A Web Tool for Building Parallel Corpora of Spoken and Sign Languages

Alex Becker*, **Fabio Kepler***†, **Sara Candeias**‡
*UNIPAMPA - Federal University of Pampa – Alegrete/Brazil
†L2F/INESC-ID – Lisbon/Portugal
‡Microsoft LDC – Lisbon/Portugal
*alex@porthal.com.br, †fabio@kepler.pro.br, ‡t-sacand@microsoft.com

Abstract

In this paper we describe our work in building an online tool for manually annotating texts in any spoken language with SignWriting in any sign language. The existence of such tool will allow the creation of parallel corpora between spoken and sign languages that can be used to bootstrap the creation of efficient tools for the Deaf community. As an example, a parallel corpus between English and American Sign Language could be used for training Machine Learning models for automatic translation between the two languages. Clearly, this kind of tool must be designed in a way that it eases the task of human annotators, not only by being easy to use, but also by giving smart suggestions as the annotation progresses, in order to save time and effort. By building a collaborative, online, easy to use annotation tool for building parallel corpora between spoken and sign languages we aim at helping the development of proper resources for sign languages that can then be used in state-of-the-art models currently used in tools for spoken languages. There are several issues and difficulties in creating this kind of resource, and our presented tool already deals with some of them, like adequate text representation of a sign and many to many alignments between words and signs.

Keywords: Sign language recognition, corpora creation, crowd-sourcing

1. Introduction

Sign languages are the main way of communication in the Deaf community and with the listening population, and are almost always expressed in the form of visual signs and expressions. There are over 200 distinct sign languages in the world, and about 70 million deaf people[1]. Some countries have an official sign language, like Brazil, where LIBRAS (Brazilian Sign Language, in Portuguese) is the second official language besides Portuguese. Brazil also has 5.7 million people with hearing impairment (of the Deaf, 2008; IBGE, 2000), with 1.7 million being sign language users.

Unfortunately, as one could infer from the numbers above, many deaf do not know a sign language. Instead, they communicate via gestures and by vocalizing (i.e., trying to utter words). This lack of communication capabilities makes them prone to be highly dependent or to live isolated socially. Nevertheless, several deaf adults cannot read nor write a spoken language.

When there is prelingual deafness, i.e., when a child loses hearing before beginning to speak, a sign language becomes the child's native language. This impairs their ability to acquire a spoken language (in its written form). And because schools are not prepared and usually apply bad learning methods, deaf children stay behind their colleagues in terms of knowledge development. They eventually reach the same level, but usually take longer.

A relatively new form of expressing sign languages is the written form. There are several proposed schemes for writing sign languages, but one being more prominently used is the SignWriting system, which was created in 1974 by Valerie Sutton (Barreto and Barreto, 2012). The SignWriting system defines sets of symbols for handshapes, facial expressions, body locations, orientation, contact, and movement. Figure 1 shows an example of symbols for handshapes.

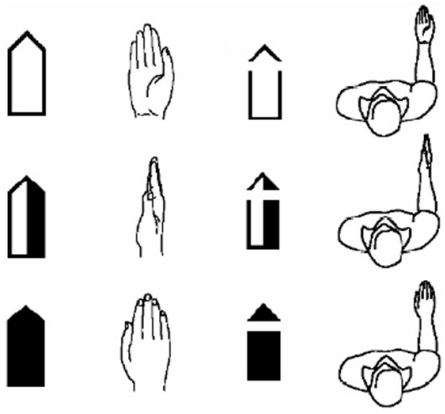

Figure 1: Example of symbols for handshapes (Source: https://en.wikipedia.org/wiki/File:SWpalms.png).

A combination of these symbols form an iconic sign which represents a proper sign (a "word") in a sign language. Since a sign represents the physical formation of a visual sign and not its meaning, the SignWriting system can be used for writing in any sign language, like alphabets can be used for writing words in many spoken languages. An example phrase can be seen in Figure 2.

Because signs in SignWriting highly resemble the visual signs, children and adults can quickly learn to read and write it. Written material with both sign and spoken languages greatly help children acquire both languages. Being able to write and read in their native language also helps them develop faster and evolve alongside their listening classmates.

Given that, adequate computer tools would greatly help the Deaf community, but unfortunately there are several gaps when dealing with sign languages in contrast to spoken lan-

[1]World Federation of the Deaf: http://wfdeaf.org/human-rights/crpd/sign-language

Figure 2: Example of a sentence written in SignWriting. From top to bottom, the words are "I learn LIBRAS", in LIBRAS.

guages. For example, machine translation, which would leverage the creation of educational material in sign language, needs a large amount of parallel data.

The main objective of this work is to build an online tool for manually annotating texts in any spoken language with SignWriting in any sign language. This will allow the creation of parallel corpora between spoken and sign languages that can be used to bootstrap the creation of efficient tools for the Deaf community. For example, a parallel corpus between English and American Sign Language could be used for training Machine Learning models for automatic translation between the two languages. Such system must be designed in a way that it eases the task of human annotators, not only by being easy to use but also by giving smart suggestions as the annotation progresses.

This paper is organized as follows: Section 2. presents a suitable computer representation of signs and its issues; Section 3. mentions some previous works; Section 4. explains some implementation details of our proposed tool and its current status; and Section 5. draws some discussion and future plans.

2. SignWriting Representation

In the technological domain, signs stored as images have limited applicability. For example, searching for a given sign would require image processing and indexing. There has been a number of attempts at encoding signs, one of the latest being the Formal SignWriting (FSW) format (Slevinski, 2015). FSW uses ASCII or Unicode for encoding strings which represent logographic words (signs). For example, the string

`M518x517S16d10494x467S33e00482x482S31b00482x482S21900496x456S20500475x476`

corresponds to the sign

Figure 3: Example of spelling variations of the sign for the word *deaf* in LIBRAS, all of which will have different FSW encodings.

FSW also defines the format of query strings for searching symbols, ranges, and positions[2].

Although FSW defines a formal language, it has no spelling normalization[3]: it does not fix the order in which symbols are encoded in the string nor the exact position of symbols in the 2-dimensional SignBox (the sign's two-dimensional space). This means that it is possible to have several strings with exactly the same 2-dimensional visual appearance, and that the same sign can be written with slightly different positions of its symbols. So, since a writer can start with any symbol and can position it precisely, it is very unlikely that two writers will produce the same spelling for any sign. Figure 3 shows a simple example.

These issues make it harder to query for specific signs and pose a challenge for sign language tools.

3. Related Work and Current Resources

There has been some recent effort in developing better technological tools for the Deaf community, notably the SignPuddle portal[4][5]. Though it provides word-sign dictionaries, they are prone to noisy annotations since they are open for public editing. Also, although it features a translator from spoken to signed languages, it is unfortunately only based on a simple dictionary prefix lookup, which yields hundreds of unrelated signs for short words and does not detect word inflections.

Some works explored the need for sign language resources in order to perform machine translation. Stein et al. (2012) talk about sign language corpora of videos, and present a project that aims to translate visual sign language to written spoken language, with a written sign language middle step. However, they do not use the SignWriting system for this intermediate step, and conclude that this pipeline might propagate errors because their chosen written representation might not capture all sign information.

Morrissey et al. (2010) explores the building of a sign language corpus also for machine translation, but they use the HamNoSys system for transcribing videos to text and do not provide links to their corpus.

Finally, several works (Li et al., 2011; Zaki and Shaheen, 2011; Almeida et al., 2014) explore the recognition of visual signs and their direct translation to spoken text, without dealing with sign text.

[2] http://signpuddle.net/mediawiki/index.php/MSW:Formal_SignWriting#9.B._Query_String

[3] http://signpuddle.net/wiki/index.php/MSW:Spelling_Normalization

[4] http://www.signpuddle.net

[5] http://www.signbank.org/signpuddle/

4. A Spoken-Sign Corpus Annotation Tool

Given the current lack of digital resources for sign languages, there is also a lack of state-of-the-art computational tools when compared to tools available for spoken languages. Natural Language Processing problems are always tackled for spoken languages since they have more resources which are needed for training state-of-the-art machine learning models. Sign languages have some particular problems compared to spoken languages, like their 3-dimensional nature (written and spoken words can be seen as being 1-dimensional). Nevertheless, they could at least be partially treated by existing machine learning models if enough training data was available.

Building annotated data, however, is not cheap nor easy even for spoken languages. As we stated above, an annotation tool greatly helps humans annotate data.

Our approach for easing the task of building resources for sign languages was to develop a web tool for the annotation of parallel bilingual corpora, more specifically, between a spoken and a sign language. Besides using SignWriting for representing a sign language text, we aim at supporting multiple sign and spoken languages, at allowing for collaborative annotation, and at providing annotation suggestions based on previous activities.

To allow multiple languages, text in any spoken language can be added, and different sign language dictionaries can be created. Since the SignPuddle portal has some dictionaries available[6], our tool can bootstrap initial annotation by allowing these dictionaries to be imported. As annotation progresses, the dictionaries are refined and improved.

Collaboration can also bootstrap and leverage annotation, and this is possible by the web nature of the tool and its user management. This functionality, however, raises the problem of spelling normalization discussed in Section 2.. We plan to address this issue by developing methods for finding and merging similar FSW strings.

To further help annotation, suggesting signs for each word should not be only based on dictionary lookups. Instead, the tool uses the relative frequency of already annotated word/sign pairs. A next step is to also use the phrase context for disambiguation by learning language modelsfor signs.

4.1. Design and Implementation

The tool was implemented in the Java Web platform using the JSF framework (Java Server Faces) and an MVC architecture (Model-View-Controller).

Figure 4 shows the application domain diagram. The two main packages are *dictionary* and *corpora*. The first is responsible for controlling the dictionaries and their entry pairs, and a frequency attribute is also maintained for improving candidate sign suggestion. The second is responsible for keeping the corpora and the annotation statuses of each document, where the tokens of each sentence of a document are linked to dictionary entries.

Although a word is aligned with zero or one sign, the tool allows multi-token annotation, i.e., a single sign being assigned to a phrase, by allowing the user to concatenate (or split) consecutive words.

Figure 5 shows the process of creating and annotating a document. It starts with the user selecting or creating a corpus. After that, the user creates a raw document (in Portuguese, for example) and runs the document preparation process, in which the system segments the document into sentences and then into tokens. Raw documents may be either manually uploaded as text or automatically fetched from Wikipedia given their URL. The user can then start selecting sentences to annotate. The system searches for candidate signs for each word (or phrase) and ranks them according to similarity, usage frequency, and in the near future by context. The user then combines or separates words and selects the best sign for each entry. As it is common in sign languages, some words in spoken languages have no sign. These are just left empty by the user. Once every sentence is annotated, the document itself is marked as annotated.

Figure 6 shows a screenshot of the corpora management interface. The user can select an existing corpus to resume annotating or create a new parallel corpus, in which case she inputs a spoken language name and selects or creates a sign language dictionary (not shown in the figure).

Figure 7 shows a screenshot of the sentence annotation interface (with a sentence in Portuguese). User selected signs are shadowed and highlighted in green, and different operations are available both for words and signs, like concatenating/splitting and adding a new FSW, respectively.

To add a new FSW the user can just write or paste its string, but to ease the task, she can also draw the sign using the SignMaker[7] tool, which was embedded in the system as shown in Figure 8. This tool allows the user to draw a sign by selecting its composing symbols, which are organized by the sets mentioned earlier: handshapes, facial expressions, body locations, orientation, contact, and movement. It automatically generates the corresponding FSW, which can then be used in the sentence annotation.

Finally, an important functionality and the tool's main reason to be is that, at any moment, the user is able to export a corpus or a collection of documents in order to generate a parallel corpus in a simple file format, which can then be used in other tools.

5. Discussion and Future Work

By building a collaborative, online, easy to use annotation tool for building parallel corpora between spoken and sign languages we aim at helping the development of proper resources for sign languages that can then be used in state-of-the-art models currently used in tools for spoken languages. There are several issues and difficulties in creating this kind of resource, and our tool already deals with some of them, like adequate text representation of a sign and many to many alignments between words and signs. In order to further help the community, we make both a running instance of the tool (obviously in very beta stage) and its source code freely available at https://bitbucket.org/unipampa/signcorpus.

[6] http://www.signbank.org/signpuddle2.0/index.php?ui=12&sgn=46

[7] By Stephen E. Slevinski Jr, available at https://github.com/Slevinski/signmaker and http://slevinski.github.io/signmaker.

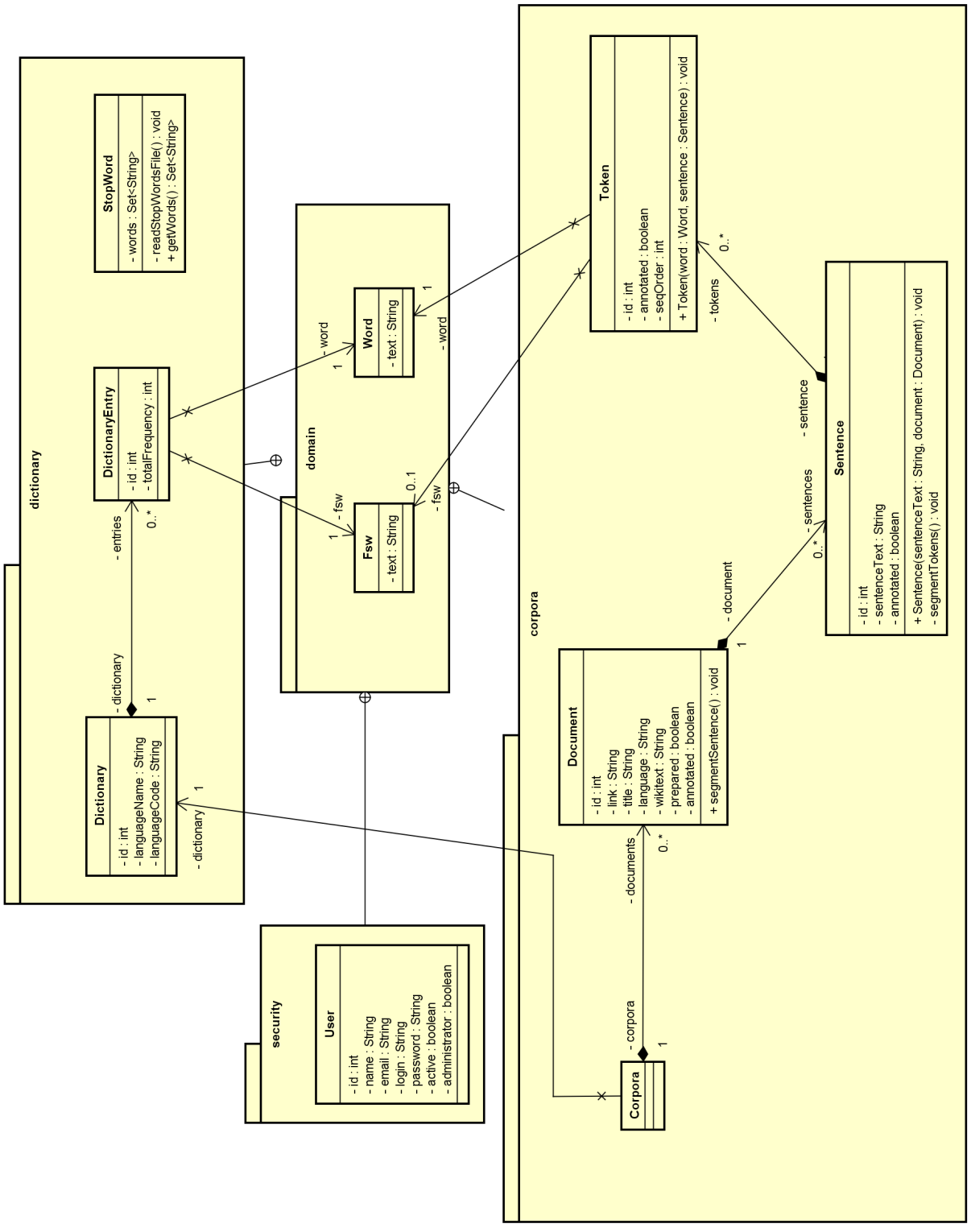

Figure 4: Domain diagram.

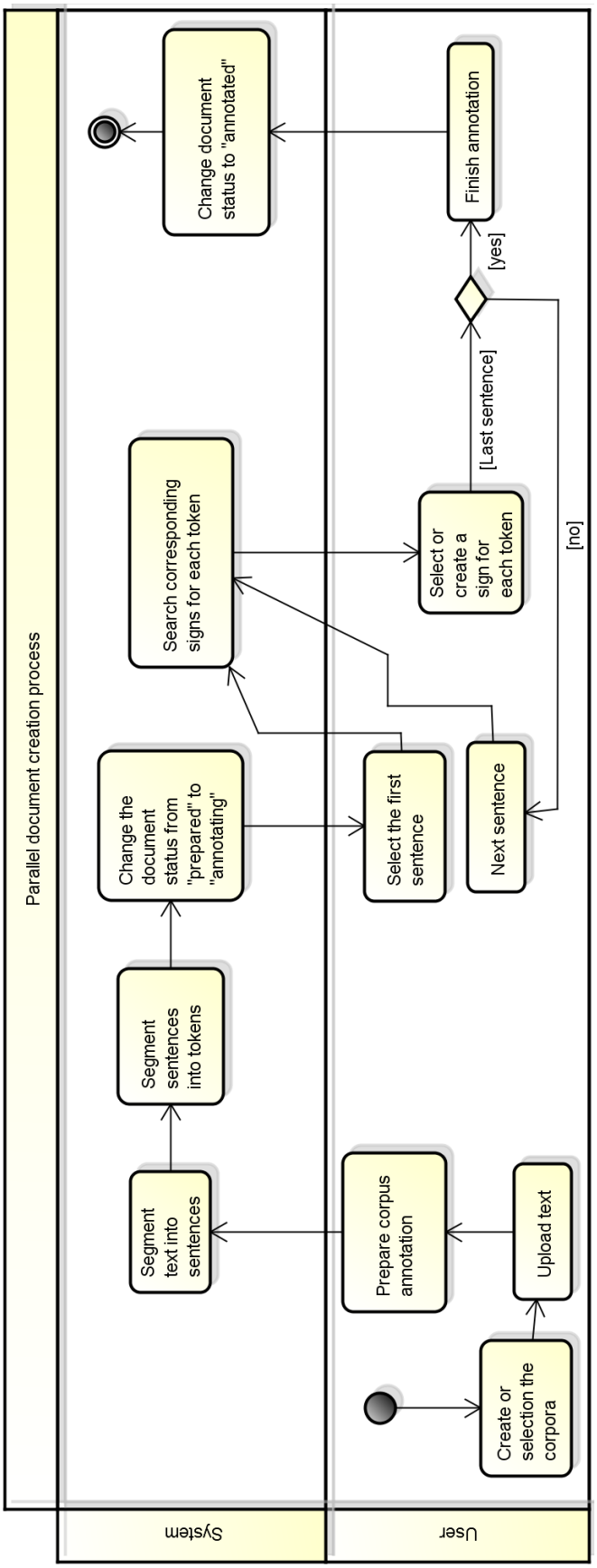

Figure 5: Process for creating a parallel document.

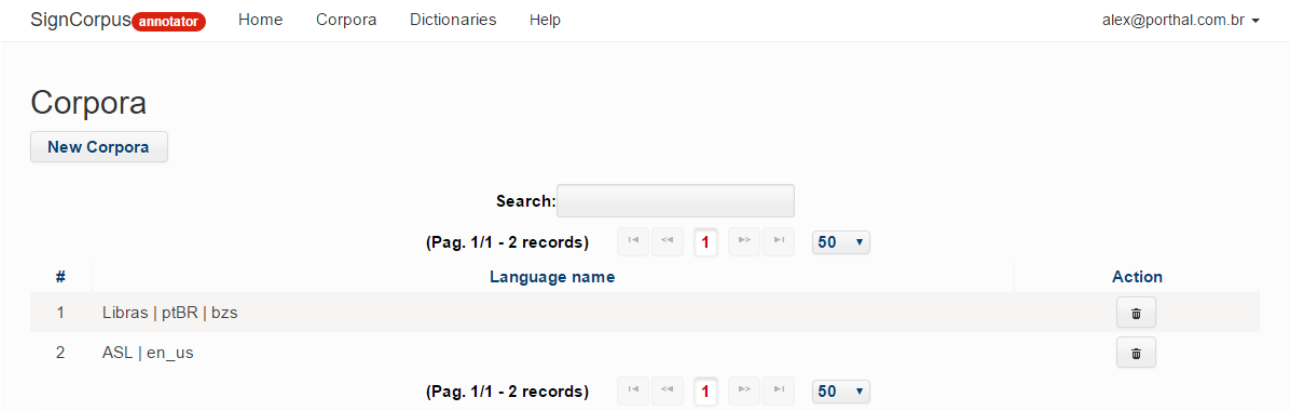

Figure 6: Screenshot of the corpora interface.

Our next step is to improve the searching and ranking of candidate signs by considering word inflections and by building language models for sign sentences.

Reporting the current usage status, we are starting arrangements with professors and researchers at UNIPAMPA for an initial effort in annotating a Portuguese-LIBRAS corpus.

Finally, and more importantly, help from the community in developing or using the tool would be very well welcomed.

6. Acknowledgments

This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

7. Bibliographical References

Almeida, S. G. M., Guimarães, F. G., and Ramírez, J. A. (2014). Feature extraction in brazilian sign language recognition based on phonological structure and using rgb-d sensors. *Expert Systems with Applications*, 41(16):7259–7271.

Barreto, M. and Barreto, R. (2012). Escrita de sinais sem mistérios. *Ed. do Autor, BH*.

IBGE. (2000). Censo demográfico 2000.

Li, K. F., Lothrop, K., Gill, E., and Lau, S. (2011). A web-based sign language translator using 3d video processing. In *Network-Based Information Systems (NBiS), 2011 14th International Conference on*, pages 356–361. IEEE.

Morrissey, S., Somers, H., Smith, R., Gilchrist, S., and Dandapat, S. (2010). Building a sign language corpus for use in machine translation. In *4th Workshop on Representation and Processing of Sign Languages: Corpora for Sign Language Technologies*.

of the Deaf, W. F. (2008). Global Survey Report, WFD Regional Secretariat for South America, Global Education Pre-Planning Project on the Human Rights of Deaf People.

Stein, D., Schmidt, C., and Ney, H. (2012). Analysis, preparation, and optimization of statistical sign language machine translation. *Machine Translation*, 26(4):325–357, mar.

Zaki, M. M. and Shaheen, S. I. (2011). Sign language recognition using a combination of new vision based features. *Pattern Recognition Letters*, 32(4):572–577.

8. Language Resource References

Slevinski, S. (2015). The signpuddle standard for signwriting text.

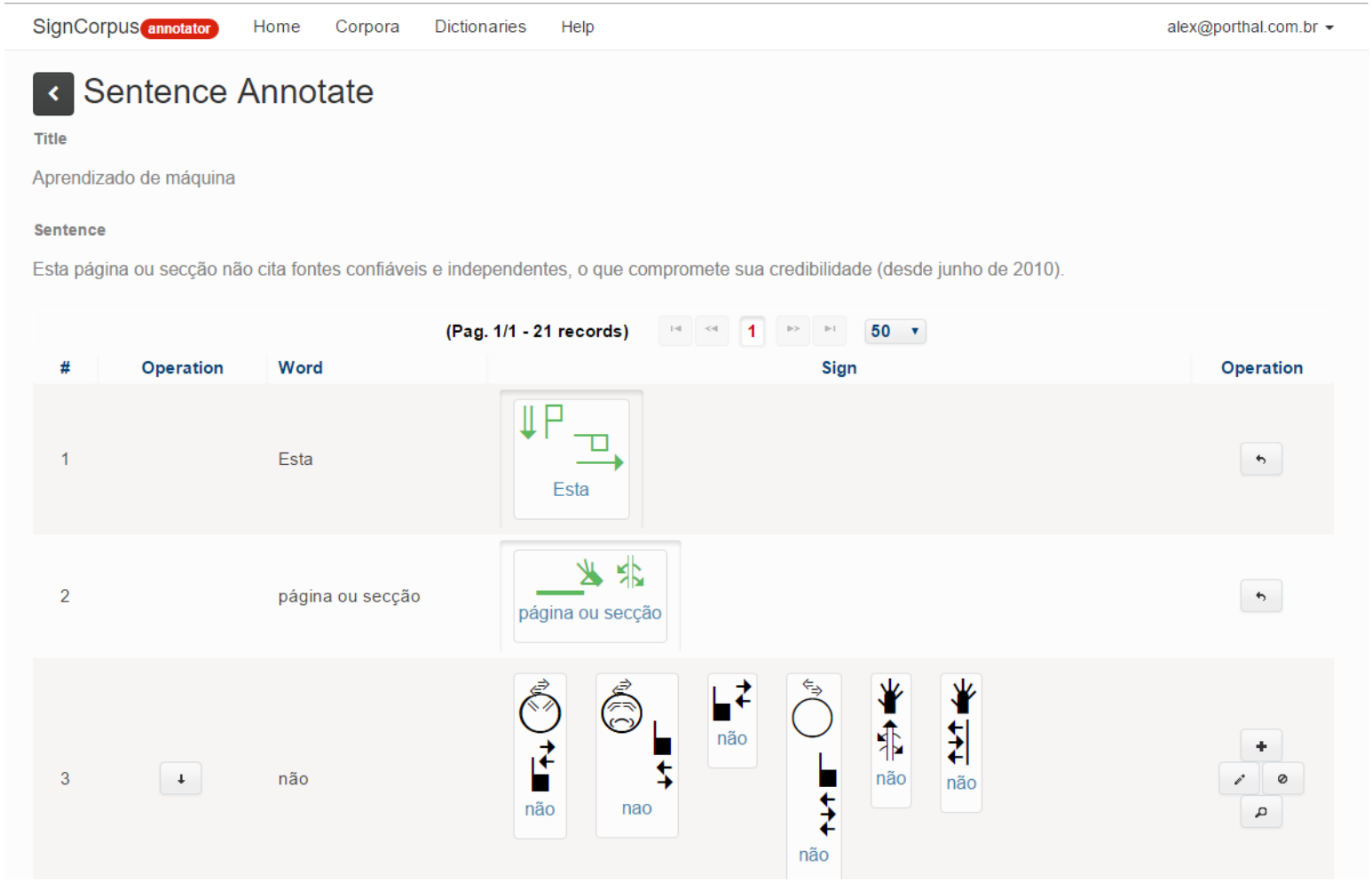

Figure 7: Screenshot of the sentence annotation interface.

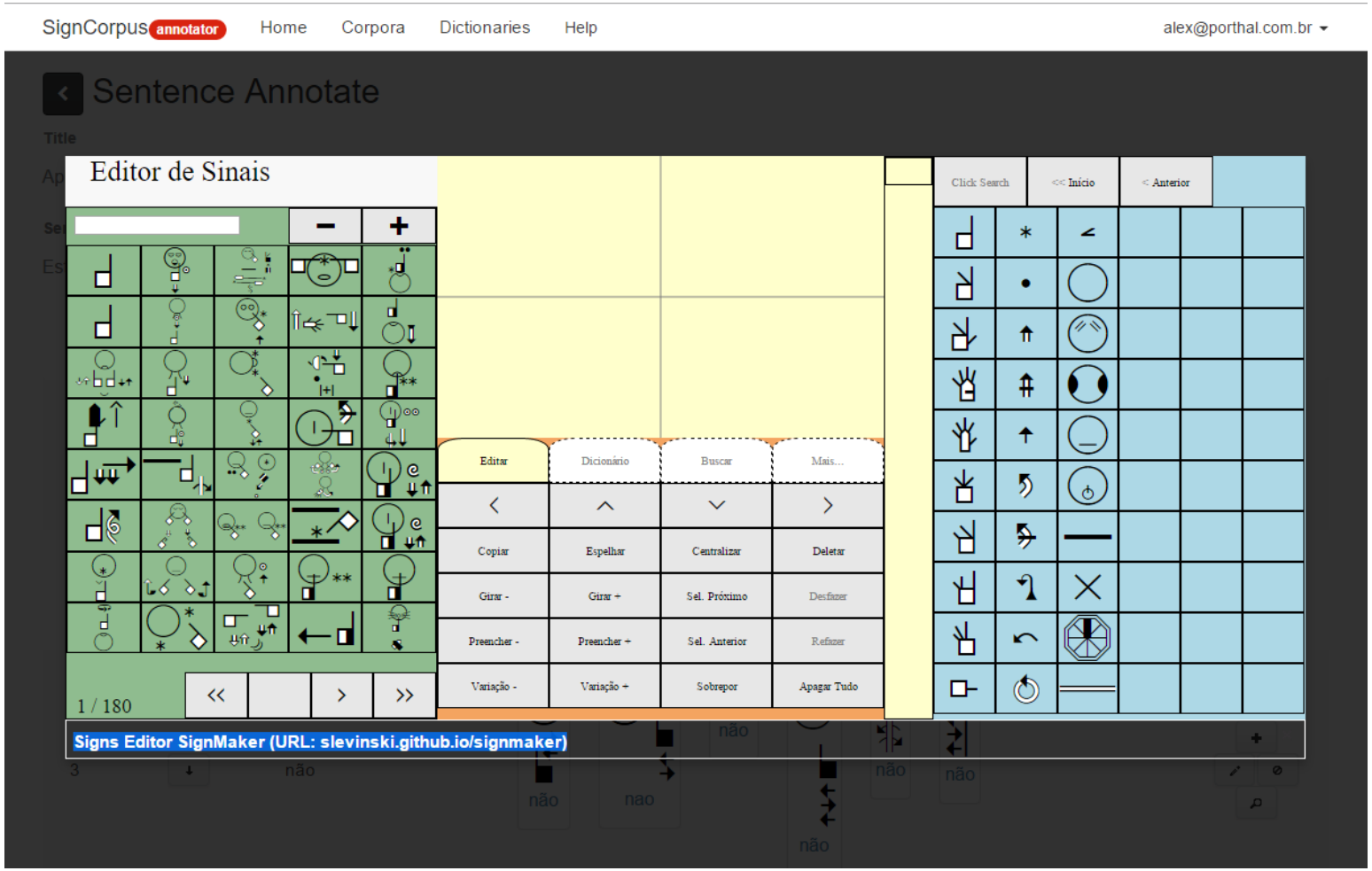

Figure 8: Screenshot of the SignMaker editor interface.