Open Text Annotators Using Apache UIMA

William Daniel Colen de Moura Silva*[1], Marcelo Finger[1], and Carlos Eduardo Dantas de Menezes[2]

[1] Departamento de Ciência da Computação, IME/USP, Rua do Matão 1010, CEP 055080-090, São Paulo SP, Brasil

[2] Faculdade de Tecnologia e Ciências Exatas, Universidade São Judas Tadeu, Rua Taquari 546, CEP 03166-000, São Paulo SP, Brasil

Abstract. This paper describes a Portuguese language grammar checker project, called CoGrOO – *Corretor Gramatical para o OpenOffice.org* (Grammar Checker for OpenOffice.org) – and the ongoing work of migrating its annotators, including Sentence Detector, Tokenizer, POS Tagger, Chunker and Shallow Parser, to the OASIS Standard platform UIMA – Unstructured Information Management Architecture. All CoGrOO UIMA annotators will be made publicly available under an open source software license.

1 Introduction

OpenOffice.org is a multiplatform and multilingual office suite, and an open source project[3]. OpenOffice.org did not possess grammar checking functionality, which made it less competitive when compared with the commercial alternative. This motivated some NLP researchers sponsored by FINEP to create CoGrOO[4], the Brazilian Portuguese grammar checker. Work started in 2004 and since its first release in 2006 it has been adopted by important companies like Petrobras and Celepar, and accumulated over than fifty thousand downloads from its official website.

CoGrOO is an open source project. It is capable of identifying Portuguese errors like: pronoun placement; nominal agreement; subject-verb agreement; usage of grave accent (*crase*) employed to indicate the coalescence of preposition "a" (to) + definite feminine singular determiner "a", yielding "à"; nominal and verbal government; and other common errors of Brazilian Portuguese writing.

Although CoGrOO has won many users, its internal components can benefit from several improvements. Those improvements will allow each component (e.g. the sentence detector, tokenizer, PoS tagger, chunker, shallow parser) to function as independent modules available to the community in the form of free software.

This paper details CoGrOO improvements, especially the code refactoring initiative that will change the current architecture to an open one, making it easier to employ, extend and modify, possibly improving its grammar checking performance, and enabling its effective use in other open source projects.

* Scholarship holder CNPq – Brazil

[3] OpenOffice.org - Free and Open Productivity suite, `http://www.openoffice.org`

[4] CoGrOO official website, `http://cogroo.sourceforge.net`

2 The Current State of CoGrOO

CoGrOO system is composed by the following modules:

1. Sentence Boundary Detector: receives a text and splits' it into sentences;
2. Tokenizer: receives a sentence and splits it into words and punctuation marks;
3. Named Finder: receives the sentence tokens and identifies the potential proper nouns, such as person names, places and organization names;
4. Part-of-Speech Tagger: receives a sentence and assigns the most probable morphological tag to its lexical items, according to their context;
5. Chunker: receives a tagged text and finds some small noun phrases (NP) and small verbal phrases (VP);
6. Subject-Verb Finder: receives a tagged sentence with NPs and VPs and searches for the subject. If it is found, it marks the NP as a subject of a VP;
7. Grammar Error Detector: this module looks for grammar errors in the input sentence. It is activated after all the previous sentence analysis steps.

CoGrOO annotators are based on OpenNLP[5], a Natural Language Processing (NLP) library, in order to create a more consistent and robust software.

The NLP tools in OpenNLP were created for English, which were initially adapted for Portuguese. This involved changing the Maxent features[1] in several internal modules and generating new models using the Brazilian Portuguese resources.

2.1 Linguistic Resources

The CETENFOLHA corpus [2] was used to train the OpenNLP Maxent models. CETENFOLHA is a 24-million words Brazilian Portuguese PoS-tagged corpus, based on journalistic essays. It is not colloquial, generally written in third person.

To improve its performance, CoGrOO requires dealing with abbreviations. An abbreviation dictionary is employed, which contains entries like "sr.", "tel.", "apto.". This dictionary is especially important for the Sentence Boundary Detector and Tokenizer modules. This dictionary was built using Jspell[3].

Other lexical dictionaries are also part of the system, whose construction was based on several other dictionaries freely distributed provided their licenses were compatible with CoGrOO, such Jspell. Lexical dictionaries are required for Part-of-Speech tagging and to generate suggestions in the Grammar Error Detector module.

2.2 Annotators Accuracy

As CoGrOO is distributed as an OpenOffice.org extension, an important non functional requirement is its package size, that should be as smaller as possible.

[5] Open source framework to develop NLP applications, `http://opennlp.sf.net`

On the other hand, the size of the package is related to the dictionaries and models sizes. Reducing them can influence performance.

To achieve a good balance between package size and annotation precision, the CoGrOO development tools include a repetitive training mechanism which allows for the fine tuning of the number of sentences in the training set, which is controlled by a cutoff parameter (minimum number an event should occur to be included). If the cutoff gets too large or too small, the performance decreases, and if the cutoff increases, the package size decreases. So an optimal cutoff is searched.

The annotators were evaluated using 10-fold Cross Validation. For the current version, the performances of the annotators are:

- Tokenizer: precision 0.954, recall 0.975 – trained with 40000 sentences, cutoff 150;
- Name Finder: precision 0.941, recall 0.946 – trained with 450000 sentences, cutoff 250;
- Tagger: accuracy 0.961 – trained with 50000 sentences, cutoff 5;
- Chunker: accuracy 0.772 (sentences with all chunks correctly annotated) – trained with 20000 sentences, cutoff 150;
- Shallow Parser: accuracy 0.688 (sentences with all subjects and main verbs correctly annotated) – trained with 20000 sentences, cutoff 100.

3 Improvements Obtained

3.1 Open Architecture

Similar to other open source softwares, one of the most valuable assets of CoGrOO is its group of users. An important initiative is to gather users and contributors from industry and research community. They have knowledge to directly improve the code by adding new features, testing and fixing bugs.

To accomplish that, CoGrOO modules were reviewed to make them more readable and easily reusable. Aiming at reusability, the development team decided to adopt an open architecture model. Two options were investigated. The Gate[6] – General Architecture for Text Engineering – and Apache UIMA[7] – Unstructured information Management Architecture, Apache UIMA was chosen because it is an OASIS Standard.

The Apache UIMA project offers reference architecture for NLP systems and a framework to help its realization. It defines *SOFA* (Subject of Analysis) as any unstructured document, like text documents in natural language, voice records etc. SOFAs are attached to a *CAS* (Common Analysis Structure), which holds the context of the analysis, including the annotations provided by an annotator for that SOFA, and the Type System, which provides information about the annotations types and the input and output of the UIMA components.

[6] GATE – General Architecture for Text Engineering, `http://gate.ac.uk`
[7] Apache UIMA, `http://incubator.apache.org/uima`

An *annotator* analyzes documents and outputs annotations on the document's context. This annotator plus a set of metadata form an *AE* (Analysis Engine), that contains the framework-provided infrastructure that allows for the easy combination of an AE with other AEs in different flows and deployments.

CoGrOO annotators are mutually decoupled, so it will be easier to substitute one of its annotators by another one. It will also be easier for a user familiar to Apache UIMA to understand the CoGrOO implementation.

Each CoGrOO annotator was converted to an Analysis Engine, which involves defining the interface of each annotator and the input and output types, defining the resource dependencies and available configuration, developing the AE metadata; and modifying the code to support the architecture.

The initial release of the CoGrOO analyzers implementing the new architecture is planned for the first quarter of 2010. The code under development is already accessible from project SVN[8].

3.2 Other Ongoing Improvements

There is an ongoing work on implementing a syntactic parser for CoGrOO using probabilistic context-free grammar, employing smoothing algorithms that deal with the sparseness of training data. Initial release of the new parser is planned for the final quarter of 2010.

Another ongoing work is the adoption of the new Portuguese Language Orthographic Agreement[9], that means to update about 0.5% of the dictionary and patch error rules and Corpus.

4 Conclusion

The free open source CoGrOO grammar checker won popularity among users of the OpenOffice.org office suite, but the potential of its components has not been widely explored by the research community.

The move to an open architecture aims to change that by providing standardized annotators for Brazilian Portuguese, including a sentence detector, tokenizer, tagger, chunker and shallow parser.

References

1. Ratnaparkhi, A.: Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. Dissertation, University of Pennsylvania (1998)
2. CETENFolha Brazilian-Portuguese annotated corpus, `http://www.linguateca.pt/cetenfolha`, last access: October 06, 2009
3. Jspell - Projeto Natura, `http://natura.di.uminho.pt/wiki/doku.php?id=ferramentas:jspell`, last access: October 06, 2009

[8] CoGrOO SVN repository, `svn://ccsl.ime.usp.br/CoGrOO`

[9] Portuguese Language Orthographic Agreement, `http://www.bbc.co.uk/portuguese/noticias/2009/01/090122_reformaportuguesqandarw_tc2.shtml`