# Building a Corpus of South African English

Gareth Dwyer - garethdwyer@gmail.com
Supervisor: James Connan

February 2014

## 1 Background

The Dictionary Unit for South African English (DSAE) produces the South African Concise Oxford Dictionary, the authoritative reference for South African English. In order to track and analyze how South African English is being used, and to identify new South African words, they require a corpus system. This system should crawl a predefined list of websites, websites which include South African newspapers and blogs, and store article texts, comments, and metadata such as author and date to a database. The system furthermore needs to analyze the text – primarily to break it into tokens, to check the text against already existing South African word lists, and to flag possible new coinages of South African words, but also to be capable of running custom analysis as specified by lexicographers and researchers, such as analysis on word frequency and popularity.

## 2 Modules

Flexibility of the system must be a high priority, both because the data sources are constantly changing as the Internet evolves, and because the needs of lexicographers and researchers are not fixed. Therefore the system should be as modularized as possible. Below are listed the core modules which would make up the base system. Ideally, each of these modules should be entirely independent of the others, so that if further functionality is needed, or if requirements change, then modules can be added or rewritten without

breaking the entire system. This should furthermore facilitate the system to be adapted for other language and uses.

**Core Modules**

- Crawler – This will crawl the specified websites and save their content to a database. It will also do basic filtering, ignoring data such as PDFs and images.
- AjaxCrawler – This will crawl dynamically created content, especially comments on newspaper articles.
- Text classifier – This will do more advanced filtering, removing 404 Error pages and anything else which should not be added to the corpus.
- Text extractor – This will extract plain text from articles, removing HTML tags, Javascript, CSS and other 'noise', and tokenize the articles to build word frequency lists.
- Metadata extractor – this will work with the Text Extractor to identify metadata such as author, date, headline, type/genre of text, etc.
- Deduplicator – This will identify duplicate and near-duplicate texts in the database, and remove or mark these.
- Analyzer – This will run various analysis operations on the corpus, with user customizable filters for date, text type, etc.
- Interface – This will be a web application to allow users to see analysis results.
- Admin interface – This will allow the user to configure the system, such as the list of sites the crawler should crawl.

# 3 Literature

Texts can be divided broadly into three categories: Those which look at building corpora systems from both a Computer Science and Linguistic aspect, those which look at Corpora purely from a Linguistic aspect, and those

which look at language processing, etc., purely from a Computer Science aspect.

**Corpora systems**

- Guevara: NoWaC: A Large Web-based Corpus for Norwegian – This text describes the methods and technology used to build a corpus for Norwegian. (Guevara, 2010)

- Baroni et al.: The WaCky wide web: a collection of very large linguistically processed web-crawled corpora – This text describes how three corpora were created and one of the papers aims is for others to use it "to rapidly develop similar corpora for other languages"(Baroni *et al.*, 2009)

**Corpora**

- Stubbs: Text and Corpus Analysis – This text is a general introduction to corpora and includes examples of useful analysis to run.(Stubbs, 1996)

**Crawling, processing and storing data**

- Du: Data Mining Techniques and Applications – This text gives an overview of data mining and also includes algorithms such as the Boolean association rule used to analyze keywords and associations.(Du, 2010)

- Bird, Klein, and Loper: Natural Language Processing with Python – This text is about the Python Natural Language Tool Kit, which provides tokenizing and other text analysis tools.(Bird *et al.*, 2009)

- Mesbah: Crawling Ajax-Based Web Applications – This text explains how the open-source crawler 'Crawljax' was built, and how it can be used for scraping dynamically created web content. (Mesbah *et al.*, 2012)

# 4 Timeline

- First Semester: Complete all of the Core Modules, except for the Analyzer, Interface and Admin Interface, although basic versions of these

will be put in place to allow for proper feedback from the Dictionary Unit. Build departmental website for the project. Read works listed above as well as any other relevant texts, still to be discovered.

- Second Semester: Complete the Analyzer, and Interface modules as well as any additional modules, a need for which may be identified during the first semester. Complete write up and build external website for the project.

# References


Baroni, Marco, Bernardini, Silvia, Ferraresi, Adriano, & Zanchetta, Eros. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, **43**(3), 209–226.

Bird, S., Klein, E., & Loper, E. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Du, Hongbo. 2010. *Data Mining Techniques and Applications*. Cengage Learning EMEA.

Guevara, Emiliano. 2010. NoWaC: A Large Web-based Corpus for Norwegian. *Pages 1–7 of: Proceedings of the NAACL HLT 2010 Sixth Web As Corpus Workshop*. WAC-6 '10. Stroudsburg, PA, USA: Association for Computational Linguistics.

Mesbah, Ali, van Deursen, Arie, & Lenselink, Stefan. 2012. Crawling Ajax-Based Web Applications Through Dynamic Analysis of User Interface State Changes. *ACM Trans. Web*, **6**(1), 3:1–3:30.

Stubbs, M.a. 1996. *Text and Corpus Analysis: Computer Assisted Studies of Language and Culture*. Language in Society. Wiley.