

SOUTH AFRICAN SIGN LANGUAGE ASSISTIVE TRANSLATION

Lynette van Zijl
Department of Computer Science
University of Stellenbosch
P/Bag X1, Matieland, 7602,
Stellenbosch, South Africa
email: lynette@cs.sun.ac.za

Guillaume Olivrin
Meraka Institute
Pretoria, South Africa
email: golivrin@csir.co.za

ABSTRACT

We describe a novel approach and practical solution to an interactive sign language machine translation system from English to South African Sign Language. We interface the system with the GNApp application, which is an augmented and alternative communication front-end, in order to resolve word sense ambiguities. This enhances the correctness of the rule-based translation system. We also argue that the GNApp interface can be applied to assist the South African Sign Language to English translation, without the need for a visual sign language recognition component.

KEY WORDS

Human-computer interface, machine translation, natural language processing, sign language.

1 Introduction

Advances in assistive technologies can bring benefits to daily communication of hearing and Deaf people. Such technologies include voice recognition, voice synthesis, signing avatars and machine translation (MT) [1, 2]. One of the most challenging aspects for assistive sign language tools is to interactively input signs into a computer, as sign languages are visual-spatial languages with no simple textual representation. Sign language recognition systems are still computationally intensive and lack in accuracy and coverage. Hence, assistive technologies based on translation systems usually perform better from text to sign (TTS), rather than from sign to text (STT). Although notations (such as HamNoSys [3] and SignWriting [4]) exist to describe sign languages, these are typically either complex to input with standard keyboards, or lack in the linguistic accuracy needed for machine translation.

In this paper, we discuss the combination of the GNApp (AAC) grid interface [5] with the South African Sign Language Machine Translation (SASL-MT) system [7]. This combination has three main advantages in our case:

- The AAC interface ameliorates the problem of passing tagged sentences with word categories and part-of-speech (POS) to MT systems, which improves the translation accuracy (see Section 2); this point forms the main focus of this article;



Figure 1. The GNApp AAC interface to compose English sentences annotated with POS.

- the AAC interface presents an opportunity to capture sign languages when *word-signs* are associated with input buttons on the various sheets;
- the AAC interface makes it possible to capture various instances of languages. The iconic and ideographic interface presents an input modality for people with low literacy in any language (for example, English, Afrikaans, and Zulu).

The rest of this article is organized as follows: in Section 2 we describe the issues surrounding machine translation (of sign languages) in our system. We then show the framework design and the use case of our system in Section 4, followed by a description of our experiments in Section 5. We discuss various disambiguation strategies that combine the user interaction and machine translation process in Section 6.1, consider future work in Section 6, and conclude in Section 7.

2 Machine Translation in SASL-MT

The SASL-MT system is a prototype MT system for translating English text to South African Sign Language (SASL). It is a rule-based system, based on an Earley-type tree-adjointing grammar (TAG) parser [8, 9]. Given the translated sentence, a signing avatar signs it in SASL. Rule-based machine translation systems in general do not perform as well as the more modern statistical machine translation systems [2]. However, the latter require large bilingual corpora to train on, and in the case of SASL, such a corpus is not available. Hence, at the time of implementation, the rule-based approach was the only feasible option.

One particular disadvantage of the rule-based translation approach lies in its necessity for part-of-speech (POS) tagging and word sense disambiguation (WSD) [10]. In POS tagging, ambiguities arise when a single word can have more than one POS tag – for example, the word ‘bark’ can be a noun or a verb. In word sense disambiguation, ambiguities can result from the context in which a word is used in a sentence. For example, in the sentence ‘*A large dog has a loud bark.*’ the word ‘bark’ is a noun which indicates a sound, and not the outer layer (bark) of a tree. Hence, the accuracy of the rule-based translation is dependent on the ability of the parser to correctly identify the relevant POS and the correct word sense for every word in the input sentence. In natural language processing systems, these problems are typically solved based on statistical methods. Many sophisticated solutions exist for POS tagging and for WSD. However, these solutions may still not be able to solve certain cases; they can be computationally expensive, and require supportive linguistic corpora. The SASL-MT prototype system therefore does not currently support POS tagging or WSD.

In AAC grid systems such as GNApp, however, many such ambiguities are automatically resolved by the input method. For example, one of the goals of AAC systems is to minimize the keystrokes, so that often one input button is associated to a complete POS-tagged sentence with the correct sense for each word. In other instances, input icons in the grid are organized by concept (for example, {Nouns}, {Verbs}, {Descriptions}, {Social} and {Misc}). Therefore, it is a natural organization to group concepts representing a given POS in the AAC grid. In this article, we concentrate on the MT aspects rather than the issues surrounding the AAC grid. We note, however, that there are standard ways of handling issues like scalability, word prediction, and other similar issues for the AAC grid [11].

We describe in this work how we combine the GNApp AAC grid input interface to the prototype SASL-MT system, and analyze the results. As expected, such a translation system performs better than a stand-alone system, with much less computational overhead and increased accuracy. We concede that this is not a general solution, but argue that it is a current, feasible and cheap solution for assistive technology systems currently hampered by ambiguities in MT systems.

An additional benefit to the combination of GNApp and SASL-MT is that it provides a practical and feasible possibility for translating from SASL back to English. In our rule-based system, this translation would require the definition of a tree grammar for SASL, and adaptation of the current rules. As SASL is linguistically under-investigated, the construction of the tree grammar is not trivial, and it is one of the major future planned developments for our project.

3 Use case and scenario

The current system is useful to a hearing person who wishes AAC support to communicate with a Deaf person (see Figure 2). In this scenario, an English speaking person inputs sentences to be interpreted in SASL. The emphasis is therefore on the target language model (SASL) and on delivering a clear and intelligible message as faithfully as possible to the original sentence. Although the meaning is transferred through an English gloss, the fact that the target structure is SASL, the home language of the Deaf user, already facilitates the communication. Ultimately the message will be delivered more naturally by a signing avatar [14]. We shall present further scenarios for future work in Section 6.

4 Framework and Design

Our system is currently composed of the GNApp framework and the SASL-MT machine translation component.

The GNApp framework provides an input interface for composing sentences in English. Icon sheets are customizable and can hold grids of photos, drawings and text in English. A sheet with SignWriting [4] and English words is shown in Figure 1: the colours represent word categorization according to the Fitzgerald key [12, 13] and POS tags are produced during input.

The SASL-MT system exploits the POS tags to parse and transform the source language structure into TAG syntax trees. Word categorization as colours are only useful to the AAC user when interactively composing sentences from a palette of vocabularies. The translation rules of the MT-system do not exploit semantic cues to produce the target sentence.

The GNApp framework was originally designed to integrate the grid interface with a choice of Text-to-Speech (TTS) engines. We adapted the system for Text-to-Sign Language communication purposes. Firstly, we added POS notations to the XML schemas of the grid sheets, namely: [Pron], [PropN], [N], [A], [V], [Part], [Adv], [Det] and [Prep]. Secondly, we extended the framework with the new `SignFacade` class which implements the TTS interface. Thirdly, we added a new `ButtonHandler` event called `signText` which is invoked upon hitting the *Sign it!* button located at the bottom right in Figure 1. The gen-

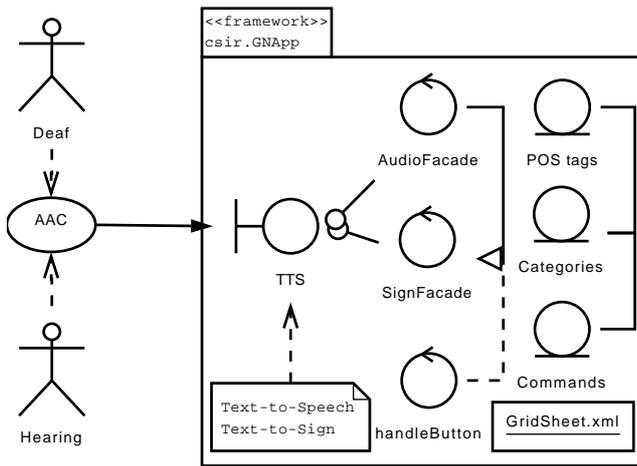


Figure 2. The GNApp AAC framework and TTS interfaces.

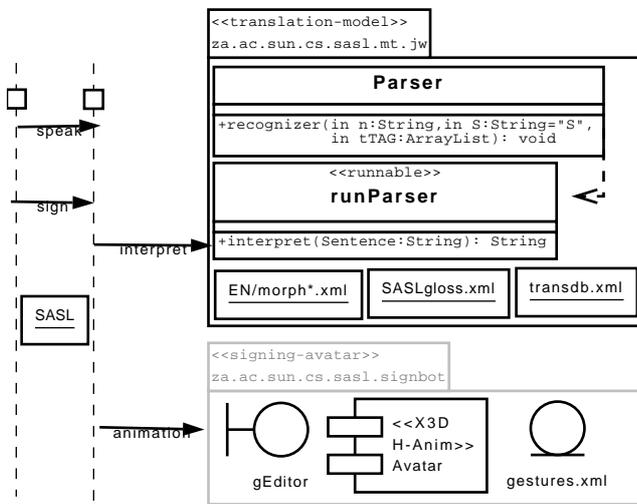


Figure 3. The SASL Machine Translation and Signing Avatar components.

erated event calls the sentence parser of the SASL-MT component. The `Parser` generates TAG trees for the `translate` method to transform using synchronous TAG rules. This last step produces a SASL-structured sentence with an English gloss. Vocabularies for the new AAC device are stored as follows:

- GNApp XML *grid* sheets specify POS-tagged vocabularies and sentence-parts;
- GNApp predicted vocabulary is stored in a separate plain text file with no POS annotations for words that are entered letter by letter and predicted;
- morphemic and syntactic databases are used by the SASL-MT parser; and
- the `GlossDB` and `TransRuleDB` XML files store the

Number of initial TAG trees

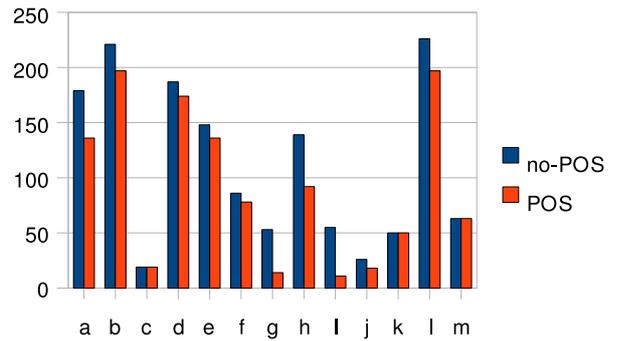


Figure 4. Number of initial TAG trees built by SASL-MT for each sentence.

list of tokens and rules available to perform translations.

All these datasets must concord in sharing the same vocabularies and sets of structure.

The reader may note that our prototype MT system has various limitations. From a linguistic point of view, the small set of translation rules requires considerable extension. This of course means that many more English structures can be parsed than can be translated. Therefore, for the system to be used optimally in an AAC scenario, only sentences that can be parsed and also translated should be made available on the AAC interface.

5 Experiments and results

The experiments consist of 12 sentences captured using the GNApp interface. Each sentence is sent to the SASL-MT system with and without POS tags to compare the gain in terms of performance, accuracy and coverage for the translation task. According to [9]:

[...] simply providing the correct word categories for each of the words in the sentence [...] reduces the number of parses generated by the parser; and also reduces the likelihood of any ambiguities arising in the parses.

The results recorded here give the relative merits of using POS annotations versus no POS annotation whilst composing sentences.

Figure 4 shows the number of initial elementary TAG trees built from the sentences and is based on the POS categories available for each word. This indicates that, by specifying the POS categories for each word, we can reduce the memory load and associated computations. We can observe from the data that there is a correlation between the number of partial trees loaded in memory and the number of POS ambiguities in a sentence. There is also a strong

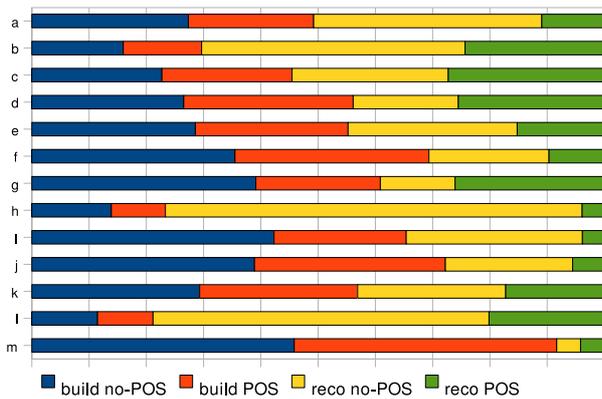


Figure 5. The parsing time ratios for SASL-MT build and recognition phases are shown. The total parsing times of each sentence has been normalized for comparison.

correlation between the gain in recognition time with POS notation, and the number of POS ambiguities.

The TAG trees building phase is the most time consuming according to Figure 5. There is still however, a substantial number of trees that need to be built even for simple sentences. One of the reasons is that for each specified POS category there are various subcategories. This calls for discussion on whether the words and sentences should be further tagged and in more detail.

Figure 5 shows that, in terms of bringing down the MT processing time, the POS notation greatly helps during the recognition phase.

The conclusion from these data is that, although there are gains through POS annotation, these require improvement. Complementary actions are proposed below in order to make MT in AAC worthwhile.

5.1 Evaluation

The parse is the critical step in the MT process that requires the most resources, time and computational power. Building TAG trees is expensive, while assembling and selecting the best candidate is achieved rapidly. The translation processing is very fast since this is only a matter of applying a few rules to these trees.

In order to evaluate our system performance, we look at it from the MT as well as the AAC perspective. In this study we have focused on evaluating the MT system according to the following criteria:

- the coverage of the input sentence,
- the accuracy in interpretation, and
- the time for parsing.

As far as the AAC interface is concerned, the criteria are:

- the input rate, both for selecting words and for composing full sentences,
- customization of the AAC grid,
- the clarity of the delivered message; that is, the accuracy of interpretation,
- the language coverage and forgiveness; that is, the robustness and coverage of the parser, and
- the choice of input and output modalities; this includes voice, a signing avatar and language representation through icons and ideographs.

As far as input rate is concerned, we designed the AAC grid sheets so as to make the input of words and phrases with POS annotation transparent and intuitive, based on the word categories and the text output for each button hit. Section 6.1 will look at strategies for interface and interaction based disambiguation strategies. Customization of the grid sheet is built into GNApp and made configurable through XML files. The need to reduplicate the vocabulary and structures in the MT system can later be addressed in the authoring mode of the GNApp grid sheets. The clarity of the delivered message is done by a controlled set of translation rules that only apply to specific structures. However, this has the consequence of reducing the coverage and production of SASL structures. The coverage of English sentences as input is greater than what is available to produce sentences from the available grid sheets and is customizable. The input and output modalities are inherent to the GNApp framework and provide a variety of input methods: mouse, keyboard shortcuts and scan line for inputs, text display, pre-recorded sounds, and synthesized voices. A signing avatar for output of SASL sentences will be integrated into the system in the near future.

6 Future Work

6.1 Disambiguation

Disambiguation can be improved on two levels: firstly, by eliciting more or better information from the user through the AAC interface and secondly, by improving certain aspects of the MT system. We discuss both of these options below.

6.1.1 Reducing ambiguity through the AAC interface

AAC interfaces present the advantage of giving the initiative to the user to compose English sentences according to a finite, well-defined vocabulary set. It is possible to take advantage of this user interaction to further disambiguate the input sentence that must be interpreted.

Soft approaches for ambiguity resolution can be employed thanks to the AAC interface. This should typically address MT disambiguation problems at semantical and sometimes syntactical levels.

The first strategy we discussed was the word categorization and word colouring. Grid organisation and vocabulary disposition on the sheet is a second way of logically and cognitively organising the building material of sentences. A third approach is the hierarchy of grids created by their interlinkage.

Usability evaluation is needed to measure the merit of these strategies for composing sentences on GNApp. In particular, to what degree does color coding, sheet presentation and grouping help to reduce MT and language ambiguities?

There are many ways to express the same concept, and the best way to put the message across is open to interpretation and preference. Yet translation rules could be specified and prioritized for any given vocabulary AAC sheet. This would help with rule disambiguation and keep the production scope of the target language narrow and precise. In cases where multiple and alternative message interpretations are possible, it may be possible to exploit user interactivity with the AAC interface to choose the best candidate:

This [*linguistic*] problem arises from the data used to derive rules for the translation system. In particular, it arises because of seemingly conflicting translations in the SASL phrase book. [9]

In a case like this, a disambiguating question could be set up within the rule based system to make an informed choice. This question will be asked to the AAC user in order to choose which message interpretation to send to the output modality.

6.1.2 Reducing ambiguity through software design

Tagging the input vocabulary sheets with POS can be done exhaustively and economically. We saw in section 5.1 that too much or too little POS annotation can be penalising. How can we experimentally strike a balance between over-tagging and under-tagging? It seems that an economic and robust way to pass a sentence to the parser is to only POS-tag those words or sets of words that have many entries in the MT lexicon. This gives the system more responsiveness and prune the wrong initial trees. The remaining 'glue words' in the sentence can be left to the parser to either decipher or ignore.

We argue that introducing further details into the AAC notation (such as POS features for the tense or form of a verb) should follow the same guidelines. Indeed, further ambiguities and more trees were built in memory because the MT system needed to consider extra features of a POS tag to disambiguate. Constraints in parsing comes at a cost.

The following strategies can be considered to improve the performance of the MT:

- reduce the number of trees by using a close domain vocabulary and POS tags;
- reduce the number of parses by introducing POS tags and POS features. If only the partial and initial trees

that match the source trees of the translation rules are allowed in memory, many undesirable cases will be eliminated; and

- reduce the number of translations and possible interpretations by prioritizing the translation rules and exploiting the context (such as the AAC sheet used and the AAC input log).

To conclude: in order to further boost the system performance, it is possible to store minimal and partial TAG trees within each virtual button. By pre-parsing all well-formed generated sentences, it would give more responsiveness to the system at runtime. This process can be achieved in the GNApp authoring mode where new vocabulary and sentence sheets can be created and customized.

Reducing the parse coverage and sentence generation to what the translation system can handle is a way to enforce accuracy in translation, to the detriment of coverage. This can be achieved by extending the prediction engine to implement a syntactic guide or wizard.

6.2 Other future issues

Can we adapt the system to allow a person with low literacy, either hearing or Deaf, to input sentences in English and produce a well-formed target language? It is unlikely that a forgiving system can be built on the current system. The reason is that robust exhaustive parsers are rare. Given the current system, it would be better to provide a grammatical guide to support the input process. In this scenario a pro-active system would seek to predict and suggest correct grammar constructs according to the inputs of the user. The system can further limit the composition and generation of sentences to that of its own grammatical coverage and translation capabilities.

In scenario number two, the MT system can be made to produce other language structures. For example, the source language can be transformed in such a way as to achieve various levels of literacy: contractions, expansions, completion and other kind of simplifications can match such targets.

A third and future scenario introduces a signing avatar [14], a computer animated humanoid that can communicate in sign language. The signing avatar can be integrated into the GNApp framework as proposed in Figure 3. It then provides the AAC device with an additional output modality other than voice.

Once the grammar of SASL has been formalized, it is theoretically possible to transform well-formed SASL utterances into English sentences. In this scenario a Deaf person can use a grid interface such as GNApp to input his language and reduce code-switching. This can typically be done using (for example) SASL video clips, HamNoSys or SignWriting directly on the buttons of the AAC interface.

7 Conclusion

We showed that the combination of an AAC grid interface and a standard rule-based MT system can enhance the practicality of a sign language communication environment. We illustrated that the use of the AAC grid interface helps to alleviate some of the obstacles in the accuracy of MT systems, namely, POS tagging and word sense disambiguation. This process is invisible to the user of the system.

Acknowledgements

The financial assistance of the National Research Foundation towards this research (under grant number 2053436) is hereby acknowledged. Opinions expressed in this paper and conclusions arrived at are those of the authors and not necessarily to be attributed to the National Research Foundation.

References

- [1] J.A. Bangham, S.J. Cox, R. Elliott, J.R.W. Glauert and I. Marshall, Virtual Signing: Capture, Animation, Storage and Transmission – an Overview of the VisiCAST Project. *IEE Seminar on Speech and Language Processing for Elderly and Disabled People*, 2000. <http://www.visicast.co.uk>.
- [2] J. Bungeroth and H. Ney, Statistical Sign Language Translation. *Proceedings of the Workshop on Representation and Processing of Sign Languages, 4th International Conference on Language Resources and Evaluation (LREC2004)*, May 2004, Lisbon, Portugal, pp. 105–108.
- [3] Hamburg Notation System, <http://www.sign-lang.uni-hamburg.de/Projects/hamNoSys.html>.
- [4] V. Sutton, Sign Writing Website. <http://www.SignWriting.org>.
- [5] L. Coetzee, N. Govender and I. Viviers, The National Accessibility Portal: an Accessible Information Sharing Portal for the South African Disability Sector. *Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A'07)*, Banff, Canada, 2007, pp. 44–53.
- [6] T. Wandmacher, J.-Y. Antoine and F. Poirier, SIBYLLE: A System for Alternative Communication Adapting to the Context and its User. *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS'07)*, October 2007, Tempe, Arizona, USA, pp 203–210.
- [7] L. van Zijl, The South African Sign Language Machine Translation Project, *Proceedings of the Eighth International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2006)*, Portland, USA, October 2006.
- [8] A.K. Joshi and Y. Schabes, Handbook of Formal Languages, volume 3, pp. 69–120, 1997.
- [9] J. Welgemoed, A Prototype Machine Translation System, *MSc thesis*, University of Stellenbosch, South Africa, December 2006.
- [10] R. Mitkov, The Oxford Handbook of Computational Linguistics. Oxford University Press, 2005.
- [11] A. Cook and S. Hussey, Assistive Technologies: Principles and Practice. Mosby, 1995.
- [12] P.L. Hudson, *Recommitment to the Fitzgerald Key*, American Annals of the Deaf, Volume 124 No 3, pp. 397–399, June 1979.
- [13] B.R. Clarke and D.A. Stewart, *Reflections on Language Programs for the Hearing Impaired.*, Journal of Special Education, Vol 20 No 2, pp 153–165, 1986.
- [14] J. Fourie, The Design of a Generic Signing Avatar Animation System. *MScEng thesis*, Stellenbosch University, South Africa, December 2006.