Marky: A Lightweight Web Tracking Tool for Document Annotation

Martín Pérez-Pérez[1], Daniel Glez-Peña[1], Florentino Fdez-Riverola[1], and Anália Lourenço[1,2]

[1] ESEI - Escuela Superior de Ingeniería Informática, Edificio Politécnico, Campus Universitario As Lagoas s/n, Universidad de Vigo, 32004 Ourense, Spain
[2] IBB - Institute for Biotechnology and Bioengineering, Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
mpperez3@esei.uvigo.es, {dgpena,riverola}@uvigo.es, analia@{ceb.uminho.pt,uvigo.es}

Abstract. Document annotation is an elementary task in the development of Text Mining applications, notably in defining the entities and relationships that are relevant to a given domain. Many annotation software tools have been implemented. Some are particular to a Text Mining framework while others are typical stand-alone tools. Regardless, most development efforts were driven to basic functionality, i.e. performing the annotation, and to interface, making sure operation was intuitive and visually appellative. The deployment of large-scale annotation jamborees and projects showed the need for additional features regarding inter- and intra-annotation management. Therefore, this paper presents Marky, a new Web-based document annotation tool that integrates a highly customisable annotation environment with a robust project management system. Novelty lays on the annotation tracking system, which supports *per* user and *per* round annotation change tracking and thus, enables automatic annotation correction and agreement analysis.

Keywords: Text mining, document annotation, annotation guidelines, inter-annotator agreement, Web application.

1 Introduction

Text Mining (TM) has a wide range of applications that require differentiated processing of documents of various natures [1]. Overall, the goal is to be able to recognise and contextualise information of relevance, notably named entities and relationships among them. Language knowledge plays a key role characterising meaningful elements in sentence composition, such as nouns and verbs. Domain implementation implies to be generally familiar with the written language and specifically aware of the terminology and "writing structure" employed in the context under analysis. For example, TM practitioners of written English are required to learn about the structure of scientific papers, and the specificities of the terminology used, in order to apply TM methods and algorithms to biomedical research documents.

J. Sáez-Rodríguez et al. (eds.), *8th International Conference on Practical Appl. of Comput. Biol. & Bioinform. (PACBB 2014)*, Advances in Intelligent Systems and Computing 294,
DOI: 10.1007/978-3-319-07581-5_32,

Ontologies and controlled vocabularies are crucial in capturing the semantics of a domain, and machine learning models have proven successful in employing these resources to automatically recognise and extract information of interest. Currently, there are many commercial and free TM frameworks and software tools available. Apache Unstructured Information Management Architecture (UIMA) [2] and General Architecture for Text Engineering (GATE) [3] are two meaningful examples of open source initiatives. Apart from the natural language processors and machine learning recognisers, the most sophisticated components of TM tools are the document annotator and the document viewer. Typically, user-system interaction relies on these components and therefore, attractiveness, intuitiveness, ergonomics and flexibility are major development directives. UIMA's U-Compare [4] and GATE's Teamware [5], as others alike, are offered as an integrated framework option. Solutions not bound to TM frameworks also exist. For example, MyMiner [6], EGAS [7] and PubTator [8] offer free Web-based solutions, benefiting from feedback on user experience collected at jamborees and annotation evaluations. Arguably, the data staging area is the component of the annotation life-cycle less developed so far. Namely, existing tools come short in features such as: monitoring intra-annotator and inter-annotator annotation patterns, assessing the suitability of annotation guidelines, and identifying unanticipated semantics, or other annotation issues, while still conducting annotation rounds. These features are equally important to large-scale annotation projects and smaller, more application-specific projects. Notably, they are quite important when the annotators involved in the project present different levels of domain expertise and/or are not so familiar with the concept and implications of document annotation.

This paper presents Marky, a freely accessible Web-based annotation tool that aims to provide for customised document annotation while supporting project management. Notably, the novelty lays on the annotation tracking system, which ensures that all actions occurring within the annotation project are recorded and may be reverted at any point. This ability is crucial to assess inter-annotator agreement and observe intra-annotator patterns and thus, this tracking system is expected to improve the overall quality of project's results.

The next sections detail Marky design and its main functionalities. Attention is called to the following key activities: the creation of annotation projects, which involves the definition of the entities of interest and the main guidelines of annotation; the deployment of annotation rounds, which includes intra-annotator and inter-annotator statistics analysis; and the use of the annotation tracking system.

2 Marky Web Application

Marky is a Web-based multi-purpose document annotation application. The application was developed using the CakePHP framework (http://cakephp.org/) [9], which follows the Model–View–Controller (MVC) software pattern. Crafting application tasks into separate models, views, and controllers has made Marky lightweight, maintainable and modular. Notably, the modular design separates back-end development (e.g. the inclusion of natural language tools) from front-end development (e.g.

documents and annotations visual representation), and allows developers to make changes in one part of the application without affecting the others.

Marky reaches for state-of-the-art and free Web technologies to offer the best possible user experience and provide for efficient project management. The HTML5 (http://www.w3.org/TR/html5/) and CSS3 (http://www.css3.info/) technologies support the design of intuitive interfaces whereas Ajax and JQuery (http://jquery.com/) technologies account for user-system interaction, notably document traversal and manipulation, event handling, animation, and efficient use of the network layer. Additionally, the Rangy library (http://code.google.com/p/rangy/) is used in common DOM range and selection tasks to abstract from the different browser implementations of these functionalities (namely, Internet Explorer versus DOM-compliant browsers). MySQL database engine supports data management.

This section describes the annotation life-cycle and the core management and analysis functionalities currently provided by the application.

2.1 Project Life-Cycle

A project accounts for the following main components: documents or corpus, species or concepts of interest, annotations and users (administrator and annotators). The project administrator and the team of annotators have one shared goal: to carry out the work adequately to meet the project's objectives.

At initiation, the annotation goal of the project is established and the team is defined. The documents to be annotated are automatically retrieved from an online source (e.g. PubMed Central) or uploaded by the administrator. The concepts of interest, in particular the different types of concepts and their association, are identified manually (Fig. 1) and their semantics is formalised in a set of annotation guidelines.

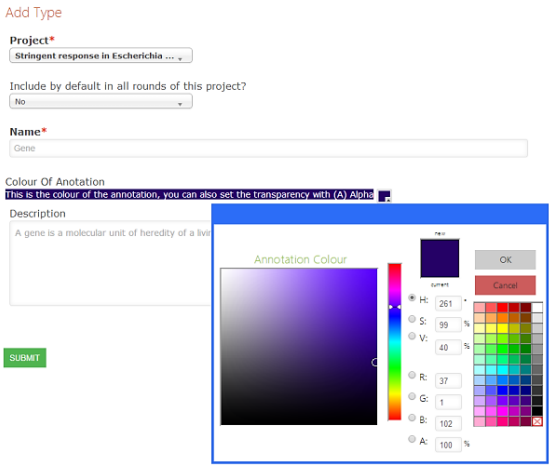

Fig. 1. Defining the types of concepts to be annotated in the project

This formalisation typically occurs after the preliminary meetings with annotators, and is meant to guide the annotation process, help annotators decide on (or disambiguate) textual mentions, and leverage annotators' domain expertise.

Often, the annotation project involves several annotators and is conducted in multiple rounds to guarantee the quality of the final annotations (Fig 2). Marky keeps track of the work done by each annotator at every round. Annotation rounds may prompt unanticipated issues, which may lead to changes in annotation guidelines, and even in the set of annotation types. Therefore, each round of annotation has associated its own set of guidelines and concept types.

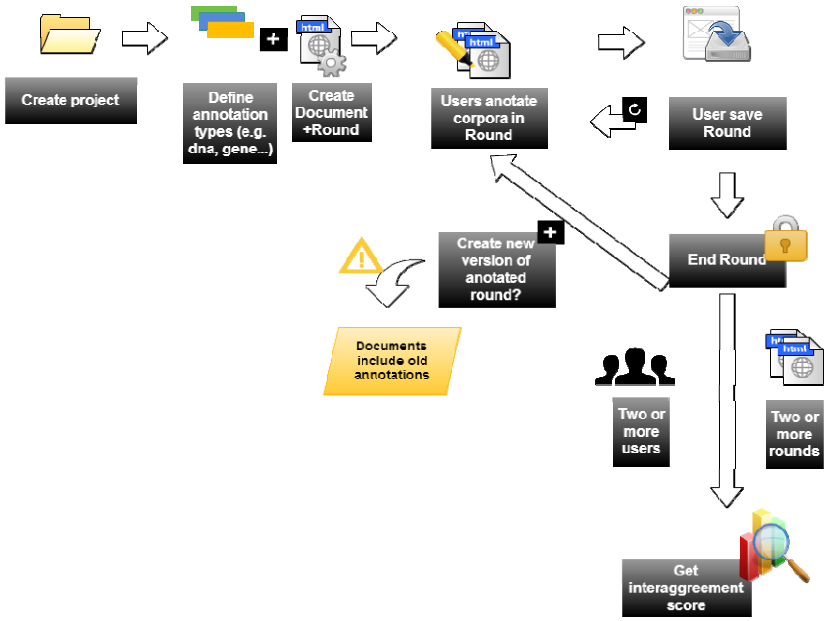

Fig. 2. The life-cycle of an annotation project in Marky

Round and user results are compared for the concept types in common. The improvement in the rates of agreement is quantified using the F-score, a common metric in IAA evaluations [10] presented below:

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

$$\mathrm{Precision} = \frac{\text{number of identical entities in set A and set B}}{\text{number of entities in set A}}$$

$$\mathrm{Recall} = \frac{\text{number of identical entities in set A and set B}}{\text{number of entities in set B}}$$

such that Set A refers to the set of annotations produced by annotator A and set B refers to the set of annotations produced by annotator B, and recall(set A, set B)=precision(set B, set A) [11].

2.2 Annotation Function

Marky offers an interactive interface allowing annotators to identify various kinds of concepts or entities within documents, according to task definition. The annotation component handles both plain text and HTML documents, and relies on state-of-the-art Web technologies, such as HTML5, CSS3, Ajax and JQuery, to offer an intuitive What-You-See-Is-What-You-Get editor.

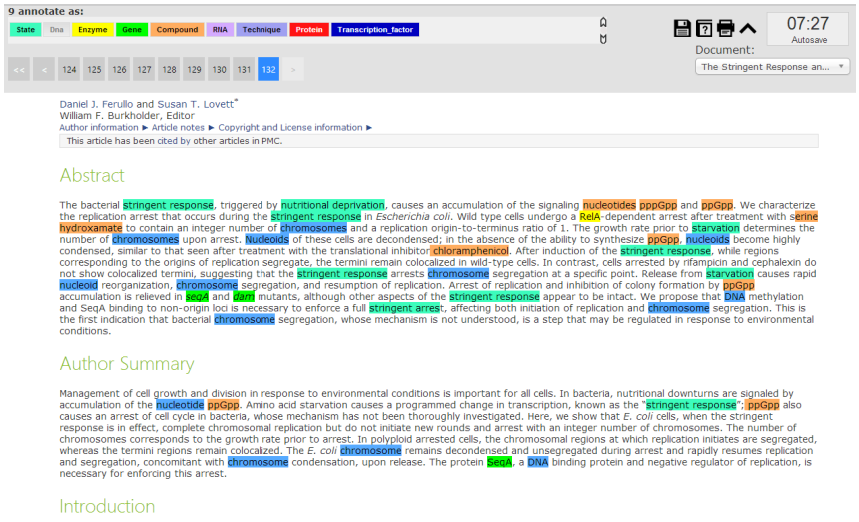

Fig. 3. Document annotation in action

At each round, the annotator has a list of documents to annotate. Documents are rendered in a Web form that supports term annotation as well as annotation visual presentation (Fig 3). By right-clicking on one or more words, the annotator marks a term or concept of interest (coloured in accordance to its type). While the annotation round is open, any annotation can be edited or removed. After the administrator closes the round, annotation statistics are calculated and round assessment is conducted, to evaluate the quality of the current version of annotations and decide upon launching a new round or not.

2.3 Annotation Tracking Function

By monitoring annotation changes, the project administrator may supervise the compliance of annotators with annotation guidelines and thus, adjust these guidelines and alert annotators about incorrect or dubious curation patterns. Notably, it is highly unlikely that two annotators annotate the very same text fragments, or completely agree on text-concept associations. Mostly, variability arises from differences in domain expertise, annotation skills, and interpretation of annotation criteria.

Per round, annotator statistics describe the volume of annotations achieved by the annotator regarding the different types of concepts in analysis (Fig 4). These data are useful to assess what concepts are most annotated and are rarely annotated by a given annotator, or within the round.

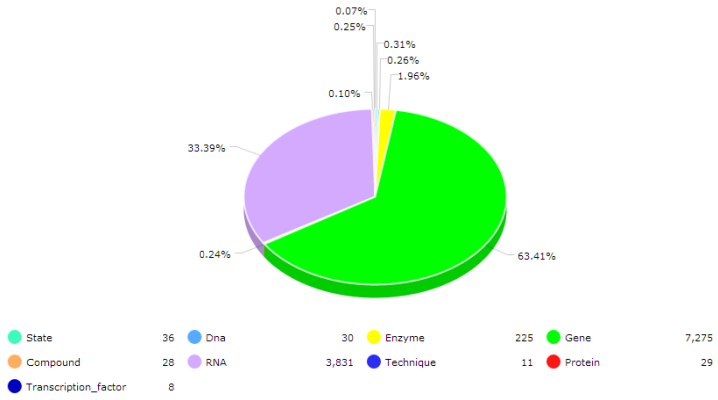

Fig. 4. Reporting annotator statistics in round

By assessing the agreement between annotators in the round, the project administrator evaluates the annotation discrepancies regarding the present set of guidelines (Fig 5). Typically, it is important to see whether certain concepts are being systematically missed or only some annotators are able to identify them. Moreover, annotators may not agree on term classification, i.e. terms are classified differently (and accounted in the statistics of different concept types), or term boundaries, i.e. annotations may only match partially.

Annotation guidelines may thus be revised so to contemplate any new semantics contexts and help to solve/minimise the annotation discrepancies observed. After deploying a new round, and besides analysing intra-round behaviour, the project administrator may compare the results between rounds to assess whether the revised set of guidelines was successful or new corrections/additions are still in need. Typically, the number of rounds performed depends considerably on time and cost constraints, leading the team to commit to a satisfactory score of agreement.

3 Conclusions

Marky is a free Web-based generic annotation tool that aims to provide highly customised annotation while supporting project management life-cycle. Indeed, Marky detaches from existing annotation tools in that it incorporates an annotation tracking system to monitor compliance with annotation guidelines and inter-annotator agreement. The ability to redo or undo annotations automatically is of help while consolidating annotation guidelines and minimises the manual work required from the annotators. Notably, the active monitoring of annotation patterns helps the administrator to "leverage" (at some extent) the expertise of the annotators by pin pointing interpretation/semantics issues that require further discussion and contextualisation.

Fig. 5. Reporting inter-annotator agreement

Regarding annotation functionalities, Marky design has favoured the use of state-of-the-art Web technologies as means to ensure wide user-system interaction and tool interoperability. Currently, Marky offers the same extent of manual operation of other tools. The ability to plug in named entity recognisers, or deploy the automatic recognition of dictionary entries will be sought after in the near future.

Acknowledgements. This work was supported by the IBB-CEB, the Fundação para a Ciência e Tecnologia (FCT) and the European Community fund FEDER, through Program COMPETE [FCT Project number PTDC/SAU-SAP/113196/2009/FCOMP-01-0124-FEDER-016012], 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 n° 316265, BIOCAPS. 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. Miner, G.: Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications. Academic Press (2012)
2. Ferrucci, D., Lally, A.: Building an example application with the Unstructured Information Management Architecture. IBM Syst. J. 43, 455–475 (2004)
3. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE, Version 6 (2011)
4. Kano, Y., Baumgartner, W.A., McCrohon, L., Ananiadou, S., Cohen, K.B., Hunter, L., Tsujii, J.: U-Compare: share and compare text mining tools with UIMA. Bioinformatics 25, 1997–1998 (2009)
5. Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., Gorrell, G.: GATE Teamware: A web-based, collaborative text annotation framework. Lang. Resour. Eval. 47, 1007–1029 (2013)
6. Salgado, D., Krallinger, M., Depaule, M., Drula, E., Tendulkar, A.V., Leitner, F., Valencia, A., Marcelle, C.: MyMiner: A web application for computer-assisted biocuration and text annotation. Bioinformatics 28, 2285–2287 (2012)
7. Campos, D., Lourenço, J., Nunes, T., Vitorino, R., Domingues, P.S.M., Oliveira, J.L.: Egas - Collaborative Biomedical Annotation as a Service. In: Fourth BioCreative Challenge Evaluation Workshop, pp. 254–259 (2013)
8. Wei, C.-H., Kao, H.-Y., Lu, Z.: PubTator: A web-based text mining tool for assisting biocuration. Nucleic Acids Res. 41, W518–22 (2013)
9. Iglesias, M.: CakePHP 1.3 Application Development Cookbook. Packt Publishing (2011)
10. Thompson, P., Iqbal, S.A., McNaught, J., Ananiadou, S.: Construction of an annotated corpus to support biomedical information extraction. BMC Bioinformatics 10, 349 (2009)
11. Brants, T.: Inter-annotator agreement for a German newspaper corpus. In: Second International Conference on Language Resources and Evaluation, LREC 2000 (2000)