xTAS: Text Analysis in a Timely Manner

Ork de Rooij
University of Amsterdam
Science Park 904
1098 XH Amsterdam, The Netherlands
orooij@uva.nl

Andrei Vishneuski
University of Amsterdam
Science Park 904
1098 XH Amsterdam, The Netherlands
a.vishneuski@uva.nl

Maarten de Rijke
University of Amsterdam
Science Park 904
1098 XH Amsterdam, The Netherlands
derijke@uva.nl

ABSTRACT

In this demonstration we present xTAS, an open source web-service developed at the University of Amsterdam which allows processing multi-lingual textual content of your documents in a timely manner. We showcase the architecture of xTAS, together with several demonstrators that use xTAS in their architecture.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text Analysis

General Terms

Algorithms, Performance, Experimentation

Keywords

text analysis, web service, distributed processing

1. INTRODUCTION

In this demonstration we present xTAS, a set of web services for processing textual content of your documents in a timely manner.

The purpose of xTAS is to allow users to perform a variety text processing tasks as fast as possible, without having to bother about databases, storage or result caching. Whether this task is simple counting of words, or more processing intensive named entity normalization across a large set of documents does not matter.

We designed xTAS to embed both existing (open source) analysis algorithms as well as proprietary UvA analysis algorithms in a scalable distributed architecture. The current list of analysis features include:

- Stemming and tokenization of documents.
- Part of Speech tagging/tokenization of documents using a variety of techniques, including TNT [6] and Stanford [7].
- Named Entity extraction using a variety of existing techniques, including TNT, LingPipe [1], LBJ [5] and Stanford [4].
- Aggregation of results over documents, for example counting words and entities to allow the generation of tag cloud summaries.

The set of analysis features is extendible. Adding an additional processing step to xTAS can be as simple as defining a (Python) function that accepts a document as input, and yields some sort of analysis result. xTAS will then take care of running your added plugin on the whole set.

1.1 Technical features

Users can communicate with xTAS by using it as a web service, or they can include it in their own applications as a (Python) library.

xTAS is designed with speed and easy of extendability in mind. As such, xTAS builds upon several existing open source packages, most notably: MongoDB [3] to store documents and results, Celery [2] to distribute analysis tasks between nodes. This allows xTAS to perform most of the necessary document processing distributed on demand, in order to minimize waiting time for users. See figure 1 for an overview of the architecture.

2. DEMONSTRATION

We will showcase xTAS during DIR2012, and provide instructions to users who want to use xTAS in their own applications. Besides this, we will also showcase several existing demos which use xTAS in their processing pipeline. This includes the following demos.

- For **Project Infiniti**,[1] a COMMIT project, we designed a demonstrator for exploratory search through a large collection of articles from several centuries worth of newspaper archives. The demonstrator showcases dynamic retrieval and processing of articles from newspapers the Koninklijke Bibliotheek, and uses xTAS to provide users with summary clouds of subsets of this archive based on their search need. See figure 2 for a screenshot.
- We designed a demonstrator as part of the **WAHSP** project[2] which provides query based summaries of the newspaper archive of the Koninklijke Bibliotheek.

[1] http://project-infiniti.nl/
[2] See http://wahsp.nl/

DIR2012 2012 Gent, Belgium

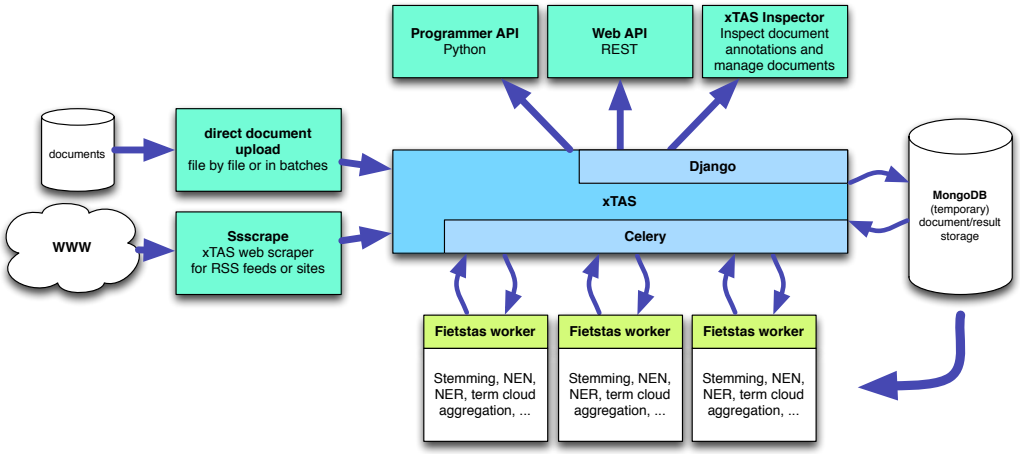

Figure 1: An overview of the xTAS infrastructure. Documents can be added to the system using either the programmer API, the REST interface, or by adding them in batches. Users then request types of processing which xTAS distributes accross the number of available worker nodes. Results can be returned as soon as they have been processed, or returned when the user requires them.

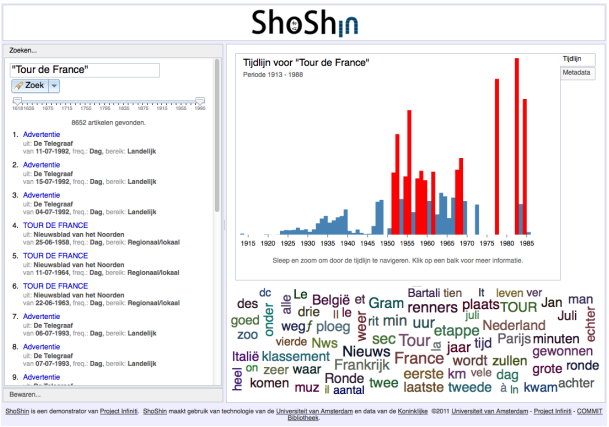

Figure 2: The Project Infiniti demonstrator, showing a word cloud and timeline view for a subset of the newspaper archive of the Koninklijke Bibliotheek.

- For **Peilend.nl** we designed a demonstrator to view sentiments in an ever increasing corpus of newspaper archives.

3. CONCLUSION

We present xTAS, a multi-user, multi-lingual text analysis service for large scale document analysis. Work in xTAS is ongoing, and currently progressing towards more analysis techniques, such as including sentiment analysis, named entity normalization and entity matching techniques.

4. ACKNOWLEDGEMENTS

This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.815, 640.004.802, 380-70-011, 727.-011.005, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP project funded by the CLARIN-nl program, under COMMIT project Infiniti and by the ESF Research Network Program ELIAS.

5. REFERENCES

[1] Alias-i. LingPipe 4.1.0. http://alias-i.com/lingpipe.

[2] Celery: Distributed Task Queue. http://celeryproject.org/.

[3] MongoDB. http://http://www.mongodb.org/.

[4] J. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. ACL, 2005.

[5] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. ACL, 2009.

[6] E. Tjong Kim Sang. Generating subtitles from linguistically annotated text. *Atranos report WP4-12, University of Antwerp*, 2003.

[7] K. Toutanova and C. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proc. of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 63–70. ACL, 2000.