tagtog: collaborative interactive semi-supervised learning and annotation web-based framework

Juan Miguel Cejuela, Burkhard Rost

December 27, 2011

Abstract

In this document we describe our system tagtog, currently under development at Technical University of Munich. The system assists annotators/curators by 1) organizing and leveraging collaborative annotation efforts between members within a group and between groups; the system is a web-based platform and data is stored in our servers which facilitates the sharing and use of this, 2) providing an interactive learning and annotation framework, where iteratively human annotators are assisted by text mining annotators and text mining annotators are assisted by humans, 3) using semi-supervised learning algorithms which exploit the vast amount of existing unannotated data to improve the underlying text mining tools and so reduce human effort; together with a continuous (prioritized/active) feedback, curators can validate automatic annotations, 4) boasting diverse evaluation and correction measures such as inter-annotator agreement or detection of annotation inconsistencies.

1 Introduction

We are currently developing an annotation and learning framework that has 2 primary goals: 1) making it simple for curator groups to work together and share results, and so split the workload, and 2) using semi-supervised learning algorithms that can drastically reduce human efforts by means of automatic annotation and continuous mutual assistance between humans and machine methods.
Automatic annotations are prioritized — being more important those samples that the machine is less confident about — to ask the human user for feedback, within an active learning framework, [1, 2].

Our system is a web-based platform, see Figure 1. The owner initiates an annotation project and invite other annotators to work on the same task. The owner or in any case those users with a higher role define the task by indicating or providing the data to be curated. The raw and annotated data is stored in our systems so that all collaborators can easily use it and share it. Annotations can be downloaded at any time. As a web-page, users don’t have to install any specific software other than a web explorer which in particular allows to work on mobile devices such as tablets.

Figure 1: Preview of tagtog. Here shown the feedback querying and confirmation process. Abstract example taken from the PIR corpus, [3].

At an early stage of development, the system is restricted to (mono) entity labeling. In this initial stage we are focusing on the usability of the platform and on the implementation and application of semi-supervised algorithms and their interaction with human annotators. tagtog can process raw text or XML documents provided the user indicates the pertinent XML schema and sections/tags of interest. In this regard the system can work with any document structure, in particular abstracts and full-text articles.
We aim for an a priori domain-oblivious system. The current workflow is as follows:

1. The user uploads a set of documents to annotate.
2. Provides some few manual annotations.
3. With this a machine learning method is (self-)trained on the whole set and then prioritizes automatic annotations to show and ask feedback for.
4. The user corrects and confirms a group of important annotations, those which provide most information gain. The granularity of these questions may concern whole abstracts, sentences, or even subsequences (group of tokens), to be determined by the system.
5. 2-4 steps are iteratively repeated until the machine learning method reaches convergence or the user so desires it. The user can of course decide at any time to only work in manual mode.

Note that multiple members can work at different terminals at the same time which effectively splits the work load since then convergence will happen sooner. The system may or may not ask the same questions to users, perhaps looking for further validation or individual user’s evaluation.

As a result the system trains—or rather users train—a machine learning method that can be later reused for the same or similar projects. Internally our annotators comply with the UIMA interface, [4], and in the future we may accept from users existing machine annotators as plug-ins. These methods are effectively domain-specific components. More domain-specific features can be employed by providing dictionaries in files with a simple format of one entry per line. For instance, a dictionary of organisms or chemical terms.

Our system already boast a tagger of protein/gene names which can rival with current best systems, results to be shown for the workshop evaluation by March 1 2012. The tagger can be used right at the beginning to speed up the learning process. For the workshop we want to evaluate our system in the task of identifying protein/gene names 1) using the already-trained tagger and 2) only relying on the semi-supervised active learning architecture.

Furthermore the system can also report the over-all inner-annotator agreement which establishes a benchmark or point of reference to compare against our system. tagtog could also detect, ask, and correct possible inconsistencies such as the same terms sometimes being labeled and sometimes not. If this was in fact not an inconsistency but rather arose from ambiguity of natural language, so told by the user, the system could specially mark these cases. Users can reason and explain their annotations by writing notes to the system.
2 Evaluation

Curators will be given a list of 25 + 25 PMIDs, unannotated recent articles with both title and abstract directly accessible from pubmed, and they will have to identify and label protein/gene names in these 2 sections. Final annotations must follow the format of the Yapex corpus, [5], a XML file where every pubmed entry has its PMID, title, and abstract, and where protein/gene names are labeled in-lined with the tag `<Protname>`, see Figure 2. Note that the title and the abstract are to be written exactly as they appear in pubmed just with the in-line tags, without extra new lines or other spacing (the introduced new lines in this example are only for editing purposes)

```xml
<PubmedArticleSet>
  <PubmedArticle>
    <PMID>11401507</PMID>
    <ArticleTitle>Molecular dissection of the <Protname>importin beta1</Protname>-recognized nuclear targeting signal of <Protname>parathyroid hormone-related protein</Protname>.</ArticleTitle>
    <AbstractText>...</AbstractText>
  </PubmedArticle>
</PubmedArticleSet>
```

Figure 2: Yapex XML in-line annotation sample format. Protein/gene names are marked with the tag `<Protname>`.

The exact labeling and separation of protein names is always debatable illustrated by the divergence of current gold corpora, [3, 6, 7, 8, 9]. As a simple base guideline we advise not to include spaces surrounding a protein name, i.e. `human <Protname>importin beta1</Protname> derivatives and not human<Protname> importin beta1 </Protname> derivatives`. We recommend not including the organism in the name, in this case `human`. We also recommend combining protein names and gene names and annotating the identifier of a protein-coding gene like `<Protname>p53</Protname>` mRNA. We advise to separate abbreviations like in `<Protname>ribosomal protein L22</Protname>` (`<Protname>rpl22</Protname>`) in any case we encourage the group of biocurators that will annotate the given document set to define or choose their own guidelines and to formalize them as written rules.

We would like to be assigned with a group of at least 2 curators so that the system’s social capabilities can be tested.
The process for the evaluation will be as follows:

1. **Manual Task**: curators manually annotate the first subset of 25 pubmed entries and produce, output, one XML file for every annotated entry following the Yapex in-line format. That is, every entry must be in its own file which exactly contains 1 `<PubmedArticleSet>` and 1 `<PubmedArticle>` with its `<PMID>`, `<ArticleTitle>`, and `<AbstractText>`, and the protein tags. We recommend using the extension “gold.xml” for these files.

2. **Assisted Task with tagtog**: a group of at least 2 members will:
   
   (a) start a new annotation project that will use from the beginning the provided protein/gene name tagger to annotate the second subset of 25 entries. We encourage curators to work at different terminals at the same time. The interactive learning process, machine querying and human feedback, will start until the system reaches convergence. The semi-automatically annotated data can then be retrieved for evaluation purposes. Later every user and independently will verify and correct if necessary the system’s output. The system will report inner-annotator agreement and in case of differences will ask the owner for a consensus. Finally the considered gold standard can be downloaded.

   (b) start a new second annotation project this time not using the already-trained tagger and so only relying on the semi-supervised active learning architecture. Same steps as with the first project will follow.

Additionally —to be approved by the workshop organizers—and to accelerate the learning process, curators could give right at the beginning the first manually-labeled set as training data.

The system can report precision, recall, and F1 score for 2 standard benchmarks: 1) **rigid performance**, exact entity matches —same token boundaries— between the gold document and the prediction will be counted as true positives, partial matches are counted as both false negatives and false positives (which worsens both precision and recall), and the rest as customary, false negatives when entity in gold but not in prediction, false positives when entity in prediction but not in gold, 2) **soft performance**, analogous except that partial matches count as true positives. Since the first benchmark may be too negative and the second one too positive, the arithmetic mean from these is also given.

For the evaluation at the workshop we are specially interested in: 1) assessing the performance of our already-trained tagger, 2) assessing the performance
of our semi-supervised active learning methods against the given tagger, 3) validating the system’s usability with respect to final real users in a real scenario.

References


