Exploring Challenges in Avatar-based Translation from European Portuguese to Portuguese Sign Language

Inês Almeida

October 2014

Abstract

Contrary to popular belief, sign language is not universal or mere mimicking of its country's spoken counterpart. As a consequence, Portuguese Sign Language (LGP) is a minority language without much investment.

To overcome language barriers between hearing and hearing impaired persons, we propose a machine translation approach from Portuguese to LGP. We further propose to present the result using a 3D avatar.

Being a very ambitious project, encompassing many disciplines, the present work aims to be a first step by presenting an overview of the areas involved and a base architecture.

In this paper, we describe the architecture and proof of concept implementation of the system along with a preliminary evaluation of the prototype.

We conclude that, though possible, this is a project with a considerable number of open and interrelated problems, being first and foremost the lack of linguistic studies for LGP.

Keywords Sign Language, Portuguese Sign Language, Natural Language Processing, Translation, Animation, Animation Synthesis

1 Introduction

1.1 Motivation

The Portuguese deaf community is composed by approximately 30.000 persons [Ins12] that need to recur to sign language to communicate.

These persons are isolated from their surroundings, as they communicate in a different language. Sign Languages (SLs) have their own vocabulary and grammatical rules that do not match closely the spoken language as the writing system does. SL is also different from country to country, making the community likewise isolated from other deaf communities and their available resources, arising the need for this work.

Although there are no official statistics, the general consensus is that many Portuguese deaf have moderate to high difficulties in understanding written content. The difficulty is due to the spoken language being the second language for the deaf. Control of a second language is non-trivial for many people even without the added difficulties of deaf condition.

A person's life is very dependent on communication. The lack of support for deaf persons can lead to isolation, less work opportunities and many other problems. This is true both in the physical world and, increasingly more, online. Therefore, considering the needs of the deaf in the new technologies of information and communication is very important.

1.2 Goals

The goals of this particular work can be summed in the following list:

1. provide an architecture overview with an high level description of each component,
2. provide a proof of concept of the system in the form of a prototype,
3. contribute with an open implementation of the architecture that can be used, worked upon and extended by others.

1.3 Overview

In this document, we start with some background in "Sign Language" (Section 2), presenting next "System Description" (Section 3) and "Architecture" (Section 4)

followed by "Implementation" (Section 5). We finish with "Conclusions and Future Work" (Section 7).

2 Sign Language

Sign Languages (SLs) pose additional difficulties to Natural Language Processing (NLP). A sign language is a very complex spacio-visual system to convey meaning with more than just iconic vocabulary. It has a grammar with morphological and syntactical rules, a phonology and rules of discourse.

SLs are not very established and their spatial grammar is still being actively studied by linguists. Being the Portuguese deaf community comparatively small, the poor development of these studies for the Portuguese Sign Language is a major concern[ACM94].

NLP techniques work either with hand-crafted rules or with large amounts of data with which to draw statistical conclusions. The first requires linguistic knowledge of the language, and the later considerable *corpus* of data that can be computationally processed. Both are a problem with sign languages that gets bigger the smaller the community is.

Models to denote sign language in written form, as to be computationally processed, are still being researched and developed. Some focus on the description of meaning (eg. gloss) while others focus on a more or less accurate physical description of the gesture, having different use cases (HamNoSys [Han04], SignWriting [Sut00], Movement-Hold [LJ89; JL11]).

A machine translation system needs to be considered and tuned for every language pair. The particularities of both languages must be taken into account, so as to decide on the best machine translation approach and natural language processing techniques.

3 System Description

The system's main goal is to translate from European Portuguese to Portuguese Sign Language.

The system should be able to accept isolated words and full sentences in text that are assumed to be correctly formed. For other use cases with different input, new conversion modules should be devised and coupled with the system.

We focus on an architecture that is capable of synthesizing the gestures from defined subunits and logic placement of the hands. We consider important that regular users may be able to insert new vocabulary in the system without any technical knowledge. It is, therefore, desirable that the gestures may be synthesized high-level descriptions and notations of sign language, but we do not focus further in designing an user interface for specifying the description.

The devised architecture should, as much as possible, be technology agnostic, as very different platforms (desktop, browser, mobile, vending machine...) are possible applications of a product deriving from this architecture.

A final and very important consideration is that all the system should be free as well as research and deployment friendly.

4 Architecture

Figure 1 presents the architecture overview, followed by an explanation of each component.

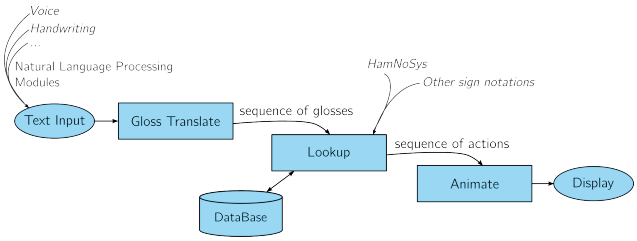

Figure 1: Proposed architecture

4.1 Gloss Translate

Translation approaches to sign language tend to follow the contemporary trends used for spoken languages [Por+14]. These other works do not usually justify their choice and it is not clear if there is a preferable method.

To decide between Statistical Machine Translation (SMT), Example-Based Machine Translation (EBMT), Rule-Based Machine Translation (RBMT) and a hybrid approach, there is the need to consider the cost of building the statistical or hand tailored data for the system to work with. The choice is dependent on the domain and task at hands.

Building and using corpora is a problematic matter for SLs, due to the difficulties of acquiring and storing the data. There is no standard writing or annotation system, or a way to easily process video or motion capture data into recognizable gestures.

To devise the system and to build a set of rules, there is also the need of linguistic studies and an understanding of the spatial grammar of the SL. For LGP, linguistic studies and references are scarce [ACM94; Bal10; Fer97].

Considering a non limited domain and the translation of full sentences, not just isolated words, we suggest a RBMT approach. We justify our option with the difficulties of using SL corpora and a better control of the structural transfer between sentences, given the big grammatical gap between LGP and Portuguese.

In Fig. 2 we present the overview of the RBMT pipeline with a deep structural transfer and analysis until a semantic level to better cope with the gap between the languages. This way, knowing the dependencies and semantic roles (subject, object, verb complements...), the structure and proper agreement can be better inferred in the target sign language.

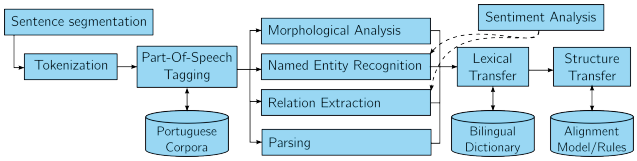

Figure 2: Overview of the translation pipeline

The system starts by splitting the input text into sentences that are further tokenized into words and punctuation.

Next, the tokens are parsed and POS-Tagged into a constituency tree. For the POS-Tagging, it is necessary a classified monolingual European Portuguese corpus.

The parsing into a constituency tree and, specially, into a dependency tree with verb as the root, allows the building of a semantic representation.

The result goes through several information extraction procedures, such as stemming, Named Entity Recognition (NER) and relation parsing. This information is fed into a semantic reasoning module to be further processed and passed to the animation stage. The sentiment analysis serves the purpose of inferring subjective information towards entities and the generality of the sentence, so that emotional animation layers and facial expression reinforcement can be added to the result.

Finally, the processed information can be used to accomplish the lexical transfer using a dictionary and the structural transfer using transfer rules and the semantic information. The semantic reasoning may need domain knowledge for gesture contextualization.

4.2 Lookup

The goal in this stage is to obtain a set of action ids for the animation from the sequence of already translated glosses. For this to happen, the actions that represent each gesture need to be associated with its meaning, so as to be retrieved from the sequence of glosses obtained in the gloss translate.

The difficulty in designing this step is derived from the fact that many Portuguese words and concepts do not have a one-to-one matching in LGP. One word in Portuguese can also lead to different words in LGP according to the context of the sentence.

It would be an advantage to be able to retrieve a gesture in LGP from any given sign language notation, besides its gloss meaning, and it would allow for more interoperability between sign language applications. Figure 3 shows an example of different gesture representations that relate to each other.

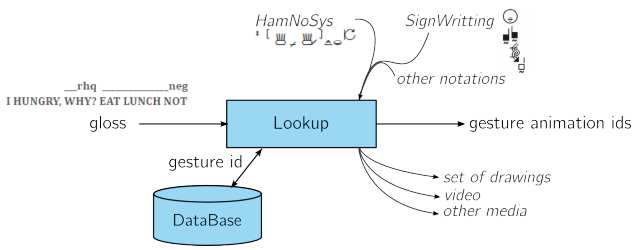

Figure 3: Dictionary lookup for different representations of a gesture

How to retrieve gestures by its features in a database is also an interesting problem with ongoing research [Eft+10].

4.3 Animate

This stage receives a sequence of actions to be composed into a fluid animation, along with a set of hints on how best to do so, for example, if the gestures are to be fingerspelled, or are adjectives to the same concept.

The animation stage is responsible for the procedural synthesis of the animation by blending gestures and gesture subunits together, but first, the actions matching each gesture need to be defined and the avatar needs to be set up.

Given the high number of different gestures and gesture variability within sentences due to context, it is logic to place and modify the gestures procedurally, to match their usage within sentences, and also to build the gestures themselves.

We propose an approach where gestures are procedurally built and defined from an high-level description, based on the following parameters identified in other works [LJ89; Lid03] as gesture subunits:

1. hand configuration,
2. hand orientation,
3. hand placement,
4. hand movement, and
5. non manual (facial expressions and body posture).

This approach allows non animators to build gestures and community dictionaries.

The base hand configurations are SL dependent and identified for LGP in [Bal10; ACM94; Fer97]. The parameter definition for orientation, placement and movement is often of relative nature. For example, gestures can be signed 'fast', 'near', 'at chest level', 'touching the cheek' and so on. The definition of speed is dependent on the overall speed of the animation, and the definition of locations is dependent on the avatar and its proportions.

We start by describing how to setup the character with a rig that has these issues in consideration.

4.3.1 Rig

To setup the character, we need an humanoid mesh with appropriate topology for animation and real-time playback and we need to further associate it with the mechanism to make it move, the *rig*. The rig is intrinsically associated with the procedural algorithms to pose the character and blend the poses into gestures and sentences. We suggest a regular approach with a skinned mesh to a skeleton and bones.

The bones should be named according to a convention for symmetry and easy identification in the code. For the arms and hands, the skeleton can approximately follow the structure of a human skeleton. The rig ideally should have an Inverse Kinematics (IK) tree chain defined for both arms rooting in the spine and ending in the hands. All the fingers should also be separate IK chains, allowing for precise posing of contacts.

The definition of the IK chains is essential for being able to easily place the hands in the desired final place both procedurally and manually. Ideally, the IK chains should consider weight influence so that bones closer to the end-effector (hands and fingertips) are more affected, and the bones in the spine and shoulder nearly not so. The rig should also provide a hook to control the placing of the shoulder.

The rig should make use of DOF, angle and other constraints for the joints so as to be easier to pose and harder to place in an inconsistent position when procedurally generating animation. The definition of joints that mimic those of a human body is a difficult problem with suboptimal results.

Finally, the rig should have markers for placement of the hands in the signing space and in common contact areas in the mesh. These markers ensure that the character mesh and proportions can vary without breaking the animation and that gestures can be defined with avatar dependent terms (eg. 'near', 'touching the nose').

The markers in the signing space can be inferred automatically using the character's skeleton measures [Ken02; Han04], forming a virtual 3D grid in front of the character as can be seen in Fig 4.

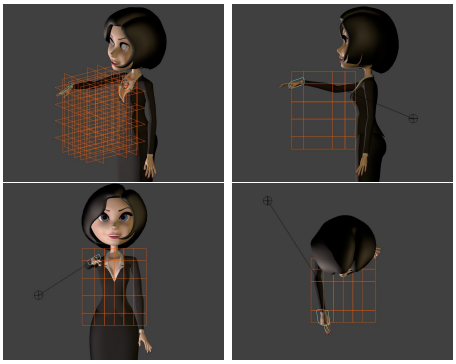

Figure 4: Virtual 3D marker grid defining the signing space in front of the character

The markers in the mesh, however, need to be defined manually and skinned to the skeleton in a consistent manner with the nearby vertices. Figure 5 shows a sample rig from our implementation, with key areas in the face and body identified by a bone with a position and orientation in space.

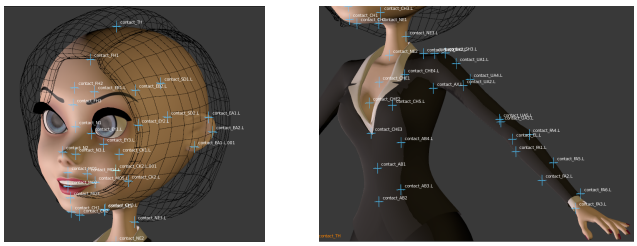

Figure 5: Definition of key contact areas in the rig

We found no study that identifies key areas in the non dominant hand, only mentions that this hand is

a very important articulating place, meaning that the dominant hand often contacts it in very precise places and ways (grasping, slapping, rubbing, stroking...).

4.3.2 Building the gestures

Having a working mechanism to pose the character, it is now necessary to record (*key*) the poses in a good timing to build a gesture. To accomplish this, there are two options: manually pose and keying using animation software, or acquire the data from motion capture and match it with the rig.

Whichever the keying methodology, all basic hand poses and facial expressions should be recorded and can then be combined given the high level description of the gesture. The description should specify the gesture using the mentioned parameters: keyed hand configurations, placement and orientation using the spatial marks and movement also using the marks and the overall speed of the animation.

Building a gesture that is avatar independent is achieved by real-time modifying of the animation curves and rig posing. The intersections defined by the grid from 4, in conjunction with marks from Fig. 5 define the set of avatar relative locations where the hands can be placed. Knowing the location where the hand should be, it can be procedurally placed with IK, guarantying physiologically possible animation with the help of other constraints. Figure 6 shows the result of hand placement in a key area using two distinct avatars with significantly different proportions.

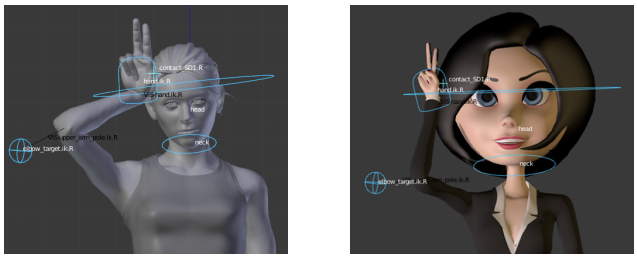

Figure 6: Avatar variability within an action using the key areas

While this approach works well for static gestures, several problems appear when introducing movement. Gestures can change any of its parameters during the realization, requiring a blending from the first definition (of location, orientation, configuration...) to the second. The type of blending is very important for the realism of the animation. Linear blending between two keys would result in robotic movement. The principles of animation, such as easing and anticipation need to be embedded in the animation curves.

Linear movement in space from one key location to another will also result in non realistic motion and even in serious collision problems when, for example, making a movement from an ear to the other. This is a problem of arcs. We found no study for this problem. Kennaway's research [Ken02; Ken04; KGZ07; Ell+07] mentions the possible need of collision handling.

Additionally, more movements need to be defined in order to accommodate other phenomena, such as finger wiggling and several types of hand waving.

4.3.3 Blending the gestures

Being able to build the individual gestures as needed for any given sentence, synthesizing the final fluid animation is now a matter of grammatically agreeing the gestures in space, of realistic interpolation of keys in time, similar to how it must be done within the building of gestures, and of blending actions with each other in a non-linear way.

The system needs a reasoning module, capable of placing gestures grammatically in the signing space, making use of the temporal line, entity allocation in space and other phenomena typically observed in SLs [Lid03].

The interpolation between animation keys is given by a curve that can be modeled to express different types of motion. The individual actions for each gesture should be concatenated with each other and with a 'rest pose' at the beginning and end of the utterance.

The animation curves should then be tweaked, following the principles of animation, easing in, defining arcs, and slightly exaggerating the defining keyframes as to better convey the expression of the gesture.

Counter animation and secondary movement is also very important for believability and perceptibility. For example, when one hand contacts the other or some part of the head, it is natural to react to that contact, by tilting the head (or hand) against the contact and physically receiving the impact. Besides the acceleration of the dominant hand, the contact is mainly perceived in how it is received, being very different in a case of gentle brushing, slapping or grasping. This may be the only detail that allows distinguishing of gestures that otherwise may convey the same meaning.

Finally, actions need to be layered in the for expressing parallel and overlapping actions. This is the case

for facial animation at the same time as manual signing and of secondary animation, such as blinking or breathing, to convey believability. The channels used by some action may be affected by another action at the same time. The actions need to be prioritized, taking precedence in the blending with less important, or ending actions.

NPR rendering can be used to reinforce key parts of a gesture, by making use of the animation curves as input to the shaders. Blurs and stark line strokes in defining features of a gesture on important keyframes are suggested.

4.4 Running the System

Considerations on how and where the system will run should be taken throughout all the pipeline.

From our research, we take that interesting applications of the system range from web navigation to television and public interfaces, considering also mobile devices in real life scenarios. The supported technologies for these platforms vary a lot and the processing power and physical memory are not high in most cases.

For maximum portability, we suggest the OpenGL rendering subsets: OpenGL for Embedded Systems (GLES) and WebGL. A browser plugin, using web technology, would also be portable solution, though not a native mobile application or proper for set top boxes.

A distributed architecture needs to be considered, where a client (the user's device) would make a request and receive the animation as reply from the server. The avatar(s) should typically be stored in the client side. As for the data (corpora, animation dictionaries...), if it is server-side, it can be shared and improved in a community effort, if it is client-side, the user is free to edit it and tailor if for his personal needs, as well as offline access.

5 Implementation

As proof of concept of our architecture and playground for experimentations, we implemented a prototype of the system. The following sections describe our choices and particularities of the system. The system can be downloaded[1].

[1] **url {http://web.ist.utl.pt/~ist163556/pt2lgp}**.

5.1 Blender

We chose Blender as the 3D package to use for modeling and animation. The main reasons for our choice are the quality and feature set of the 3D tools and that it is an open-source project with an also open community of developers and artists.

Blender has a powerful and flexible API paradigm, that allows accessing and operating on all the data (animation, mesh...) via scripting.

5.2 Addon Philosophy

The choice of Blender as a modeling and animation tool implies some others, such as how to interface our system with the 3D model and the animation data and how to display the animation.

Though it would be possible to implement a standalone system that interfaces with Blender, we saw no reason to deviate from Blender's addon philosophy. Blender offers a Python API for scripts to interact with the internal data structures, operators on said data, and with the interface. The interface can be customized with new panels and buttons though at the moment it does not allow the construction of entirely new widgets or editors.

Blender also offers the infrastructure to easily share and install addons.

With these considerations, the prototype was implemented as an addon, with all the logic, NLP and access to the rig and animation data done in Python. The interface is a part of Blender using the pre-existing widgets and the avatar is rendered in real-time using the viewport renderer.

5.3 Natural Language Processing and Translation

Following the choice of Python to implement the system's logic, we chose to use the Natural Language ToolKit (NLTK)[2][BKL09] for NLP tasks. An overview of the implemented pipeline is shown in Fig 7.

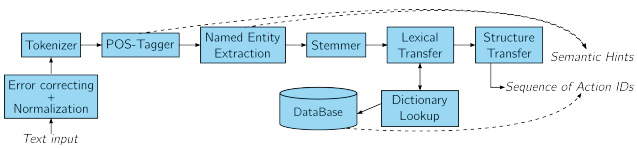

Figure 7: Natural Language Processing pipeline used in the implementation

[2] URL: `http://www.nltk.org` (visited on 09/2014)

- **Error correcting and normalization** A step that enforces lowercase and the use of latin characters. Common spelling mistakes in the words used for the test cases are also corrected.

- **Tokenizer** The input string is split into sentences and then into words. The tokenizer, provided by NLTK, uses Portuguese language training. Example of a tokenized input: `['o', 'joão', 'come', 'a', 'sopa']`

- **POS-Tagger** We make use of NLTK's n-gram taggers, starting with a bigram tagger, with a backoff technique for an unigram tagger and the default classification of 'noun', being the most common class for Portuguese. We used the treebank 'floresta sintá(c)tica' corpus [Afo+02] for training the taggers. Using the same example, the result would be: `[('o', 'art'), ('joão', 'prop'), ('come', 'v-fin'), ('a', 'prp'), ('sopa', 'n')]`

- **Named Entity Extraction** We apply NER for proper names of persons by matching against a list of common Portuguese names.

- **Stemmer** As a form of morphologic parsing, we apply a stemmer that identifies suffixes and prefixes to use as an adjective or classifier to the gloss. The result is treated as a whole. For example, the Portuguese word *coelhinha*, roughly meaning 'little female rabbit', is understood, by its suffix, to be a female and small derivation of the root *coelh(o)*.

- **Lexical Transfer** The expanded and annotated list of words are converted to their corresponding glosses using a dictionary lookup. This works by referencing to the database described below, resulting in items such as `['GLOSS', ['SOPA']]` and `['FINGERSPELL', ['J', 'O', 'A', 'O']]`.

- **Structure Transfer** The prototype supports reordering of adjectives and quantities to the end of the affecting noun, for example the input *dois coelhos* ('two rabbits') would result in `[['GLOSS', ['COELHO']], ['NUMERAL', ['2']]]`.

  The prototype also supports basic reordering of sequences of 'noun - verb - noun', in an attempt to convert the SVO ordering used in Portuguese to the more common structure of OSV used in LGP.

5.4 Lookup

The animation lookup, given a gloss, is done via a JSON file mimicking a database. This was thought to be the simplest and more portable solution for the data, without the strong need to recur to more solid ways of persistence in a proof of concept stage.

As to the design, the database consists of a set of *glosses* and a set of *actions*. The action ids are mapped to blender actions, that are in turn referenced by the glosses. One gloss may link to more than one action, that are assumed to be played sequentially. This design can be inspected in Figure 8, where it can be seen that *coelho* ('rabbit') has a one-to-one mapping, that *casa* ('house') corresponds to one action and that *cidade* ('city') is a composed word, formed by *casa* and a morpheme with no isolated meaning.

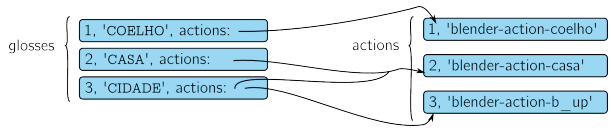

Figure 8: Database design used in the implementation

5.5 Animation

For the animation pipeline, we start by setting an avatar for use in the system by rigging and skinning. We chose *rigify* as a base for the rig, that then needs to be extended with the spatial marks.

All the 57 different hand configurations for LGP were manually posed and keyed from references gathered from [Bal10] and the Spread the Sign project.

The animation is synthesized by directly accessing and modifying the action and f-curve data.

We always start and end a sentence with the rest pose. For concatenating the actions, we *blend* from one to the other in a given amount of frames by using Blender's Non Linear Action (NLA) tools that allow action layering. Channels that are not used in the next gesture, are blended with the rest pose instead.

Figure 9 illustrates the result for the gloss sentence '`SOPA J-O-A-O COME`'.

We adjust the number of frames for blending according to the hints received. For fingerspelling mode, we expand the duration of the hand configuration (that is originally just one frame) and blend it with the next fingerspelling in less frames than when blending between normal gloss actions. We also expand this duration when entering and leaving the fingerspell. For the

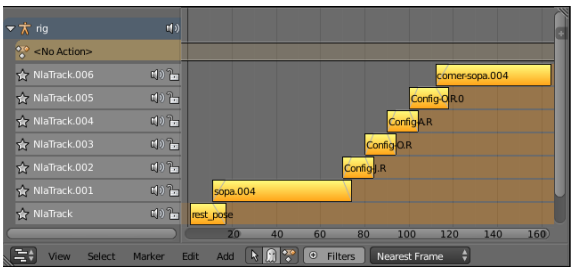

Figure 9: Action layering resulting of a translation

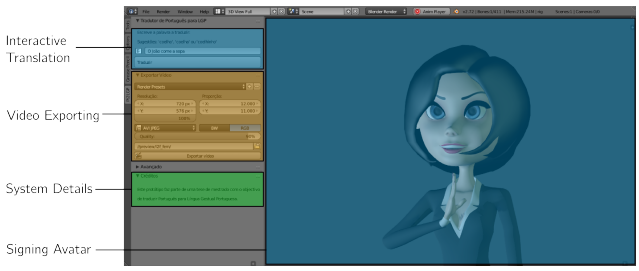

Figure 11: User Interface for the prototype

fingerspelling mode, we needed to directly access the fcurve data, as this can not be done with the NLA tools.

5.6 Avatar

Our implementation allows freedom in the choice of avatar, needing only that the character be imported into Blender and skinned to a *rigify* armature.

With the manual definition of key locations as explained in "Rig" (Section 4.3.1), avatar variability is supported. Throughout the development of the proof of concept, several characters were tested with success, with examples in Fig 10. From the left to the right, there is *Sintel* and *Blenderella*, made freely available by the Blender Foundation, next, a model from the BlendSwap community[3] where it is easy to search by rigify and freely licensed characters, and finally, the rightmost character, *Catarina*, acquired by the L^2F group. A workflow using MakeHuman[4] was also tested with success.

Figure 10: Example of supported avatars

5.7 User Interface

The interface, in its most basic form, consists of an input text box, a button to translate, and a 3D view with the signing avatar. The 3D view can be rotated and zoomed, allowing to see the avatar from different perspectives.

The breakdown in Fig. 11 shows the main translation interface in blue.

[3] `http://www.blendswap.com/blends/view/13464`
[4] URL: `http://www.makehuman.org` (visited on 09/2014)

Additionally, we provide an interface for exporting video of the signing (indicated in orange) and a short description of the project (in green).

It is still possible to use all the functionalities of Blender, thus making advanced usage of the system.

6 Evaluation

The preliminary evaluation was conducted in an informal manner, both by observation and by collecting feedback from the deaf communities of two Portuguese deaf associations.

6.1 Usefulness

As the prototype is a proof of concept of a translation application for the deaf, it was a priority, from the beginning, to query for the usefulness and usability of the system. Two deaf associations were reached for feedback, being asked for comments on the whole idea behind this work and if and how an application such as this would be useful.

Both associations are skeptical towards the possibility of achieving accurate translations or of animating enough vocabulary for a final product, but the feedback was positive for the idea of an application that would translate to LGP, even if just isolated words, or other limited use.

6.2 User Interface Usability

The interface of the prototype was tested with hearing and deaf people, by observation and by requesting the execution of tasks.

The interface proved to be usable, but not robust enough for some people. The interface is tightly integrated with Blender's and was deemed too easy to change accidentally without the user having the knowledge to return it to the initial state (eg. changing the

layer with keyboard or moving the viewport resulting in loosing the avatar).

Although not ideal, the interface within Blender was a good choice for a prototype, being the mentioned problems minor.

6.3 Translation Quality

The correctness and perceptibility was evaluated by 6 adult deaf persons and interpreters.

The avatar was set to play the translations for 3 words and a full sentence and the viewers were asked to say or write in Portuguese what was being signed, with no previous information about the possibilities.

All of the testers recognized correctly the results, without hesitations, saying that the signs were all very clear and only lacking facial reinforcement to be more realistic. Feedback on clarity and readability was very positive.

6.4 Adequacy of the Avatar

The feedback from the deaf testers regarding the avatar looks was very positive. There were no negative comments besides the observation that there is no facial animation.

All hearing testers were also highly engaged with the system, testing multiple words and combinations, frequently mimicking the avatar.

The interest and attention observed, indicates that users had no difficulty engaging with the avatar and found it either neutral or appealing. When asked about it, the answers were positive and the gesture blending and transitions, when noticed, was commented to be very smooth.

Sometimes the animation was deemed too slow or too fast. It is possible that this is a subjective matter, or that the animation generation should take play speed in consideration according to the expertise of the user.

7 Conclusions and Future Work

We have presented a system architecture that generates fluid animation of LGP utterances given a text input in Portuguese. We have further tested this architecture with a proof of concept implementation and a preliminary evaluation with the deaf community.

We conclude that there is still much to be done before we can reach the stage of having a working product at the service of the deaf community. Nevertheless, we observe that the LGP is a natural language possible of being translated using MT approaches. The translation process, however, poses additional and specific difficulties due to the spatial nature of sign languages and the significantly different grammars of Portuguese and LGP. Sign languages have many important particularities that need to be considered in NLP.

For the use of any translation approach, we need more corpora and linguistic studies on LGP. Besides being fundamental to any rule-based approach, linguistic studies are important by themselves, having socio- and psico-linguistic goals of their own and contributing to the standardization and comprehension of the LGP.

If more linguistic studies are essential to a RBMT approach, quality corpora is the key to the SMT approaches, and are also missing.

Linguistic studies and corpora are both important to any translation approach as they usually interrelate in hybrid approaches in order to improve the final results. Any NLP application would benefit with further investment in linguistic studies and corpora.

Building quality corpora is a difficult task in sign languages and, consequently, in LGP. There needs to be a choice regarding how to annotate the corpora. A choice needs to be done regarding the annotation system that range from gloss, that is meaning based to phonetic transcription. It can be annotated manually or with extracted parameters from video and motion capture.

The experience with this work was positive. There is a great variety of research, but still a lot of ground to explore on this subject. The preliminary tests with the proof of concept showed that a working product would be highly desirable and would improve the lives of many.

The next steps would be to go deeper in each section, reaching out for people experienced in their respective fields, to gather not only theoretical knowledge but, especially, practical.

7.1 Contributions

This thesis opens way for future works that can improve the current situation as a way to aid in communication and to increase accessibility where it would not be possible or too costly by other means.

We list our contributions as follows:

1. presentation of related work and necessary background studies
2. architectural overview of a system to perform MT from Portuguese to LGP and present the synthesized animation using a 3D avatar.
3. proof of concept of the architecture with an open implementation that can be further worked upon
4. list of problems that need further research
5. demonstration of the prototype at the European Researchers Night in Lisbon[5]
6. demonstration the prototype and the animation synthesis pipeline at Blender's Annual Conference in Amsterdam[6]

References

[Afo+02] S Afonso et al. 'Floresta Sintáctica: A treebank for Portuguese.' In: *LREC* (2002), pp. 1698–1703.

[ACM94] M.A. Amaral, A. Coutinho and M.R.D. Martins. *Para uma gramática da Língua Gestual Portuguesa*. Colecção universitária. Caminho, 1994.

[Bal10] Ana Bela Baltazar. *Dicionário de Língua Gestual Portuguesa*. Porto Editora, 2010.

[BKL09] Steven Bird, Ewan Klein and Edward Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[Eft+10] Eleni Efthimiou et al. 'Dicta-sign–sign language recognition, generation and modelling: a research effort with applications in deaf communication'. In: *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*. 2010, pp. 80–83.

[Ell+07] R. Elliott et al. 'Linguistic modelling and language-processing technologies for Avatar-based sign language presentation'. In: *Universal Access in the Information Society* 6.4 (Oct. 2007), pp. 375–391. DOI: 10.1007/s10209-007-0102-z.

[Fer97] A.V. Ferreira. *Gestuário: língua gestual portuguesa*. SNR, 1997.

[Han04] Thomas Hanke. 'HamNoSys-representing sign language data in language resources and language processing contexts'. In: *LREC* (2004).

[Ins12] Instituto Nacional de Estatística, I.P. *Census - Final results. Portugal - 2011*. 2012, p. 558.

[JL11] Robert E Johnson and Scott K Liddell. 'A Segmental Framework for Representing Signs Phonetically'. In: *Sign Language Studies* 11.3 (2011), pp. 408–463. DOI: 10.1353/sls.2011.0002.

[Ken04] J. R. Kennaway. 'Experience with and requirements for a gesture description language for synthetic animation'. In: *Gesture-Based Communication in Human-Computer Interaction*. Springer, 2004, pp. 300–311.

[Ken02] J. R. Kennaway. 'Synthetic animation of deaf signing gestures'. In: *Gesture and Sign Language in Human-Computer Interaction*. Springer, 2002, pp. 146–157.

[KGZ07] J. R. Kennaway, John Glauert and I. Zwitserlood. 'Providing signed content on the Internet by synthesized animation'. In: *ACM Transactions on Computer-Human Interaction* 14.3 (Sept. 2007), 15–es. DOI: 10.1145/1279700.1279705.

[Lid03] Scott K Liddell. *Grammar, gesture, and meaning in American Sign Language*. Cambridge: Cambridge University Press, 2003, p. 400.

[LJ89] Scott K Liddell and Robert E Johnson. 'American Sign Language: The Phonological Base'. In: *Sign Language Studies* 1064.1 (1989), pp. 195–277. DOI: 10.1353/sls.1989.0027.

[Por+14] Jordi Porta et al. 'A rule-based translation from written Spanish to Spanish Sign Language glosses'. In: *Computer Speech & Language* 28.3 (May 2014), pp. 788–811. DOI: 10.1016/j.csl.2013.10.003.

[Sut00] Valerie Sutton. *SignWriting*. Deaf Action Committee (DAC), 2000.

[5] http://ec.europa.eu/research/researchersnight
[6] https://www.blender.org/conference