreement rates corresponding to exact annotation matches, where text spans identified by a pair of annotators match exactly, and relaxed annotation matches, where text spans identified by a pair of annotators at least overlap with each other by a user-specified number of characters, but do not necessarily match exactly.

3. Results

This section presents a case scenario that demonstrates the main features and GUI abilities of Marky. A manually annotated corpus describing the integrated cellular responses to nutrient starvation in the model-organism *Escherichia coli* was used as case study [8]. The corpus is formed by 129 abstracts and contains annotations for main biological species such as DNA, RNA, gene, protein, enzyme, transcription factor and compound. The original set of annotations was broken apart

Annotations

Below is a detailed table with annotations and users who have annotated, select those annotations that will in the final consensus

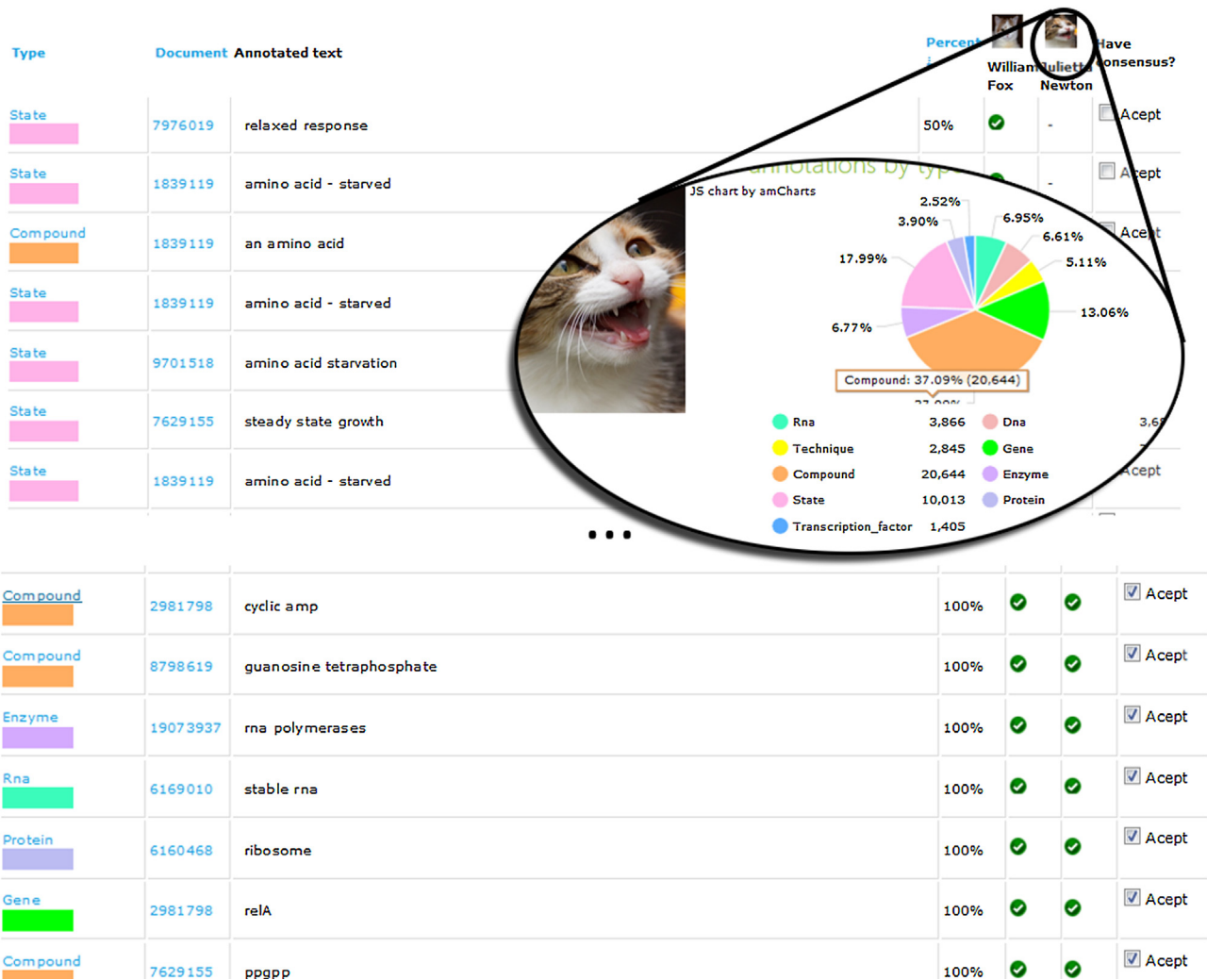


Fig. 7 – Final annotation consensus and corpus exportation at Marky. The project administrator is presented with the annotation agreement per annotation and annotation type, and may specify a minimum threshold to automatically accept annotations.

so to create different types of inconsistencies among two annotators into multiple, fictional rounds of annotation. Next, we show how Marky manages such project. Further inspection of the case study is possible by accessing the full demo version at <http://sing.ei.uvigo.es/markyapp>.

3.1. Project management

At the creation, the project administrator has to indicate the title and a short description of the project, introduce a first version of the annotation guidelines and the annotation scheme, and associate the annotators. Here, the project is named “Stringent response in *Escherichia coli*” and includes two fictitious annotators – Julietta Newton and William Fox (Fig. 4).

During the course of the project, the annotation guidelines and the annotation scheme, i.e. the set of annotation types considered, change in response to the IAA inconsistencies

that are represented and solved at each round. In particular, the project includes four rounds of annotation as follows: in round one there is a high number of inconsistencies, recreating issues in semantic interpretation and annotation types that are handled inconsistently; in round two, the annotation type “compound” is excluded and the issues are minimised; in round three, the annotation type “compound” is included again with new guidelines of annotation; and, in round four the IAA is considered acceptable and the user may generate the final consensus.

3.2. Round and user statistics evaluation

The evaluation of annotation consistency has been made as flexible as possible so that the project administrator may look over inconsistencies among annotators, within and throughout rounds, as well as differences in the annotation pattern exhibited by individual annotators. The ultimate goal is to

identify the causes for a high number of differences or a low *F*-score, so that the annotation scheme and guidelines may be suitably refined (Fig. 5).

Annotation types which cause large numbers of differences (i.e. a small number of highly occurring terms) are good candidates for improving IAA, because a better handling of these categories has the potential of eliminating a large numbers of differences. A very low *F*-score identifies categories which are handled inconsistently (i.e. a large number of term mismatches), and their handling will probably require a meeting with the annotators and a thorough redefinition of the annotation guidelines.

The above described quality assessments have been programmed and optimised to support high dimensionality. Tests were performed on an Intel i7 (3.4 GHz) with 1 GB of RAM (Fig. 6). The IAA and the verification of consistency across rounds are based on annotation offset matching and are quicker to calculate. However, the verification of consistency across annotation types is more complex and thus, more demanding. This operation needs to account for annotations that agree both in type and offset, annotations that match only in offset (i.e. the same text fragment was annotated but the annotation types are inconsistent), and annotations that are unique to an annotator.

Additionally, we can observe that the system only shows a noticeable deterioration of process time when dealing with a very large set of annotations, typically 80,000 or more annotations. This deterioration is justified by requisites of database indexing and memory management that will be tackled in future releases of the tool.

3.3. Results exportation

Obtaining the final consensus is essential to finalise the production of the annotated corpus. Likely, the project will terminate without reaching a 100% of IAA. Therefore, the project administrator needs to determine when to finalise the iterative annotation process and then, decide which annotations that did not get consensus among the annotators should be included. Such task may be laborious given the number of annotations that can be made in a medium to large size corpus. For this reason, Marky enables the manual inclusion of non-consensual annotations as well as the automatic inclusion of annotations that meet a minimum agreement threshold (Fig. 7).

The consensual corpus is then copied to a final round (round four in this case study) and becomes available for download. The consensus package includes the original corpus and the annotations in stand-off format.

4. Conclusions

The production of high quality annotated corpora, essential for knowledge acquisition tasks, is challenged by inevitable inter-annotator discrepancy. Annotators can hardly agree completely on what to annotate and exactly how to annotate. For this reason, document annotation projects usually involve multiple domain experts working in an iterative and

collaborative manner to identify and resolve discrepancies progressively.

Such a detailed annotation life cycle takes significant time and effort, and demands from document annotation tools as much assistance as possible. The key to improve annotation quality is to actively monitor inconsistencies throughout the rounds, and resolve the most critical issues as soon and as effectively as possible. Compared to other annotation tools, Marky provides novel and powerful means to assess annotation quality and make amendments to annotations automatically. Furthermore, Marky is built using the CakePHP framework in order to promote extensibility and abstraction across the diversity of applications that may require the preparation of annotated corpora. Its modular design relies on state-of-the-art Web technologies as means to ensure wide user–system interaction and tool interoperability.

Plans for Marky development in the near future include, among others: increased flexibility and reusability of standard GUI library components for text annotation; provision of basic GUI library components for annotation of different types of concepts and relationships; and, support for various forms of automated tagging (e.g. ontology-driven annotation and machine learning classifiers) by external software modules.

Mode of availability

Marky software, comprehensive documentation of the functionalities, and usage examples, are freely available for non-commercial use at <http://sing.ei.uvigo.es/marky>.

Conflict of interests

The authors do not have any competing interests.

Acknowledgements

The authors thank the project PTDC/SAU-ESA/646091/2006/FCOMP-01-0124-FEDER-007480FCT, the Strategic Project PEst-OE/EQB/LA0023/2013, the Project “Bio-Health – Biotechnology and Bioengineering approaches to improve health quality”, Ref. NORTE-07-0124-FEDER-000027, co-funded by the Programa Operacional Regional do Norte (ON.2 – O Novo Norte), QREN, FEDER, the project “RECI/BBB-EBI/0179/2012 – Consolidating Research Expertise and Resources on Cellular and Molecular Biotechnology at CEB/IBB”, Ref. FCOMP-01-0124-FEDER-027462, FEDER, and the Agrupamento INBIOMED from DXPCTSUG-FEDER unha maneira de facer Europa (2012/273). The research leading to these results has received funding from the European Union’s Seventh Framework Programme FP7/REGPOT-2012-2013.1 under grant agreement no. 316265 (BIOCAPS) and the [14VI05] Contract-Programme from the University of Vigo. This document reflects only the author’s views and the European Union is not liable for any use that may be made of the information contained herein.

REFERENCES

- [1] D. Rebholz-Schuhmann, A. Oellrich, R. Hoehndorf, Text-mining solutions for biomedical research: enabling integrative biology, *Nat. Rev. Genet.* 13 (2012) 829–839.
- [2] G. Miner, *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*, Academic Press, 2012.
- [3] V. Slavov, P. Rao, S. Paturi, T.K. Swami, M. Barnes, D. Rao, et al., A new tool for sharing and querying of clinical documents modeled using HL7 Version 3 standard, *Comput. Methods Programs Biomed.* 112 (2013) 529–552.
- [4] M. Martínez-Romero, J.M. Vázquez-Naya, J. Pereira, A. Pazos, BioOSS: a system for biomedical ontology selection, *Comput. Methods Programs Biomed.* 114 (2014) 125–140.
- [5] A. Gastounioti, V. Koliás, S. Golemati, N.N. Tsiaparas, A. Matsakou, J.S. Stoitsis, et al., CAROTID – a web-based platform for optimal personalized management of atherosclerotic patients, *Comput. Methods Programs Biomed.* 114 (2014) 183–193.
- [6] R.I. Doğan, R. Leaman, Z. Lu, NCBI disease corpus: a resource for disease name recognition and concept normalization, *J. Biomed. Inform.* 47 (2014) 1–10.
- [7] E.M. van Mulligen, A. Fourier-Reglat, D. Gurwitz, M. Molokhia, A. Nieto, G. Trifiro, et al., The EU-ADR corpus: annotated drugs, diseases, targets, and their relationships, *J. Biomed. Inform.* 45 (2012) 879–884.
- [8] R. Carreira, S. Carneiro, R. Pereira, M. Rocha, I. Rocha, E.C. Ferreira, et al., Semantic annotation of biological concepts interplaying microbial cellular responses, *BMC Bioinformatics* 12 (2011) 460.
- [9] M. Neves, U. Leser, A survey on annotation tools for the biomedical literature, *Brief. Bioinform.* 15 (2012) 327–340.
- [10] Y. Kano, W.A. Baumgartner, L. McCrohon, S. Ananiadou, K.B. Cohen, L. Hunter, et al., U-Compare: share and compare text mining tools with UIMA, *Bioinformatics* 25 (2009) 1997–1998.
- [11] K. Bontcheva, H. Cunningham, I. Roberts, A. Roberts, V. Tablan, N. Aswani, et al., GATE Teamware: a web-based, collaborative text annotation framework, *Lang. Resour. Eval.* 47 (2013) 1007–1029.
- [12] R. Rak, A. Rowley, W. Black, S. Ananiadou, Argo: an integrative, interactive, text mining-based workbench supporting curation, *Database (Oxford)* (2012), bas010.
- [13] D. Salgado, M. Krallinger, M. Depaule, E. Drula, A.V. Tendulkar, F. Leitner, et al., MyMiner: a web application for computer-assisted biocuration and text annotation, *Bioinformatics* 28 (2012) 2285–2287.
- [14] D. Campos, J. Lourenço, S. Matos, J.L. Oliveira, Egas: a collaborative and interactive document curation platform, *Database (Oxford)* (2014) 1–12.
- [15] C.-H. Wei, H.-Y. Kao, Z. Lu, PubTator: a web-based text mining tool for assisting biocuration, *Nucleic Acids Res.* 41 (2013) W518–W522.
- [16] C. Cano, T. Monaghan, A. Blanco, D.P. Wall, L. Peshkin, Collaborative text-annotation resource for disease-centered relation extraction from biomedical text, *J. Biomed. Inform.* 42 (2009) 967–977.
- [17] H. López-Fernández, M. Reboiro-Jato, D. Glez-Peña, F. Aparicio, D. Gachet, M. Buenaga, et al., BioAnnote: a software platform for annotating biomedical documents with application in medical learning environments, *Comput. Methods Programs Biomed.* 111 (2013) 139–147.
- [18] M. Iglesias, *CakePHP 1.3 Application Development Cookbook*, Packt Publishing, 2011.
- [19] P. Thompson, S.A. Iqbal, J. McNaught, S. Ananiadou, Construction of an annotated corpus to support biomedical information extraction, *BMC Bioinformatics* 10 (2009) 349.
- [20] G. Hripcsak, A.S. Rothschild, Agreement, the *f*-measure, and reliability in information retrieval, *J. Am. Med. Inform. Assoc.* 12 (2005) 296–298.
- [21] T. Brants, Inter-annotator agreement for a German newspaper corpus, in: *Second Int. Conf. Lang. Resour. Eval.*, 2000.