The GATE Crowdsourcing Plugin: Crowdsourcing Annotated Corpora Made Easy

Kalina Bontcheva, Ian Roberts, Leon Derczynski, Dominic Rout
University of Sheffield
{kalina,ian,leon,d.rout}@dcs.shef.ac.uk

Abstract

Crowdsourcing is an increasingly popular, collaborative approach for acquiring annotated corpora. Despite this, reuse of corpus conversion tools and user interfaces between projects is still problematic, since these are not generally made available. This demonstration will introduce the new, open-source GATE Crowdsourcing plugin, which offers infrastructural support for mapping documents to crowdsourcing units and back, as well as automatically generating reusable crowdsourcing interfaces for NLP classification and selection tasks. The entire workflow will be demonstrated on: annotating named entities; disambiguating words and named entities with respect to DBpedia URIs; annotation of opinion holders and targets; and sentiment.

1 Introduction

Annotation science (Hovy, 2010; Stede and Huang, 2012) and general purpose corpus annotation tools (e.g. Bontcheva et al. (2013)) have evolved in response to the need for creating high-quality NLP corpora. Crowdsourcing is a popular collaborative approach that has been applied to acquiring annotated corpora and a wide range of other linguistic resources (Callison-Burch and Dredze, 2010; Fort et al., 2011; Wang et al., 2012). Although the use of this approach is intensifying, especially paid-for crowdsourcing, the reuse of annotation guidelines, task designs, and user interfaces between projects is still problematic, since these are generally not made available, despite their important role in result quality (Khanna et al., 2010).

A big outstanding challenge for crowdsourcing projects is that the cost to define a single annotation task remains quite substantial. This demonstration will introduce the new, open-source GATE Crowdsourcing plugin, which offers infrastructural support for mapping documents to crowdsourcing units, as well as automatically generated, reusable user interfaces[1] for NLP classification and selection tasks. Their use will be demonstrated on annotating named entities (selection task), disambiguating words and named entities with respect to DBpedia URIs (classification task), annotation of opinion holders and targets (selection task), as well as sentiment (classification task).

2 Crowdsourcing Stages and the Role of Infrastructural Support

Conceptually, the process of crowdsourcing annotated corpora can be broken down into four main stages, within which there are a number of largely infrastructural steps. In particular, data preparation and transformation into CrowdFlower units, creation of the annotation UI, creation and upload of gold units for quality control, and finally mapping judgements back into documents and aggregating all judgements into a finished corpus.

The rest of this section discusses in more detail where reusable components and infrastructural support for automatic data mapping and user interface generation are necessary, in order to reduce the overhead of crowdsourcing NLP corpora.

2.1 Project Definition

An important part of project definition is the mapping of the NLP problem into one or more crowdsourcing tasks, which are sufficiently simple to be carried out by non-experts and with a good quality. What are helpful here are reusable patterns for how best to crowdsource different kinds of NLP corpora. The GATE Crowdsourcing plugin

[1] Currently for CrowdFlower, which unlike Amazon Mechanical Turk is available globally.

currently provides such patterns for selection and classification tasks.

This stage also focuses on setup of the task parameters (e.g. number of crowd workers per task, payment per task) and piloting the project, in order to tune in its design. With respect to task parameters, infrastructural support is helpful, in order to enable automatic splitting of longer documents across crowdsourcing tasks.

2.2 Data Preparation

This stage, in particular, can benefit significantly from infrastructural support and reusable components, in order to collect the data (e.g. crawl the web, download samples from Twitter), preprocess it with linguistic tools (e.g. tokenisation, POS tagging, entity recognition), and then map automatically from documents and sentences to crowdsourcing micro-tasks.

2.3 Running the Crowdsourcing Project

This is the main phase of each crowdsourcing project. It consists of three kinds of tasks: task workflow and management, contributor management (including profiling and retention), and quality control. Paid-for marketplaces like Amazon Mechanical Turk and CrowdFlower already provide this support. As with conventional corpus annotation, quality control is particularly challenging, and additional NLP-specific infrastructural support can help.

2.4 Data Evaluation and Aggregation

In this phase, additional NLP-specific, infrastructural support is needed for evaluating and aggregating the multiple contributor inputs into a complete linguistic resource, and in assessing the resulting overall quality.

Next we demonstrate how these challenges have been addressed in our work.

3 The GATE Crowdsourcing Plugin

To address these NLP-specific requirements, we implemented a generic, open-source GATE Crowdsourcing plugin, which makes it very easy to set up and conduct crowdsourcing-based corpus annotation from within GATE's visual interface.

3.1 Physical representation for documents and annotations

Documents and their annotations are encoded in the GATE stand-off XML format (Cunningham et al., 2002), which was chosen for its support for overlapping annotations and the wide range of automatic pre-processing tools available. GATE also has support for the XCES standard (Ide et al., 2000) and others (e.g. CoNLL) if preferred. Annotations are grouped in separate annotation sets: one for the automatically pre-annotated annotations, one for the crowdsourced judgements, and a consensus set, which can be considered as the final resulting corpus annotation layer. In this way, provenance is fully tracked, which makes it possible to experiment with methods that consider more than one answer as potentially correct.

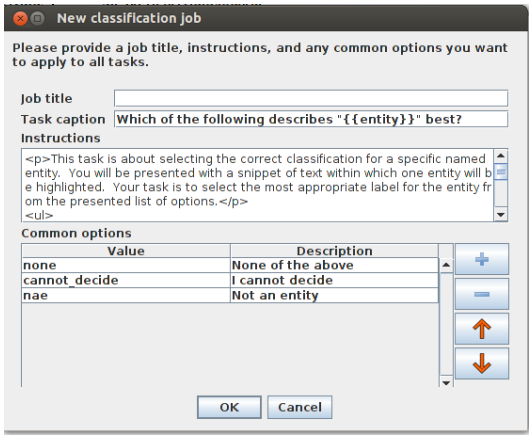

Figure 1: Classification UI Configuration

3.2 Automatic data mapping to CrowdFlower

The plugin expects documents to be pre-segmented into paragraphs, sentences and word tokens, using a tokeniser, POS tagger, and sentence splitter – e.g. those built in to GATE (Cunningham et al., 2002). The GATE Crowdsourcing plugin allows choice between these of which to use as the crowdsourcing task unit; e.g., to show one sentence per unit or one paragraph. In the demonstration we will show both automatic mapping at sentence level (for named entity annotation) and at paragraph level (for named entity disambiguation).

3.3 Automatic user interface generation

The User Interfaces (UIs) applicable to various task types tend to fall into a set of categories, the most commonly used being categorisation, selection, and text input. The GATE Crowdsourcing plugin provides generalised and re-usable, automatically generated interfaces for categorisation

Figure 2: Classification Interface: Sense Disambiguation Example

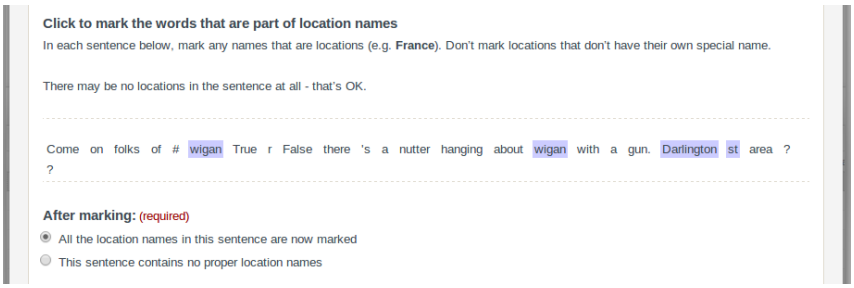

Figure 3: Sequential Selection Interface: Named Entity Recognition Example

and selection.

In the first step, task name, instructions, and classification choices are provided, in a UI configuration dialog (see Figure 1). In this example, the instructions are for disambiguating named entities. We have configured three fixed choices, which apply to each entity classification task.

For some categorisation NLP annotation tasks (e.g. classifying sentiment in tweets into positive, negative, and neutral), fixed categories are sufficient. In others, where the available category choices depend on the text that is being classified (e.g. the possible disambiguations of Paris are different from those of London), choices are defined through annotations on each of the classification targets. In this case case, the UI generator then takes these annotations as a parameter and automatically creates the different category choices, specific to each crowdsourcing unit. Figure 2 shows an example for sense disambiguation, which combines two unit-specific classes with the three fixed classification categories shown before.

Figure 3 shows the CrowdFlower-based user interface for word-constrained sequential selection, which in this case is parameterised for named entity annotation. In sequential selection, sub-units are defined in the UI configuration – tokens, for this example. The annotators are instructed to click on all words that constitute the desired sequence (the annotation guidelines are given as a parameter during the automatic user interface generation).

Since the text may not contain a sequence to be annotated, we also generate an explicit confirmation checkbox. This forces annotators to declare that they have made the selection or there is nothing to be selected in this text. CrowdFlower can then use gold units and test the correctness of the selections, even in cases where no sequences are selected in the text. In addition, requiring at least some worker interaction and decision-making in every task improves overall result quality.

3.4 Quality control

The key mechanism for spam prevention and quality control in CrowdFlower is test data, which we also refer to as gold units. These are completed examples which are mixed in with the unprocessed data shown to workers, and used to evaluate worker performance. The GATE Crowdsourcing plugin supports automatic creation of gold units from GATE annotations having a feature `correct`. The value of that feature is then taken to be the answer expected from the human annotator. Gold units need to be 10%–30% of the units to be annotated. The minimum performance threshold for workers can be set in the job configuration.

3.5 Automatic data import from CrowdFlower and adjudication

On completion, the plugin automatically imports collected multiple judgements back into GATE

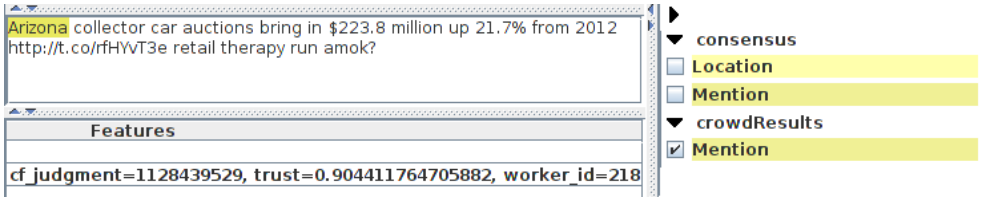

Figure 4: CrowdFlower Judgements in GATE

and the original documents are enriched with the crowdsourced information, modelled as multiple annotations (one per contributor). Figure 4 shows judgements that have been imported from CrowdFlower and stored as annotations on the original document. One useful feature is the trust metric, assigned by CrowdFlower for this judgement.

GATE's existing tools for calculating inter-annotator agreement and for corpus analysis are used to gain further insights into the quality of the collected information. If manual adjudication is required, GATE's existing annotations stack editor is used to show in parallel the annotations imported from CrowdFlower, so that differences in judgement can easily be seen and resolved. Alternatively, automatic adjudication via majority vote or other more sophisticated strategies can be implemented in GATE as components.

4 Conclusion

This paper described the GATE Crowdsourcing plugin[2] and the reusable components that it provides for automatic mapping of corpora to micro-tasks and vice versa, as well as the generic sequence selection and classification user interfaces. These are easily configurable for a wide range of NLP corpus annotation tasks and, as part of this demonstration, several example crowdsourcing projects will be shown.

Future work will focus on expanding the number of reusable components, the implementation of reusable automatic adjudication algorithms, and providing support for crowdsourcing through games-with-a-purpose (GWAPs).

Acknowledgments This was part of the uComp project (www.ucomp.eu). uComp receives the funding support of EPSRC EP/K017896/1, FWF 1097-N23, and ANR-12-CHRI-0003-03, in the framework of the CHIST-ERA ERA-NET.

[2]It is available to download from http://gate.ac.uk/ .

References

Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus. Roberts, Valentin. Tablan, Niraj Aswani, and Genevieve Gorrell. 2013. GATE Teamware: A Web-based, Collaborative Text Annotation Framework. *Language Resources and Evaluation*, 47:1007—1029.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: an Architecture for Development of Robust HLT Applications. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 7–12 July 2002*, ACL '02, pages 168–175, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karen Fort, Gilles Adda, and K. Bretonnel Cohen. 2011. Amazon mechanical turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413 –420.

Eduard Hovy. 2010. Annotation. In *Tutorial Abstracts of ACL*.

N. Ide, P. Bonhomme, and L. Romary. 2000. XCES: An XML-based Standard for Linguistic Corpora. In *Proceedings of the second International Conference on Language Resources and Evaluation (LREC 2000), 30 May – 2 Jun 2000*, pages 825–830, Athens, Greece.

Shashank Khanna, Aishwarya Ratan, James Davis, and William Thies. 2010. Evaluating and improving the usability of Mechanical Turk for low-income workers in India. In *Proceedings of the first ACM symposium on computing for development*. ACM.

Manfred Stede and Chu-Ren Huang. 2012. Interoperability and reusability: the science of annotation. *Language Resources and Evaluation*, 46:91–94. 10.1007/s10579-011-9164-x.

A. Wang, C.D.V. Hoang, and M. Y. Kan. 2012. Perspectives on Crowdsourcing Annotations for Natural Language Processing. *Language Resources and Evaluation*, Mar:1–23.