SAWT: Sequence Annotation Web Tool

Younes Samih and **Wolfgang Maier** and **Laura Kallmeyer**
Heinrich Heine University
Department of Computational Linguistics
Universitätsstr. 1, 40225 Düsseldorf, Germany
{samih,maierwo,kallmeyer}@phil.hhu.de

Abstract

We present SAWT, a web-based tool for the annotation of token sequences with an arbitrary set of labels. The key property of the tool is simplicity and ease of use for both annotators and administrators. SAWT runs in any modern browser, including browsers on mobile devices, and only has minimal server-side requirements.

1 Introduction

Code-switching (Bullock and Toribio, 2009) occurs when speakers switch between different languages or language variants within the same context. In the Arab world, for instance, it is a common phenomenon. Both Modern Standard Arabic (MSA) and Dialectal Arabic (DA) variants co-exist, MSA and DA being used for formal and informal communication, respectively (Ferguson, 1959). Particularly recently, the computational treatment of code-switching has received attention (Solorio et al., 2014).

Within a project concerned with the processing of code-switched data of an under-resourced Arabic dialect, Moroccan *Darija*, a large code-switched corpus had to be annotated token-wise with the extended label set from the EMNLP 2014 Shared Task on Code-Switching (Solorio et al., 2014; Samih and Maier, 2016). The label set contains three labels that mark MSA and DA tokens, as well as tokens in another language (English, French, Spanish, Berber). Furthermore, it contains labels for tokens which mix two languages (e.g., for French words to which Arabic morphology is applied), for ambiguous words, for Named Entities, and for remaining material (such as punctuation).

The annotation software tool had to fulfill the following requirements.

- It should excel at sequence annotation and not do anything else, i.e., ”featuritis” should be avoided, furthermore it should be as simple as possible to use for the annotators, allowing for a high annotation speed;
- It should not be bound to a particular label set, since within the project, not only code-switching annotation, but also the annotation of Part-of-Speech was envisaged;
- It should allow for post-editing of tokenization during the annotation;
- It should be web-based, due to the annotators being at different physical locations;
- On client side, it should be platform-independent and run in modern browsers including browsers on mobile devices, using modern technologies such as Bootstrap[1] which provide a responsive design, without requiring a local installation of software;
- On server side, there should be safe storage; furthermore, the administration overhead should be kept minimal and there should only be minimal software requirements for the server side.

[1]http://getbootstrap.com

Even though several annotation interfaces for similar tasks have been presented, such as COLANN (Benajiba and Diab, 2010), COLABA (Diab et al., 2010), and DIWAN (Al-Shargi and Rambow, 2015), they were either not available or did not match our needs.

We therefore built SAWT. SAWT has been successfully used to create a code-switched corpus of 223k tokens with three annotators (Samih and Maier, 2016). It is currently used for Part-of-Speech annotation of Moroccan Arabic dialect data. The remainder of the article is structured as follows. In section 2 we present the different aspects of SWAT, namely, its data storage model, its server side structure and its client side structure. In section 3, we review related work, and in section 4, we conclude the article.

2 SAWT

SAWT is a web application. Its client side is machine and platform independent and runs in any modern browser. On the server side, only a PHP-enabled web server (ideally Apache HTTP server) and a MySQL database instance are needed.

We now describe our strategy for data storage, as well as the server side and the client side of SAWT.

2.1 Data storage

Data storage relies on a MySQL database. One table in the database is used to store the annotator accounts. At present, there is no separate admin role, all users are annotators and cannot see or modify what the other annotators are doing.

The annotation of a text with a label set by a given user requires two MySQL tables. One table contains the actual text which is to be annotated by the user, and the other table receives the annotation; this table pair is associated with the user account which is stored in the user table mentioned above. In the first table, we store one document per row. We use the first column for a unique ID; the text is put in the second column. It is white-space tokenized at the moment it is loaded into the annotation interface (see below). In the second table, we store the annotation. Again, the document ID is put into the first column. The labels are stored in the remaining columns (one column per label).

2.2 Server side and administration

The complete code of SAWT will be distributed on github. The distribution will contain the complete web application code, as well as two Python scripts to be used for configuration.

The first script configures the SAWT installation and the database. It takes a configuration file as parameter (the distribution will contain a configuration file template), in which the following parameters must be specified:

- *List of tags*: A space-separated list of tags to be used in the annotation. From this list, the PHP code for the model and the view are generated which handle the final form in the interface. The generated code is then copied to the correct locations within the complete web application code.
- *Server information*: MySQL server IP, port and user account information.
- *Predictor*: The interface can show suggestions for each token, provided that a suitable sequence labeling software with a pre-trained model runs on the web server. If suggestions are desired, then in the configuration file, the corresponding path and parameters for the software must be given. If the parameter is left blank, no suggestions are shown.
- *Search box activation*: A boolean parameter indicating if a search box is desired. In the search box, the annotator can look up his previous annotations for a certain token.
- *Utility links*: The top border of the user interface consists of a link bar, the links for which can be freely configured. In our project, e.g., they are used for linking to the list of Universal POS tags (Petrov et al., 2012), to a list of Arabic function words, to an Arabic Morphological Analyzer (MADAMIRA) (Pasha et al., 2014), and to an Arabic screen keyboard, as can be seen in figures 2 and 3.

Once the configuration script has been run, the web application code must be copied to a suitable place within a web server installation.

In order to upload a text which is to be annotated by a certain user, the second script must be used. It takes the following command line parameters.

- *Input data*: The name of the file containing the data to be annotated. The text must be pre-tokenized (one space between each token), and there must be one document per line.

- *Server information*: MySQL server IP, port, and user account information.

- *Annotator information*: Annotator user name. If the annotator account does not exist in the respective database table, it is created, and a password must be specified.

Of course, this script can be used any number of times. At runtime, it will connect to the database and create two tables for the annotation (as mentioned above, one for the data itself and one for the annotation). It will insert the data in the first one, and insert the user account in the user account table, if necessary.

In general, for security reasons, two different servers should be used for front-end (web application) and back-end (database), but in principle, nothing stands in the way of installing everything on a single machine or even locally.

2.3 Client side and annotator interface

The client side interface is written with several technologies. As a basis, we have used a MVC PHP framework, namely CodeIgniter version 3.0.[2] Furthermore, in order to achieve a responsive mobile-ready design, we have employed to the Bootstrap framework, HTML 5, and JQuery.[3]

When accessing the URL where SAWT is located, the annotator is queried its user name and password. After logging in, the annotation interface is shown. On top of the page, a link bar makes available several tools which are useful for the annotation, to be freely configured during installation (see above). If configured (see above), a search box is shown, in which the annotator can look up his previous annotations of a token. In a top line above the text, the ID of the document is shown, the number of tokens to be annotated, and the annotation progress, i.e., the number of tokens which have already been annotated (in previous documents). Also it is shown if the current document itself has already been annotated. Finally, there are buttons to navigate within the documents (first, previous, next, last).

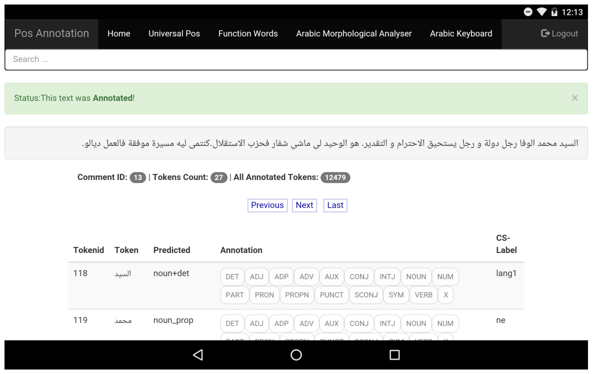
For the annotation, the interface pulls the first document to be annotated from the database, applies white-space tokenization, and renders it for presentation to the user. The material to be annotated is presented with one document per page and token per line. Each line has four columns, the first one showing the absolute token ID within the complete corpus, the second one showing the token to be annotated, the third one showing a prediction of a possible tag (if configured), and the fourth one showing the possible labels. There is an edit facility, in which the annotator can correct an erroneous tokenization of the document. If an edit is performed, the modified document is white-space tokenized again and reloaded in the interface.

For label selection, we offer check-boxes. Even though radio buttons would seem to be the more natural choice, check-boxes allow us to assign several tags to a single token. This is, e.g., essential for Part-of-Speech annotation in Arabic: Due to a rich morphology, a single word can incorporate several POS functions (Habash, 2010). When the user has finished the annotation of a document, a button must be clicked. This button first triggers a validation function which checks the annotation for completeness. If there are tokens which have not been annotated, a colored alert bar is shown. Otherwise, a form is submitted which saves the annotation in the database; then the next document is loaded and rendered for annotation. We have implemented the policy that an annotator cannot change the annotation of a document once it is submitted. However, a minimal change in the code could allow a post-editing of the annotation.

We have tested the interface extensively in Google Chrome (on both PC and Android) and Mozilla Firefox.

As an example, figure 2 shows a screenshot of the annotator interface configured for Part-of-Speech annotation with the Google Universal Part-of-Speech tag set (Petrov et al., 2012). Figure 3

[2]`http://codeigniter.net`
[3]`http://jquery.com`



Status:This text was **Annotated**!

السيد محمد الوفا رجل دولة و رجل يستحيق الاحترام و التقدير. هو الوحيد لي ماشي شفار فحزب الاستقلال.كنتمى ليه مسيرة موفقة فالعمل ديالو.

Comment ID: 13 | Tokens Count: 27 | All Annotated Tokens: 12479

Tokenid	Token	Predicted	Annotation	CS-Label
118	السيد	noun+det	DET ADJ ADP ADV AUX CONJ INTJ NOUN NUM PART PRON PROPN PUNCT SCONJ SYM VERB X	lang1
119	محمد	noun_prop	DET ADJ ADP ADV AUX CONJ INTJ NOUN NUM	ne

Figure 1: Screenshot of SAWT: Annotation on Android device

shows a screenshot of code-switching annotation done in the context of our earlier work (Samih and Maier, 2016). Finally, figure 1 shows a screenshot of the POS annotation interface used on the Asus Nexus 7 2013 tablet running Google Chrome on Android 6.

3 Related Work

As mentioned above, we are not aware of a software which would have fulfilled our needs exactly. Previously released annotation software can be grouped into several categories.

Systems such as GATE (Cunningham et al., 2002), CLaRK (Simov et al., 2003) and MMAX2 (Müller and Strube, 2006) are desktop-based software. They offer a large range of functions, and are in general oriented towards more complex annotation tasks, such as syntactic treebank annotation.

In the context of Arabic dialect annotation, several systems have been created. COLANN_GUI (Benajiba and Diab, 2010), which unfortunately was not available to us, is a web application that specialized on dialect annotation. DIWAN (Al-Shargi and Rambow, 2015) is a desktop application for dialect annotation which can be used online.

The systems that came closest to our needs were WebANNO (Yimam et al., 2013) and BRAT (Stenetorp et al., 2012). Both are web-based and built with modern technologies. They allow for a multi-layered annotation, including a token-wise annotation. However, we decided against them due to fact that we just needed the token-wise annotation and we wanted the simplest annotator interface possible. For just sequence annotation, our annotator interface allows for a very high speed, since only one click per token is required.

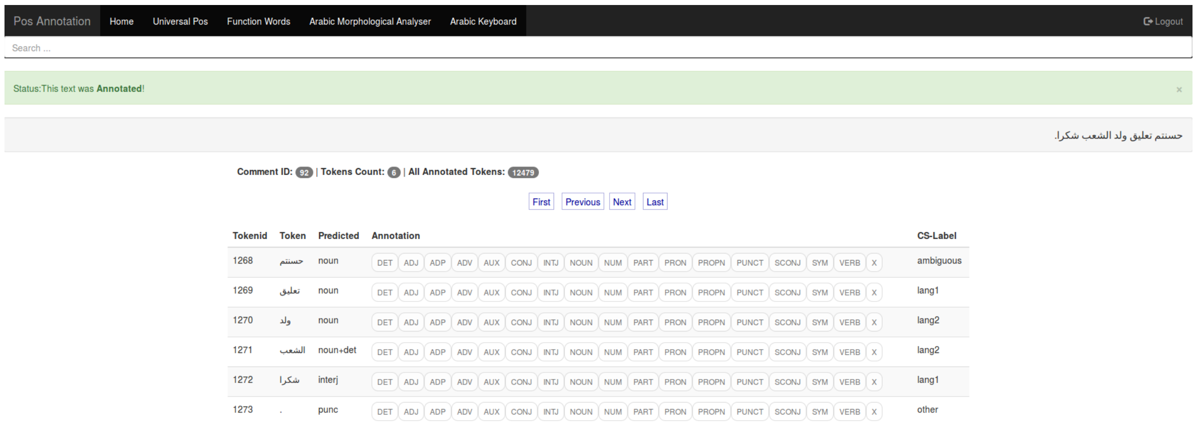
4 Conclusion

We have presented SAWT, a web-based tool for sequence annotation. The main priorities of the tool are ease of use on the client side and a low requirements for the server side.

SAWT is under active development. We are currently simplifying the installation process on server side and plan to offer an admin role in the front-end. Furthermore, we want to provide a way of obtaining the annotation in a standardized format (TEI) directly from the database.

Acknowledgments

We would like to thank Ikram Zeraa and Miloud Samih for enlightening discussions and the three anonymous reviewers for helpful comments and suggestions. The authors were partially funded by Deutsche Forschungsgemeinschaft (DFG).



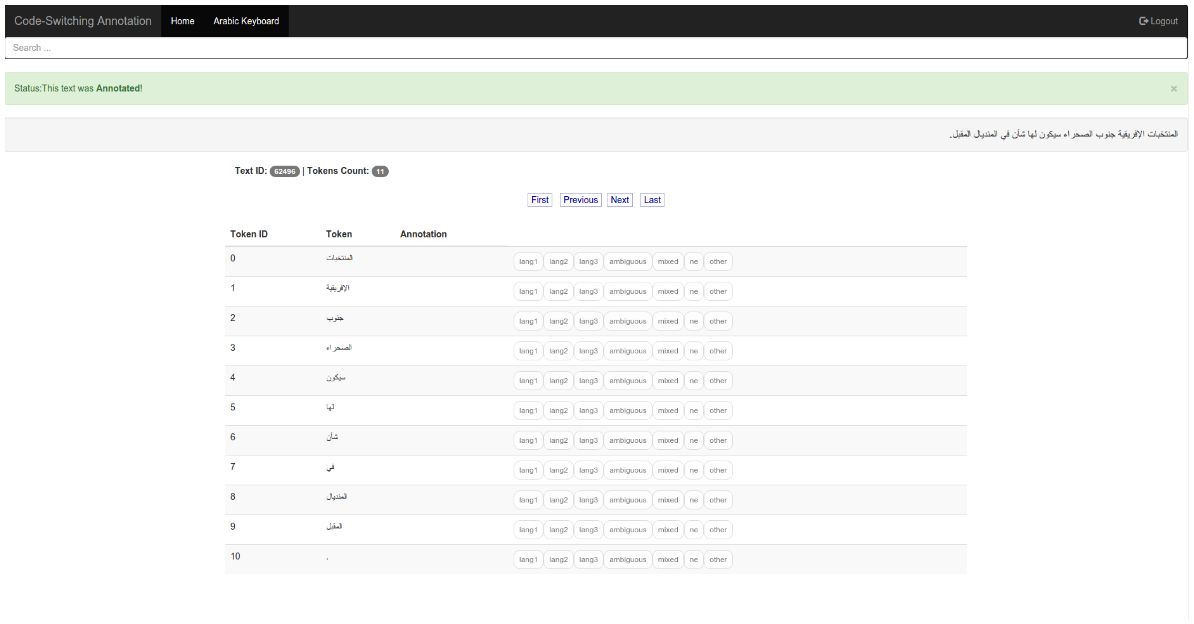
Status:This text was **Annotated**!

حسنتم تعليق ولد الشعب شكرا.

Comment ID: 92 | **Tokens Count:** 6 | **All Annotated Tokens:** 12479

Tokenid	Token	Predicted	Annotation	CS-Label
1268	حسنتم	noun	DET ADJ ADP ADV AUX CONJ INTJ NOUN NUM PART PRON PROPN PUNCT SCONJ SYM VERB X	ambiguous
1269	تعليق	noun	DET ADJ ADP ADV AUX CONJ INTJ NOUN NUM PART PRON PROPN PUNCT SCONJ SYM VERB X	lang1
1270	ولد	noun	DET ADJ ADP ADV AUX CONJ INTJ NOUN NUM PART PRON PROPN PUNCT SCONJ SYM VERB X	lang2
1271	الشعب	noun+det	DET ADJ ADP ADV AUX CONJ INTJ NOUN NUM PART PRON PROPN PUNCT SCONJ SYM VERB X	lang2
1272	شكرا	interj	DET ADJ ADP ADV AUX CONJ INTJ NOUN NUM PART PRON PROPN PUNCT SCONJ SYM VERB X	lang1
1273	.	punc	DET ADJ ADP ADV AUX CONJ INTJ NOUN NUM PART PRON PROPN PUNCT SCONJ SYM VERB X	other

Figure 2: Screenshot of SAWT: Annotation with Universal Part Of Speech tags



Status:This text was **Annotated**!

المنتخبات الإفريقية جنوب الصحراء سيكون لها شأن في المنديال المقبل.

Text ID: 62496 | **Tokens Count:** 11

Token ID	Token	Annotation
0	المنتخبات	lang1 lang2 lang3 ambiguous mixed ne other
1	الإفريقية	lang1 lang2 lang3 ambiguous mixed ne other
2	جنوب	lang1 lang2 lang3 ambiguous mixed ne other
3	الصحراء	lang1 lang2 lang3 ambiguous mixed ne other
4	سيكون	lang1 lang2 lang3 ambiguous mixed ne other
5	لها	lang1 lang2 lang3 ambiguous mixed ne other
6	شأن	lang1 lang2 lang3 ambiguous mixed ne other
7	في	lang1 lang2 lang3 ambiguous mixed ne other
8	المنديال	lang1 lang2 lang3 ambiguous mixed ne other
9	المقبل	lang1 lang2 lang3 ambiguous mixed ne other
10	.	lang1 lang2 lang3 ambiguous mixed ne other

Figure 3: Screenshot of SAWT: Annotation with code-switching labels

References

Faisal Al-Shargi and Owen Rambow. 2015. Diwan: A dialectal word annotation tool for Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 49–58, Beijing, China. Association for Computational Linguistics.

Yassine Benajiba and Mona Diab. 2010. A web application for dialectal Arabic text annotation. In *Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects*, Valletta, Malta. ELRA.

Barbara E. Bullock and Almeida Jacqueline Toribio. 2009. *The Cambridge handbook of linguistic code-switching*. Cambridge University Press.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 168–175. Association for Computational Linguistics.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. COLABA: Arabic dialect annotation and processing. *Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects*, pages 66–74.

Charles Ferguson. 1959. Diglossia. *Word*, 15:325–340.

Nizar Y. Habash. 2010. Introduction to Arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with mmax2. *Corpus technology and language pedagogy: New resources, new tools, new methods*, 3:197–214.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.

Younes Samih and Wolfgang Maier. 2016. An arabic-moroccan darija code-switched corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portoroz, Slovenia.

Kiril Simov, Alexander Simov, Milen Kouylekov, Krasimira Ivanova, Ilko Grigorov, and Hristo Ganev. 2003. Development of corpora within the CLaRK system: The BulTreeBank project experience. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 243–246. Association for Computational Linguistics.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April. Association for Computational Linguistics.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. Webanno: A flexible, web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 1–6, Sofia, Bulgaria, August. Association for Computational Linguistics.