Open Stylometric System *WebSty*: Towards Multilingual and Muiltipurpose Workbench

Maciej Piasecki — Wrocław University of Science and Technology
Tomasz Walkowiak — Wrocław University of Science and Technology
Maciej Eder — Inst. of Polish Language PAS and Pedagogical Univ. of Kraków
{maciej.piasecki,tomasz.walkowiak}@pwr.edu.pl,
maciej.eder@ijp-pan.krakow.pl

Abstract

WebSty is an open, web-based stylometric system designed for SS&H users. It was designed according to the CLARIN philosophy: no need for installation, minimised requirements on the users' technical skills and knowledge, focus on SS&H tasks. In the paper we present its latest extension with several visualisation methods, techniques for the extraction of characteristic features, and support for two more languages, namely English and German.

1. Introduction

Stylometry is associated with analysis of language features extracted from texts and aimed at tracing similarities between texts. It is used to identify groups of texts that exhibit subtle similarities hidden to the naked eye but traceable by multidimensional statistical techniques. A classical type of analysis are authorship attribution, or an experimental setup where an anonymous (or disputed) texts are compared against a set of texts of known authorship, in order to identify the nearest neighbourship relations (Stamatatos 2009). In SS&H text analysis is becoming an interesting methodological proposition to assess textual similarities beyond authorship. In the study of literature, one might be interested in distant reading techniques to pinpoint genre characteristics, literary period, intertextuality, etc. In sociology, one might want to analyse textual biases in press materials from different press agencies, in psychology one might trace change of style as a function of age, or correlations between text and mental diseases (Le et al. 2011).

Application of the stylometric methods can be difficult for SS&H researchers, mostly because the combination of the variety of data formats, language tools and data analysis tools is not straightforward, but also application of the tools usually requires specialised knowledge and technical skills. Moreover, the entire NLP workflow is controlled by a large number of hyperparameters whose influence on the overall results of the stylometric analysis is complex.

WebSty[1] is an open stylometric system with web-based user interface designed to be used without any installation, and which offers a variety of dedicated language processing tools, provides ready to use processing chains, and assists users in setting up the processing parameters. It was initially focused on processing texts in Polish and offered a limited number of visualisation and data analysis methods. Below we present an new version which was expanded with a more flexible and efficient processing architecture, several visualisation methods and techniques for the extraction of characteristic features. The modular architecture of WebSty enabled adding support for more languages, namely English and German, in a relatively easy way.

2. Related Work

In spite of the long tradition of stylometry there is only a limited number of online systems. The well known *Voyant*[2] is a online tool for limited statistical analysis of texts supplemented with a good GUI and several visualisation methods. A range of NLP tools was added on the a basis of the Stanford CoreNLP, e.g. PN recognition. However, the functionality of Voyant is based mostly on tracing word forms and their relative frequencies across text and limited to English. Only simple statistical measures: tf.idf and Z-score are available to compare word forms vs. documents. Popular *Stylo* (Eder

[1] http://websty.clarin-pl.eu

[2] Voyant: http://docs.voyant-tools.org, CoreNLP: https://nlp.stanford.edu/software/, Mallet: http://mallet.cs.umass.edu

et al. 2016) is a library in the R programming language for different stylometric tasks. It is designed to analyse shallow morphological features (function words and letter n-grams) harvested from the locally stored plain text files, but it can also be used to analyse preprocessed corpora. The package offers both selected exploratory methods, and supervised Machine Learning (ML) algorithms. It needs to be locally installed. *Mallet* is an off-line document classification system working on the basis of machine learning but it is mostly used for topic analysis.

In addition, we can find on the Web a couple of simple online applications for authorship attribution[3], e.g. *Signature* (only word-level features) and *AICBT* (limited number of feature types for English). There is a number of off-line applications, like *JGAAP* (an entire processing workflow), *JStylo* (rich set of feature types, recognition of obfuscation), and *StyleTool* (quite rudimentary). Neither of the discussed systems support parallel processing of large amounts of data, nor they use multiple language tools and processing methods, and advanced extraction of characteristic features.

3. Language Processing Architecture

A multi-user, web-based system generates problems related to the system availability and performance. The system should be *scalable*, *responsive* and *available* all the time. Language tools (LTs) have excessive CPU/memory consumption. Needless to say, the number of users and/or tasks at a given time is fairly unpredictable, which makes resources allocation even more problematic. WebSty is based on a *service-oriented software* idea, that has gained great popularity, according to which each LT implemented as a *microservice* and run as a separate process with pre-loaded data models. The number of microservices run in parallel is limited by hardware. Each type of a LT has its own queue. A NLP microservice collects tasks from a given queue and sends back messages when the results are available.

The usage of microservices communicating via lightweight mechanisms solves the problems of a variety of programming languages used, and complexity of tight integration. As the number of microservices run in parallel is limited by hardware, the queening system is crucial for the system performance and effective scalability. The most required and most frequently used, LT microservices have to be run in several instances, and the queuing system acts as a load balancer. AMQP[4] protocol for lightweight communication mechanisms and RabbitMQ broker are involved. An additional server grants the access from the Internet, and works as a proxy for the core system delivering REST API for WebSty. This allows for easy integration with almost any kind of applications. The exchange of data between microservices, i.e. input/output of LT tools is done by a network file system. It makes the integration of new LT tools easier, since they are mostly designed in the manner that they expect a file as in the input and produce files as an output.

In order to achieve high availability, the system was deployed on a scalable hardware and software architecture that forms a private cloud (ten Blade Servers, connected by fast fiber channel with highly scalable midrange virtual storage designed to consolidate workloads into a single system for simplicity of management). XENServer controls each machine and forms a private cloud. Each frequently used NLP microservice is deployed on a separate virtual machine (Walkowiak, 2017).

4. Data Analysis

Documents can be uploaded in many formats, e.g. MS Word, PDF, plain texts etc. The format is automatically detected and text extracted. For larger data sets, a connection between WebSty and CLARIN-PL D-Space-based repository was established: data sets first deposited in the repository, can be selected for processing in WebSty. However, due to user demands, it is also possible to upload documents from a zip file identified by its URL. Text documents or fragments (texts can be automatically divided) are first converted into *feature* vectors of numerical values, that are filtered, transformed and finally processed by various data analysis method. The ultimate goal is to divide the

[3] Signature: http://www.philocomp.net/humanities/signature.htm, JGAAP: https://github.com/evllabs/JGAAP, JStylo: https://psal.cs.drexel.edu/index.php/JStylo-Anonymouth, StyleTool: https://github.com/lnmaurer/StyleTool

[4] AMQP: https://www.amqp.org, RabbitMQ: https://www.rabbitmq.com

vectors into similarity classes by clustering algorithms (unsupervised approach) or classifiers trained by ML on examples of text documents with known properties (supervised approach).

Features should reveal text properties that are characteristic for its author and his style, and should not be correlated with the semantic content. A feature can refer to any level of the language analysis, but should be based only on LTs that express a relatively small error. In WebSty, we implemented so far features based on the frequency of: word forms (words in text), punctuation, lemmas, grammatical classes (a rich tagset), Parts of Speech (as sets of grammatical classes), grammatical categories, bigrams and trigrams of grammatical classes, and semantic types of Proper Names. The lemmas and grammatical classes are obtained from morphosyntactic taggers, occurrence and types of name entities come from a Named Entity Recogniser. WebSty[5] has been already expanded with support for English and a prototype version of support for German. We plan to continue this process of converting WebSty architecture into a **multilingual system**. The extension depends on existing taggers and Named Entity Recognizers for the supported language. Due different tagsets used by taggers, we focus on the Universal Tagset as the input for the feature extraction.

The features which are suspected of introducing too much noise or not be relevant, can be **filtered** out on the basis of: raw value (e.g. minimal number of documents), weighted value (after preprocessing) or their type (e.g. specified lemmas, grammatical classes, bigrams, etc.).

Raw frequencies are often skewed, e.g. by document length, document content, or by general properties of a given lemma which is very frequent. WebSty offers several **weighting methods**: *tf* (normalised text frequency), *tf.idf*, vector normalisation, PMI (Pointwise Mutual Information) simple and discounted, and *tscore*. As the number of features can be very high, a few dimensionality reduction techniques were included, which include *SVD* (Singular Value Decomposition), *LSA* (Latent Semantic Analysis) (Landauer & Dumais, 1997) and Random projection..

Similarity of texts is computed from transformed vectors by several measures: *cosine*, *Dice*, *Jacquard*, *ratio* (a heuristics measuring the average ratio of commonality), *shd* (a heuristics measuring the precision of mutual rendering of the two vectors). Several data analysis algorithms prefer the distance measure between vectors: *Manhattan, Canberra, euclidean*, *Simple* (L1 on vectors normalised by a square root function) (Eder, 2016) *Burrows's Delta*, *Argamon* (Euclidean distance combined with Z-score normalisation), and *Eder's delta* (Eder, 2016). WebSty also provides psychologically motivated conversion of similarity to distance by arc cosine function.

For **clustering** vectors, the combined *agglomerative-flat* clustering method from Cluto (Zhao & Karypis, 2005) was selected as it providing two perspectives: pairwise hierarchy of similarity and flat division into a predefined, expected number of clusters. Clustering is controlled by three parameters: number of clusters, similarity measure and clustering criterion function. Selecting the criterion function is complicated issue, thus some ready to use defaults are provided in WebSty.

5. Data Visualisation and Exploration

The data analysis process produces: similarity/distance values calculation (2D matrix) and results of clustering. The latter can be downloaded as an XLSX file or presented in a graphical form as a **dynamic dendrogram** – an interactive binary tree, where each node (and a subtree) can be collapsed (JavaScript and *D3.js*[6] *library*). **Similarity results** are presented as: a *heatmap* (a matrix showing similarity by colours) and a *schemaball*. In the schemaball plot the user can select a file name and analyse similarity of texts by connections shown, their colour and thickness. For larger text collections, a multidimensional scaling can be very helpful to visualise a set of multidimensional vectors in 2D or 3D space (interactive presentation). WebSty offers four methods of multidimensional scaling: *metric*, preserving distances, *non-metric*, preserving orders in distances, *t-distributed Stochastic Neighbor Embedding* (Maaten et al., 2008), preserving similarities, and *spectral embedding* (Belkin et al., 2003), preserving local neighborhood. Results after scaling are presented as points in 2D space (*3D.js library*) or in 3D space. The interactive 3D plot utilises the *three.js* library, based on *WebGL* and using the user's graphic card 3D acceleration.

[5] WebSty: http://ws.clarin-pl.eu/webstyen.shtml?en, Universal Tagset: http://universaldependencies.org/u/pos/

[6] D3.js: https://d3js.org/, Three: https://threejs.org/, WebGL: http://www.khronos.org/registry/webgl/specs/latest/

In response to frequent users' questions: which features are are responsible for producing a given cluster, we added a module for **selection of important features**. It is based on a set of statistical and ML methods assuming that enough training data is provided. The implemented methods include: statistical tests (for example Mann-Whitney), information metrics (for example InfoGain), recursive feature elimination using supervised classifiers (like Naive Bayes) and feature importances available in tree based classifiers, e.g. Random Forest.

6. Conclusions and further development

WebSty was implemented as a part of the CLARIN-PL infrastructure and made publicly available to the CLARIN-PL users. WebSty (different versions) has been already applied to several research tasks from the area of SS&H, as well as used in teaching. Among its applications worth mentioning is the literary analysis of styles of web blogs (Maryl et al., 2016).

WebSty has been so far focused on unsupervised processing by clustering. We are working on an extended version offering support for using a supervised approach in which classifiers are trained by ML on the basis of manually annotated data sets, e.g. sets of texts annotated with authors' names. We also working on extending WebSty into a system enabling unsupervised and supervised semantic analysis of text data sets, e.g. identification of text fragments that are related to situations of specific types or to specific phenomena. Topic analysis will be very soon included into WebSty as both, a tool by itself and also as tool used for data preprocessing.

Bibliography (selected)

Belkin, M., Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation, 15(6): 1373–1396.

Eder, M., Rybicki, J. and Kestemont, M. (2016). Stylometry with R: a package for computational text analysis. R Journal, 8(1): 107–121, http://journal.r-project.org/archive/2016-1/eder-rybicki-kestemont.pdf.

Landauer, T. & Dumais, S. (1997) A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition. Psychological Review, 1997, 104, pp. 211-240.

Le, X., Lancashire, I., Hirst, G. and Jokel, R. (2011). Longitudinal detection of dementia through lexical and syntactic changes in writing: a case study of three British novelists. Literary and Linguistic Computing, 26(4): 435–461.

van der Maaten, L.J.P.; Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), pp.: 2431–2456.

Maryl, M., Piasecki, M. & Młynarczyk, K. (2016) Where Close and Distant Readings Meet: Text Clustering Methods in Literary Analysis of Weblog Genres. In Eder, M. & Rybicki, J. (Eds.) Digital Humanities 2016 Conference Abstracts, Jagiellonian University and Pedagogical University, pp. 273-275.

Stamatatos, E. (2009). A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology, 60(3): 538–556.

Walkowiak, T. (2017). Language Processing Modelling Notation – orchestration of NLP microservices. In: Proceedings of the Twelfth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, (to appear).

Zhao, Y. and Karypis, G. (2005). Hierarchical Clustering Algorithms for Document Datasets. *Data Mining and Knowledge Discovery*, **10**(2): 1.