Arabic Text to Arabic Sign Language Translation System for the Deaf and Hearing-Impaired Community

Abdulaziz Almohimeed
University of Southampton
United Kingdom
aia07r@ecs.soton.ac.uk

Mike Wald
University of Southampton
United Kingdom
mw@ecs.soton.ac.uk

R. I. Damper
University of Southampton
United Kingdom
rid@ecs.soton.ac.uk

Abstract

This paper describes a machine translation system that offers many deaf and hearing-impaired people the chance to access published information in Arabic by translating text into their first language, Arabic Sign Language (ArSL). The system was created under the close guidance of a team that included three deaf native signers and one ArSL interpreter. We discuss problems inherent in the design and development of such translation systems and review previous ArSL machine translation systems, which all too often demonstrate a lack of collaboration between engineers and the deaf community. We describe and explain in detail both the adapted translation approach chosen for the proposed system and the ArSL corpus that we collected for this purpose. The corpus has 203 signed sentences (with 710 distinct signs) with content restricted to the domain of instructional language as typically used in deaf education. Evaluation shows that the system produces translated sign sentences outputs with an average word error rate of 46.7% and an average position error rate of 29.4% using leave-one-out cross validation. The most frequent source of errors is missing signs in the corpus; this could be addressed in future by collecting more corpus material.

1 Introduction

Machine translation (MT) has developed rapidly since 1947, when Warren Weaver first suggested the use of computers to translate natural languages (Augarten, 1984). Presently, this technology offers a potential chance for ArSL signers to benefit by, for instance, giving them access to texts published in Arabic. ArSL and general sign language (SL) have inherent ambiguity problems that should be taken into account while designing any ArSL translation system. Therefore, ArSL translation must be done through close collaboration with the deaf community and signing experts. This paper describes a full prototype MT system that translates Arabic texts into deaf and hearing-impaired peoples' first language, Arabic Sign Language (ArSL). It is the result of extended collaboration between engineers and a team consisting of three deaf native signers and one ArSL interpreter.

Most existing systems have wrongly assumed that ArSL is dependent on the Arabic language (Mohandes, 2006; Alnafjan, 2008; Halawani, 2008; Al-Khalifa, 2010). These systems make word-to-sign translations without regard to ArSL's unique linguistic characteristics, such as its own grammar, structure, and idioms, as well as regional variations (Abdel-Fateh, 2004) or translate into finger-spelling signs that only exist in Arabic, not in ArSL.

This paper begins by providing a brief background of ArSL. It then addresses the problems and misconceptions plaguing previous ArSL systems. Thereafter, it describes related works built on the assumption of one of the two misconceptions mentioned above. The rest of the paper will present an example-based machine translation (EBMT) system that translates published Arabic texts to make them accessible to deaf and hearing-impaired people who use ArSL.

Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies, pages 101–109,
Edinburgh, Scotland, UK, July 30, 2011.

2 Background

SL is composed of basic elements of gesture and location previously called 'cheremes' but modern usage has changed to the even more problematic 'optical phoneme' (Ojala, 2011). These involve three components: hand shape (also called hand configuration), position of the hand in relation to the signer's body, and the movement of direction of the hand. These three components are called manual features (MFs). In addition, SL may involve non-manual features (NMFs) that involve other parts of the body, including facial expression, shoulder movements, and head tilts in concurrence with MFs. Unlike written language, where a text expresses ideas in a linear sequence, SL employs the space around the signer for communication, and the signer may use a combination of MFs and NMFs. These are called multi-channel signs. The relationship between multi-channel signs may be parallel, or they may overlap during SL performance. MFs are basic components of any sign, whereas NMFs play an important role in composing signs in conjunction with MFs. NMFs can be classified into three types in terms of their roles. The first is essential: If an NMF is absent, the sign will have a completely different meaning.

An example of an essential NMF in ArSL is the sign sentence: "Theft is forbidden", where as shown in Figure 1(a), closed eyes in the sign for "theft" are essential. If the signer does not close his or her eyes, the "theft" sign will mean "lemon". The second type of NMF is a qualifier or emotion. In spoken language, inflections, or changes in pitch, can express emotions, such as happiness and sadness; likewise, in SL, NMFs are used to express emotion as in Figure 1(b). The third type of NMF actually plays no role in the sign. In some cases, NMFs remain from a previous sign and are meaningless. Native signers naturally discard any meaningless NMFs based on their knowledge of SL.

(a) Essential NMF

(b) Emotion NMF

Figure 1: (a) The sign for "theft", in which the signer uses the right hand while closing his eyes. (b) His facial expressions show the emotion of the sign.

3 Problem Definition

ArSL translation is a particularly difficult MT problem for four main reasons, which we now describe.

The first of the four reasons is the lack of linguistic studies on ArSL, especially in regard to grammar and structure, which leads to a major misunderstanding of natural language and misleads researchers into failing to build usable ArSL translation systems. These misunderstandings about ArSL can be summed up by the following:

- SL is assumed to be a universal language that allows the deaf anywhere in the world to communicate, but in reality, many different SLs exist (e.g., British SL, Irish SL, and ArSL).
- ArSL is assumed to be dependent on the Arabic language but it is an independent language that has its own grammar, structure, and idioms, just like any other natural language.
- ArSL is not finger spelling of the Arabic alphabet, although finger spelling is used for names and places that do not exist in ArSL or for other entities for which no sign exists (e.g., neologisms).

The related work section will describe an ArSL translation system that was built based on one of these misunderstandings.

The second factor that should be taken into account while building an ArSL translation system is the size of the translation corpus, since few linguistic studies of ArSL's grammar and structure have been conducted. The data-driven approach adopted here relies on the corpus, and the translation accuracy is correlated with its size. Also, ArSL does not have a written system, so there are no existing ArSL documents that could be used to build a translation corpus, which must be essentially visual (albeit with annotation). Hence, the ArSL corpus must be built from scratch, limiting its size and ability to deliver an accurate translation of signed sentences.

The third problem is representing output sign sentences. Unlike spoken languages, which use sounds to produce utterances, SL employs 3D space to present signs. The signs are continuous, so some means are required to produce novel but fluent signs. One can either use an avatar or, as here, concatenate video clips at the expense of fluency.

The last problem is finding a way to evaluate SL output. Although this can be a problem for an MT system, it is a particular challenge here as SL uses multi-channel representations (Almohimeed et al., 2009).

4 Related Works

As mentioned above, we deem it necessary for engineers to collaborate with the deaf community and/or expert signers to understand some fundamental issues in SL translation. The English to Irish Sign Language (ISL) translation system developed by Morrissey (2008) is an example of an EBMT system created through strong collaboration between the local deaf community and engineers. Her system is based on previous work by Veale and Way (1997), and Way and Gough (2003; 2005) in which they use tags for sub-sentence segmentation. These tags represent the syntactic structure. Their work was designed for large tagged corpora.

However, as previously stated, existing research in the field of ArSL translation shows a poor or weak relationship between the Arab deaf community and engineers. For example, the system built by Mohandes (2006) wrongly assumes that ArSL depends on the Arabic language and shares the same structure and grammar. Rather than using a data-driven or rule-based approach, it uses so-called "direct translation" in which words are transliterated into ArSL on a one-to-one basis.

5 Translation System

The lack of linguistic studies on ArSL, especially on its grammar and structure, is an additional reason to favour the example-based (EMBT) approach over a rule-based methodology. Further, the statistical approach is unlikely to work well given the inevitable size limitation of the ArSL corpus, imposed by difficulties of collecting large volumes of video signing data from scratch. On the other hand, EBMT relies only on example-guided suggestions and can still produce reasonable translation output even with existing small-size corpora. We have adopted a chunk-based EBMT system, which produces output sign sentences by comparing the Arabic text input to matching text fragments, or 'chunks'. As Figure 2 shows, the system has two phases. Phase 1 is run only once; it pre-compiles the chunks and their associated signs. Phase 2 is the actual translation system that converts Arabic input into ArSL output. The following sections will describe each component in Figure 2.

5.1 Google Tashkeel Component

In Arabic, short vowels usually have diacritical marks added to distinguish between similar words in terms of meaning and pronunciation. For example, the word كُتب means *books*, whereas كتب means *write*. Most Arabic documents are written without the use of diacritics. The reason for this is that Arabic speakers can naturally infer these diacritics from context. The morphological analyser used in this system can accept Arabic input without diacritics, but it might produce many different analysed outputs by making different assumptions about the missing diacritics. In the end, the system needs to select one of these analysed outputs, but it might not be equivalent to the input meaning. To solve this problem, we use Google Tashkeel (`http://tashkeel.googlelabs.com/`) as a component in the translation system; this software tool adds missing diacritics to Arabic text, as shown in Figure 3. (In Arabic, *tashkeel* means "to add shape".) Using this component, we can guarantee

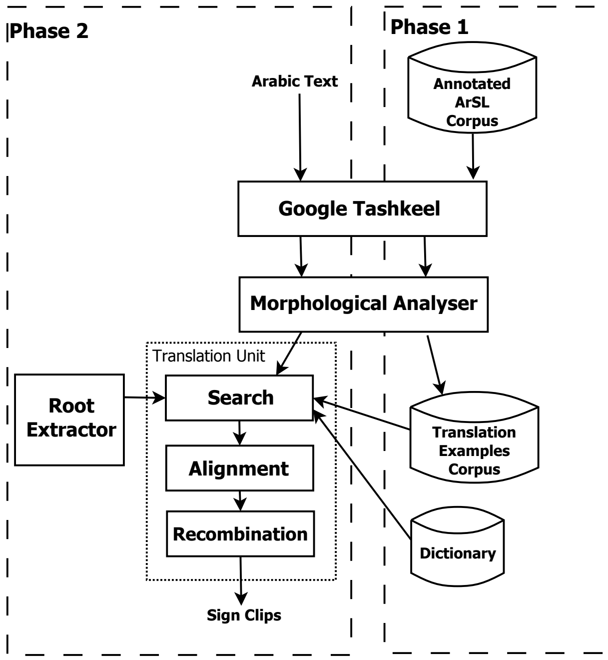


Figure 2: Main components of the ArSL chunks-based EBMT system. Phase 1 is the pre-compilation phase, and Phase 2 is the translation phase.

that the morphological analyser described immediately below will produce only one analysed output.

5.2 Morphological Analyser

The Arabic language is based on root-pattern schemes. Using one root, several patterns, and numerous affixes, the language can generate tens or hundreds of words (Al Sughaiyer and Al Kharashi, 2004). A *root* is defined as a single morpheme that provides the basic meaning of a word. In Arabic, the root is also the original form of the word, prior to any transformation process (George, 1990). In English, the root is the part of the word that remains after the removal of affixes. The root is also sometimes called the stem (Al Khuli, 1982). A *morpheme* is defined as the smallest meaningful unit of a language. A *stem* is a single morpheme or set of concatenated morphemes that is ready to accept affixes (Al Khuli, 1982). An *affix* is a morpheme that can be added before (a prefix) or after (a suffix) a root or stem. In English, removing a prefix is usually harmful because it can reverse a word's meaning (e.g., the word *disadvantage*). However, in Arabic, this action does not reverse the meaning of the word (Al Sughaiyer and Al Kharashi, 2004). One of the major differences between Arabic (and the Semitic language family in general) and English (and similar languages) is that Arabic is 'derivational' (Al Sughaiyer and Al Kharashi, 2004), or non-catenative, whereas English is concatenative.

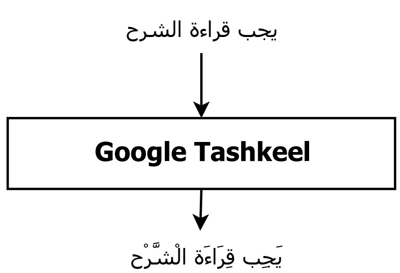


Figure 3: An example of an input and output text using Google Tashkeel. The input is a sentence without diacritics; the output shows the same sentence after adding diacritics. English translation: *You should read the explanation.*

Figure 4 illustrates the Arabic derivational system. The three words in the top layer (كتب, خبز, ذهب) are roots that provide the basic meaning of a word. Roman letters such as *ktb* are used to demonstrate the pronunciation of Arabic words. After that, in the second layer, "xAxx" (where the small letter x is a variable and the capital letter A is a constant) is added to the roots, generating new words (كاتب, خابز, ذاهب) called stems. Then, the affix "ALxxxx" is added to stems to generate words (الكاتب, الخابز, الذاهب).

Morphology is defined as the grammatical study of the internal structure of a language, which includes the roots, stems, affixes, and patterns. A morphological analyser is an important tool for predicting the syntactic and semantic categories of unknown words that are not in the dictionary. The primary functions of the morphological analyser are the segmentation of a word into a sequence of morphemes and the identification of the morpho-syntactic relations between the morphemes (Semmar et al., 2005).

Due to the limitation of the ArSL corpus size, the syntactic and semantic information of unmatched chunks needs to be used to improve the translation system selection, thereby increasing the system's

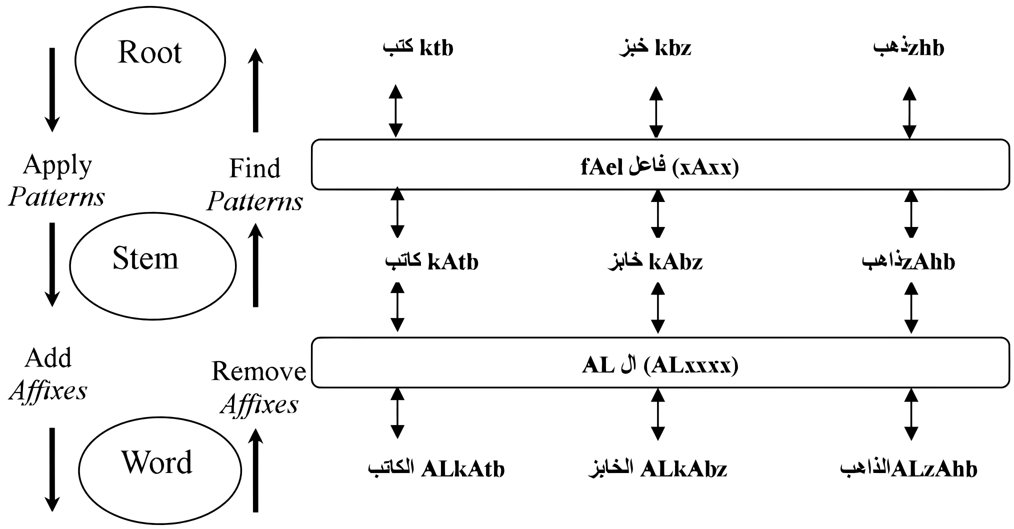


Figure 4: An example of the Arabic derivational system. The first stage shows some examples of roots. An Arabic root generally contains between 2 and 4 letters. The second stage shows the generated stems from roots after adding the pattern to the roots. The last stage shows the generated words after the prefixes are added to the stems.

accuracy. To analyse this information, Buckwalter's morphological analyser was used (Buckwalter, 2004). In addition, we implemented a root extractor based on a tri-literal root extraction algorithm (Momani and Faraj, 2007). In this work, sentences without diacritics are passed to the morphological analyser, which therefore produces multiple analyses (distinguished by different assumptions about the missing diacritics) from which the 'best' one must be chosen. This is not an easy decision for a computer system to make. The approach we have implemented uses the Google Tashkeel output in conjunction with the Levenshtein distance (Levenshtein, 1966) to select among the multiple analyses delivered by Buckwalter's morphological analyser. Figure 5 gives an example showing how the morphological and root extractor analyses the syntactic, semantic and root information.

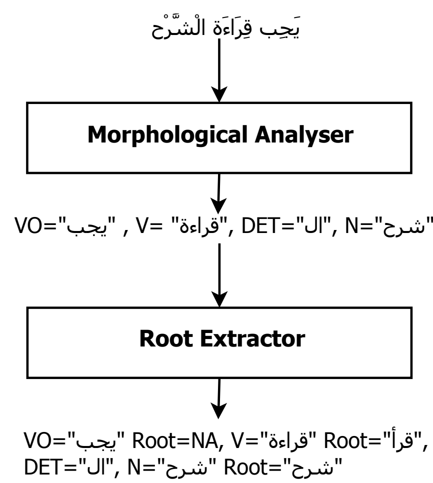


Figure 5: An example showing how the morphological analyser and root extractor are utilised for the same sentence as in Fig. 3.

5.3 Corpus

An annotated ArSL corpus is essential for this system, as for all data-driven systems. Therefore, we collected and annotated a new ArSL corpus with the help of three native ArSL signers and one expert interpreter. Full details are given in Almohimeed et al. (2010). This corpus's domain is restricted to the kind of instructional language used in schools for deaf students. It contains 203 sentences with 710 distinct signs. The recorded signed sentences were annotated using the ELAN annotation tool (Brugman and Russel, 2004), as shown in Figure 6. Signed sentences were then saved in EUDICO Annotation Format (EAF).

The chunks database and sign dictionary are de-

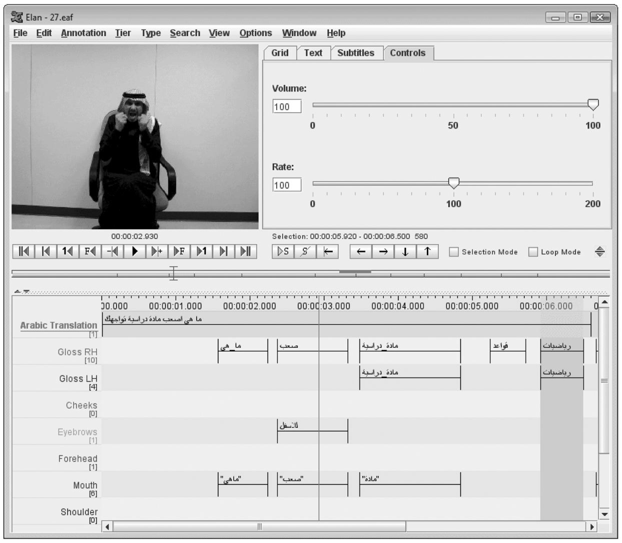
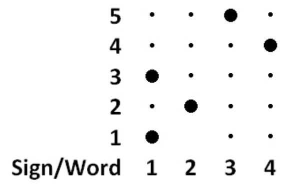


Figure 6: An example of a sign sentence annotated by the ELAN tool.



Starting from Word #	T1	T2	T3	T4
1	[1-1,3]	[1,2-1,2,3]	[1,2,3-1,2,3,5]	[1,2,3,4-1,2,3,4,5]
2	[2-2]	[2,3-2,5]	[2,3,4-2,4,5]	
3	[3-5]	[2,3-4,5]		
4	[4,4]			

Figure 7: An example of how the system finds chunks by finding continuous words and signs.

rived from this corpus by parsing the EAF file to extract the MFs and NMFs to build a parallel corpus of ArSL and associated Arabic chunks. Before detecting and extracting chunks, words are linked with their equivalent signs in each sentence. After a manual words-to-signs alignment, chunk extraction begins. This is done automatically by finding consistent word/sign sequence pairs. The refined technique proposed by Och and Ney (2003) is employed in this system to extract chunks. Figure 7 illustrates how the system does so.

The chunks table has four fields. The first contains all the Arabic words in the chunk, and the second contains an identifier for the video clips of the signs. The third field contains syntactic and semantic information about the Arabic words. The last field indicates the relative position of the parallel ArSL and text chunks. After extraction of the chunks, the database is sorted from largest chunks (in terms of words) to smallest. Details of the tool that carries out these steps will be published in a future paper.

5.4 Translation Unit

As depicted earlier in Figure 2, the translation unit contains three components. The first is the search component, which is responsible for finding chunks that match the input. It starts matching words from the beginning of the chunks table and scans the table until the end. Overlapping chunks have higher priority for selection than separate chunks. Then, for any remaining unmatched input words, it starts matching stems from the beginning through to the end of the chunks table. The second is the alignment component, which replaces chunks with their equivalent signs. For the remaining input words that do not have a chunk match, a sign dictionary is used to translate them. If the word does not appear in the dictionary (which is possible due to the size of the corpus), the system starts searching for the stem of the word and compares it with the stems in the dictionary. If the stem also does not appear in the database or dictionary, the system searches for a matching root. This process will increase the chance of translating the whole input sentence. The last component is recombination, which is responsible for delivering sign output using the sign location on both the chunks table and dictionary. The component will produce a series of sign clips, and between two clips, it will insert a transition clip, as shown in Figure 8.

The output representation has been tested by the team of three native signers on several hundred

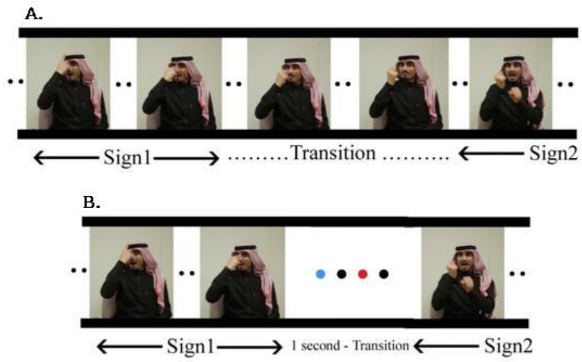


Figure 8: Image A shows an example of the original representation, while B shows the output representation.

selected sign sentences in which natural transitions were replaced by a one-second pause. Moreover, the sign in actual sentences has been replaced by the equivalent sign in the sign dictionary. This test showed that the meaning of the sentences was clearly expressed to the signers; all three evaluated the test sentences by giving them 5 points out of 5, which means the sentence clearly expresses its meaning. In addition, the fluency of sentences was deemed acceptable since the evaluators choose 4 points out of 5. In view of this positive result, we did not feel it worthwhile to evaluate the effect of variation in (one-second) pause duration, although this will be adjustable by the user in the final implementation.

6 Illustration

In this section, we illustrate the workings of the prototype system on three example sentences.

Figures 9, 10, and 11 shows the main stages of the translation of Arabic sentence to ArSL for some selected inputs. The input sentence in Figure 9 is 2 words, 5 in Figure 10, and 7 in Figure 11. As shown in the figures, the system starts collecting the morphological details of the Arabic input. Then, it passes it to the translation unit where it first searches for a matching chunk in the chunks table. When many matches are received, the system takes the largest chunk (recall that the system gives overlapping chunks higher priority than isolated chunks and that when no chunks are found in the table, the system uses the stem rather than the word to find a match). When a match is not found, the system uses the dictionary to translate the sign by looking for the word. In the next stage, alignment, the system identifies the corresponding translation chunk from both the chunks table and dictionary. The system uses the location field in the chunks table and dictionary to determine the location of the translated chunk. The last stage is recombination, during which the system delivers a sign sentence in a Windows Media Video (WMV) format, as shown in Figure 8.

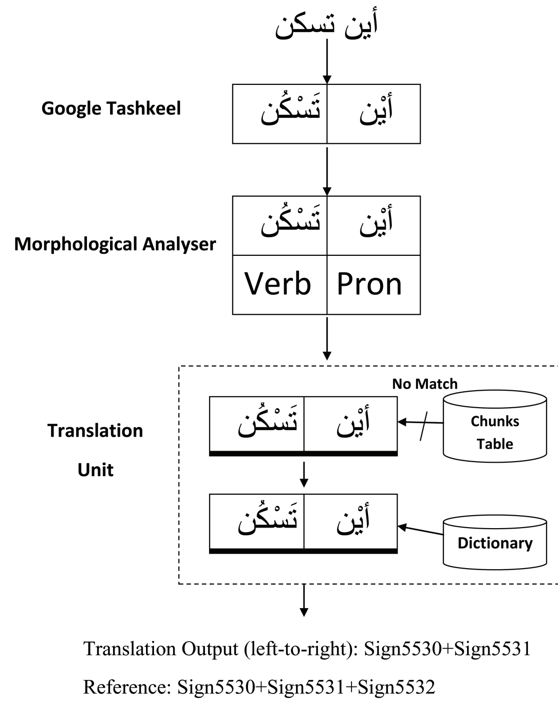


Figure 9: Example translation from the first Arabic sentence to ArSL. The square selection represents a chunk match. The crossed arrow means that there was no chunk match and that it has been translated using the dictionary. In this case, the output is incorrect (Sign5532 is missing). English translation: *Where do you live?*

7 Leave-One-Out Cross Validation

The full evaluation results (203 sentences) were acquired using leave-one-out cross validation. This technique removes a test sentence from the dataset and then uses the remaining dataset as the translation corpus. The word error rate (WER) was, on average, 46.7%, whereas the position-independent word error rate (PER) averaged 29.4%. The major source of

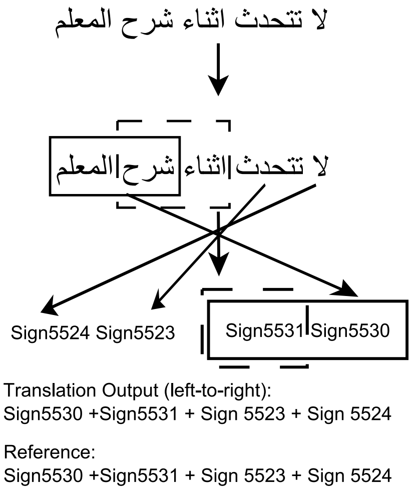


Figure 10: Example translation from the second Arabic sentence to ArSL. In this case, the output is correct. English translation: *Don't talk when the teacher is teaching*.

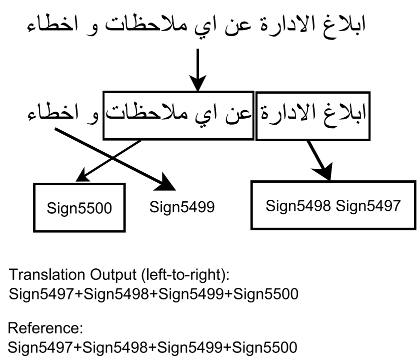


Figure 11: Example translation from the third Arabic sentence to ArSL. Again, the output is correct. English translation: *Let the Principal know about any suggestions or comments that you have*.

error is that signs in some translated sentences do not have equivalent signs in the dictionary. In principle, this source of error could be reduced by collection of a larger corpus with better coverage of the domain, although this is an expensive process.

8 Conclusion

This paper has described a full working prototype ArSL translation system, designed to give the Arabic deaf community the potential to access published Arabic texts by translating them into their first language, ArSL. The chunk-based EBMT approach was chosen for this system for numerous reasons. First, the accuracy of this approach is easily extended by adding extra sign examples to the corpus. In addition, there is no requirement for linguistic rules; it purely relies on example-guided suggestions. Moreover, unlike other data-driven approaches, EBMT can translate using even a limited corpus, although performance is expected to improve with a larger corpus. Its accuracy depends primarily on the quality of the examples and their degree of similarity to the input text. To overcome the limitations of the relatively small corpus, a morphological analyser and root extractor were added to the system to deliver syntactic and semantic information that will increase the accuracy of the system. The chunks are extracted from a corpus that contains samples of the daily instructional language currently used in Arabic deaf schools. Finally, the system has been tested using leave-one-out cross validation together with WER and PER metrics. It is not possible to compare the performance of our system with any other competing Arabic text to ArSL machine translation system, since no other such systems exist at present.

Acknowledgments

This work would not have been done without the hard work of the signers' team: Mr. Ahmed Alzaharani, Mr. Kalwfah Alshehri, Mr. Abdulhadi Alharbi and Mr. Ali Alholafi.

References

Mahmoud Abdel-Fateh. 2004. Arabic Sign Language: A perspective. *Journal of Deaf Studeis and Deaf Education*, 10(2):212–221.

Hend S. Al-Khalifa. 2010. Introducing Arabic sign language for mobile phones. In *ICCHP'10 Proceedings of the 12th International Conference on Computers Helping People with Special Needs*, pages 213–220 in Springer Lecture Notes in Computer Science, Part II, vol. 6180, Linz, Austria.

Muhammad Al Khuli. 1982. *A Dictionary of Theoretical Linguistics: English-Arabic with an Arabic-English Glossary*. Library of Lebanon, Beirut, Lebanon.

Imad Al Sughaiyer and Ibrahim Al Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.

Abdulaziz Almohimeed, Mike Wald, and R. I. Damper. 2009. A new evaluation approach for sign language machine translation. In *Assistive Technology from Adapted Equipment to Inclusive Environments, AAATE 2009, Volume 25*, pages 498–502, Florence, Italy.

Abdulaziz Almohimeed, Mike Wald, and Robert Damper. 2010. An Arabic Sign Language corpus for instructional language in school. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC*, pages 81–91, Valetta, Malta.

Abeer Alnafjan. 2008. Tawasoul. Master's thesis, Department of Computer Science, King Saud University, Riyadh, Saudi Arabia.

Stan Augarten. 1984. *Bit by Bit: An Illustrated History of Computers*. Tickner and Fields, New York, NY.

Hennie Brugman and Albert Russel. 2004. Annotating multimedia/multi-modal resources with ELAN. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC*, pages 2065–2068, Lisbon, Portugal.

Tim Buckwalter. 2004. Issues in arabic orthography and morphology analysis. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, CAASL*, pages 31–34, Geneva, Switzerland.

Metri George. 1990. *Al Khaleel: A Dictionary of Arabic Syntax Terms*. Library of Lebanon, Beirut, Lebanon.

Sami M. Halawani. 2008. Arabic Sign Language translation system on mobile devices. *IJCSNS International Journal of Computer Science and Network Security*, 8(1):251–256.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Mohamed Mohandes. 2006. Automatic translation of Arabic text to Arabic Sign Language. *ICGST International Journal on Artificial Intelligence and Machine Learning*, 6(4):15–19.

Mohanned Momani and Jamil Faraj. 2007. A novel algorithm to extract tri-literal arabic roots. In *Proceedings ACS/IEEE International Conference on Computer Systems and Applications*, pages 309–315, Amman, Jordan.

Sara Morrissey. 2008. *Data-Driven Machine Translation for Sign Languages*. Ph.D. thesis, Dublin City University, Dublin, Ireland.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Sinja Ojala. 2011. Studies on individuality in speech and sign. Technical Report No. 135, TUCS Dissertations, Turku Centre for Computer Science, University of Turku, Finland.

Nasredine Semmar, Faïza Elkateb-Gara, and Christian Fluhr. 2005. Using a stemmer in a natural language processing system to treat Arabic for cross-language information retrieval. In *Proceedings of the Fifth Conference on Language Engineering*, pages 1–10, Cairo, Egypt.

Tony Veale and Andy Way. 1997. Gaijin: A bootstrapping approach to example-based machine translation. In *Proceedings of the Second Conference on Recent Advances in Natural Language Processing, RANLP*, pages 27–34, Tzigov Chark, Bulgaria.

Andy Way and Nano Gough. 2003. wEBMT: developing and validating an example-based machine translation system using the world wide web. *Computational Linguistics*, 29(3):421–457.

Andy Way and Nano Gough. 2005. Comparing example-based and statistical machine translation. *Natural Language Engineering*, 11(3):295–309.