# Development of Visual Tagging Tool

Chris J. Lu, Ph.D.[1,2], Guy Divita[1,2], Allen C. Browne [1]
[1]National Library of Medicine, Bethesda, MD; [2]Lockheed Martin/MSD, Bethesda, MD

## Abstract

*The Visual Tagging Tool (VTT), developed at the National Library of Medicine (NLM), is a simple, lightweight, portable, Java Swing based annotation tool. It is designed to easily mark up text. The tool is intuitive for those with pc-based skills. The tool reads and writes a file format that is easily replicated by other applications. This allows for the visual representation and correction for applications such as Part of Speech (POS) taggers. The tool is distributed by NLM via an Open Source License agreement.*

## 1. Introduction

Tagging and annotation are critical to the success of complex NLP applications. A generic GUI (Graphical User Interface) tagging tool that allows users to easily mark up text and visualize the tagged text is need. VTT is designed and implemented as a simple and intuitive generic tagging tool with a variety of useful features.

## 2. VTT File Format

VTT file format incorporates a stand-off representation of the text and a set of additional sections delimited by pre-defined section headers. The original text is preserved as-is in a section of the file. The markup section refers to positions with the text. This allows for the possibility for overlapping markups or tags. The markup section of the file includes one line records with piped delimited fields indicating the beginning offset, length, tag name, subcategory, an optional annotation field, and the text contained in the markup. Additional fields may be present and are ignored for display purposes. Each VTT file also includes the tagset definition. VTT requires no additional resources to display or modify a file. Additional fields may be present and are ignored for display purposes. Another section of the VTT file keeps track of modification dates, annotators, and version of the software.

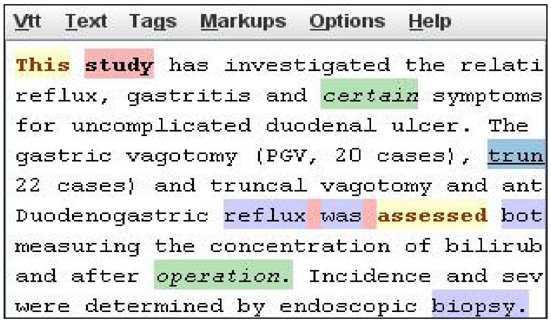
## 3. Tagset Criteria

VTT is designed to be simple. To this end tags can have one or two levels of hierarchy via subtags. Relationships cannot be tagged in VTT. Other more complex tools are available for such purposes [1-2].

## 4. Visualization

VTT is a GUI tool that displays tagged text in different visual styles (colors, fonts, sizes, etc.). Users can select text using the mouse and select a tag using a pull down menu of existing tags. Quick keys (mapped to keys 0-9) may be set to particular tags or tag-subtag sets.

VTT provides generic GUI features, such as print VTT files; define tags by GUI dialog boxes; provide redo and undo features on Markup operations; zoom in/out; find a term; etc. A complete user tutorial is included. Figure-1 shows an example of VTT screen.



This study has investigated the relati
reflux, gastritis and certain symptoms
for uncomplicated duodenal ulcer. The
gastric vagotomy (PGV, 20 cases), trun
22 cases) and truncal vagotomy and ant
Duodenogastric reflux was assessed bot
measuring the concentration of bilirub
and after operation. Incidence and sev
were determined by endoscopic biopsy.

**Figure- 1: VTT Example**

## 5. Continued Development and Future Work

VTT includes facilities to compare two versions of the same file. There is the intension to expand the evaluation capabilities to compare one corpus to another, gather statistics and drill down through instances of disagreement.

## 6. Applications and Conclusion

VTT has been used at NLM as an output format for the dTagger, and for PHI (Protected Health Information) de-identification project. Within the de-identification project, non-computer professionals became proficient with minimal training. The VTT is available at URL: http://specialist.nlm.nih.gov/vtt.

## References

1. Cunningham H, et al. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. Proceedings of the 40th Anniversary Meeting of ACL'02. Philadelphia, July 2002

2. Ogren, PV. Knowtator: A Protégé plug-in for annotated corpus construction. Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. New York, New York, 2006, pp. 273-27