Rule-based Conversion of Closely-related Languages: A Dutch-to-Afrikaans Convertor

Gerhard B van Huyssteen

Human Language Technologies Research Group
Meraka Institute, CSIR, Pretoria
gvhuyssteen@csir.co.za

Suléne Pilon

Centre for Text Technology (CTexT)
North-West University, Potchefstroom
sulene.pilon@nwu.ac.za

Abstract

For fast-tracking the development of resources for resource-scarce languages, one could transfer existing technologies from one language to another well-sourced, closely-related language. In this contribution, we describe the development and performance of a rule-based Dutch-to-Afrikaans convertor, with the aim to transform Dutch text so that it looks more like an Afrikaans text (even though it might not even be a good Dutch translation). The rules we used is based on systematic orthographic, morphosyntactic and lexical differences between the two languages. We report on an accuracy of 71% on word-level, after minor optimisation with regard to iteration of rules. In a small-scale evaluation on running text, we obtain a BLEU score of 0.2519. We conclude that such a rule-based approach to conversion of closely-related languages holds much promise, with potential application in technology transfer (or even machine translation) between such languages.

1. Introduction

One method to fast-track the development of resources for resource-scarce languages is to port/transfer/re-engineer existing technologies from one language *L1* to another closely-related language *L2*. The basic hypothesis is that "[if] the languages *L1* and *L2* are similar enough, then it should be easier [and quicker] to recycle software applicable to *L1* than to rewrite it from scratch for *L2*", thereby taking care of "most of the drudgery before any human has to become involved" [1]. [2] argues that resource-scarce languages could benefit from such an approach, especially where *L1* is a global, well-resourced language.

To illustrate the above hypothesis in real terms, let us assume that Dutch (*L1*) and Afrikaans (*L2*) are similar enough for purposes of technology transfer. The hypothesis then is that it would be easier and quicker to use and adapt, for example, an existing Dutch chunker to annotate Afrikaans sentences, than to develop an Afrikaans chunker from scratch. One could therefore use the Dutch chunker to annotate an Afrikaans corpus, then correcting systematic errors manually or semi-automatically, before training an Afrikaans chunker using the annotated data.

One could also add an extra layer of processing to this basic algorithm: instead of annotating original Afrikaans text directly with technologies for Dutch, one could firstly "transform"/convert the Afrikaans text through rule-based translation so that it looks more like Dutch (even though it might not even be a good Dutch translation). The idea is that one could improve the success of technology transfer between Dutch (*L1*) and Afrikaans (*L2*) if the text to be annotated by the Dutch technologies appears to be more Dutch-like. After the text has been annotated, one could then reverse the process to render an Afrikaans text with annotations, which could then be used for training Afrikaans technologies.

Our current research focuses on the development of an automatic, rule-based Dutch-to-Afrikaans convertor (D2AC). Our aim here is to develop a proof-of-concept for rule-based machine translation (MT) between Afrikaans and Dutch. Therefore, the rationale for the directionality (i.e. Dutch-to-Afrikaans instead of Afrikaans-to-Dutch, as one would expect from the hypothesis above) is based on the assumption that it would be easier to convert Dutch to Afrikaans, because Dutch is generally considered to be morphologically more complex than Afrikaans. While it gives us the opportunity to focus on matters related to the basic concept (such as system architecture), rather than to get stuck on issues of linguistic complexity, we also get the opportunity to discover linguistic issues that will be of importance for an Afrikaans-to-Dutch convertor (A2DC).

For purposes of the research we report on here, our concern is the development and evaluation of a unidirectional D2AC. Our focus here is not yet on technology transfer, but rather on the quality of the conversion algorithm as a proof-of-concept. In as such, our efforts are aimed at processing on the grapheme and word-level and not on the syntactic level (i.e. a convertor that is not concerned with issues related to word ordering, anaphoric relations, idioms, negation, etc.).

In Section 2 we discuss related work, specifically regarding rule-based MT between closely-related languages. Section 3 presents a concise overview of some of the most prototypical linguistic differences between Afrikaans and Dutch. In section 4 we present an overview of D2AC's architecture, as well as details of some of its components/modules. Thereafter in Section 5 we evaluate the performance of D2AC on word-level as well as on running text. Section 6 concludes with a view on future work.

2. Related work

Several approaches have been used to develop MT systems, viz. rule-based, corpus-based and hybrid approaches. When using a rule-based approach, the source language (SL) is parsed yielding some form of intermediary representation, which is then used to generate the target language (TL). Rule-based approaches usually necessitate extensive lexicons with morphological, syntactic and semantic information and large sets of rules, which usually have to be handcrafted by language experts of the SL and the TL.

These resources are expensive and laborious to develop, since every difference between the two languages needs to be handled by means of hand-written rules. Therefore, developers of MT systems usually favour a corpus-based or hybrid approach [3], especially when developing an MT system for two very different languages such as English and Arabic. However, when two languages display a high level of similarity on various grammatical levels, it should be less arduous to write transfer rules to do a translation of the one language into the other, similar language.

Languages that display such similarities are generally referred to as closely-related languages, but it is difficult to define what exactly constitutes "closely-relatedness" and to ascertain when it is possible to say of two languages that they are closely-related. Hence, it is necessary to determine if two languages display a sufficient level of similarity to use a rule-based approach in the development of an MT system for them. In an attempt to give a more concise definition, [4] distinguish between language variants (considered to be one language, e.g. Serbian and Croatian; also Dutch and Flemish), very close languages (similarity in morphology, syntax and lexis, e.g. Czech and Slovak; also Dutch and Afrikaans), closely-related languages (similarity in morphology and lexis, e.g. Czech and Polish; also Dutch and German), and related languages (shared origin and influences, without necessarily sharing linguistic similarities, e.g. Czech and Latvian; also Dutch and Swedish). They then demonstrate that the assumption that it is easier to develop MT systems for closely-related languages holds only for very close languages [4].

Several systems that translate between closely-related languages have been developed. Many of these systems make use of transfer-rules to handle some aspects of translation (see for example [2], [5], [6], [7], [8], and [9]). Different strategies are employed to minimise or simplify the transfer rules, thereby expediting the development of the MT system.

For purposes of this research, we assume that the similarities between Afrikaans and Dutch are adequate for the languages to be categorised somewhere between closely-related and very close languages. (We use "closely-related" to define the relationship in this article, in order to avoid the more lengthy, though probably more correct "very close/closely-related"). Subsequently we assume that a simple, rule-based strategy can be used in the development of a Dutch-Afrikaans MT system. The relationship between Afrikaans and Dutch will need to be investigated more thoroughly by means of comparative studies with other language pairs in order to do a more accurate categorisation of the closeness of the language pair.

3. Afrikaans and Dutch as closely-related languages

The language choice for this project is based on the fact that Afrikaans is a resource-scarce language, while its closely-related parent language, Dutch, is a global, well-sourced language. These two languages therefore make an ideal choice to test the hypothesis explained in Section 1. The purpose of this section is to provide a concise overview of some of the most prototypical differences between Afrikaans and Dutch on orthographic, morphosyntactic and lexical levels; for comprehensive comparisons see [10], [11], [12] and [13].

3.1. Orthographic level

Due to various natural phonological processes, and since the spelling systems of both Dutch and Afrikaans are based on pronunciation, the orthographic realisations of these two languages differ rather systematically. For example, due to devoicing the Dutch [z] in the onset position is realised as [s] in Afrikaans (e.g. Du. *zomer* and Afr. *somer* 'summer'). Other examples of consonant changes include [sx] to [sk] (e.g. Du. *school* and Afr. *skool* 'school'), [tsi] to [si] (e.g. Du. *vakantie* and Afr. *vakansie* 'holiday'), [f] to [v] (e.g. Du. *halve* to Afr. *halwe* 'half'), etc. Many differences also occur due to elision, such as procope (e.g. the historical Du. *dispens* vs. contemporary Afrikaans *spens* 'pantry'), syncope (e.g. loss of the intervocalic [x] in Du. *hagel* vs. Afr. *hael* 'hail'; the intervocalic [d] in Du. *zadel* vs. Afr. *saal* 'saddle'; the intervocalic [v] in Du. *oven* vs. Afr. *oond* 'oven'), and apocope (e.g. loss of the final [t] after [t; x; k; p; s] in Du. *markt* vs. Afr. *mark* 'market'). Similarly, some differences could be ascribed to processes of addition, such as prothesis (e.g. Du. *des avonds > 's avonds* vs. Afr. *saans* 'in the evening'), epenthesis (e.g. Du. *eigenlijk* vs. Afr. *eintlik* 'actually'), and epithesis (e.g. Du. *oven* vs. Afr. *oond* 'oven') (see [14]). Due to divergence in spelling conventions a few systematic differences also occur, for example ***c*** vs. ***k*** (e.g. Du. *controle* and Afr. *kontrole* 'control'), ***ch*** vs. ***g*** (e.g. Du. *echter* and Afr. *egter* 'however'), or ***c*** vs. ***s*** (e.g. Du. *producent* and Afr. *produsent* 'producer').

Similar systematic differences could be observed with regard to vowels. Consider for example changes due to vowel reduction in cases like [ʌu] vs. [əu] (e.g. Du. *blauw* and Afr. *blou* 'blue'), or [ɛ] vs. [ə] (e.g. Du. *stengel* and Afr. *stingel* 'stem'). Other examples of vowel changes include [ɔ] vs. [œ] (e.g. Du. *vork* and Afr. *vurk* 'fork'), [ʏ] vs. [ɔ] (e.g. Du. *nummer* vs. Afr. *nommer* 'number'), [o:] vs. [ø] (e.g. Du. *door* vs. Afr *deur* 'through'), [ɛ] vs. [ɑ] (e.g. Du. *vers* and Afr. *vars* 'fresh'), [a:] vs. [æ/ɛ] (e.g. Du. *paard* and Afr. *perd* 'horse'), etc. Similar to consonants, some differences can be ascribed to divergence in spelling conventions, such as ***y*** vs. ***i*** (e.g. Du. *systeem* and Afr. *sisteem* 'system'), ***au*** vs. ***ou*** (e.g. Du. *auteur* and Afr. *outeur* 'author'), or most notably ***ij*** (often handwritten as ***ÿ***) vs. ***y*** (e.g. Du. *bijna* and Afr. *byna* 'almost').

3.2. Morphosyntactic level

With regard to grammar, Afrikaans and Dutch differ most on the morphological/morphosyntactic level. Some of the most prominent aspects relate to loss/simplification of the more complex inflection system in Dutch, such as simplification of verbal conjugation (imperfective and pluperfect tenses, the infinitive, verb congruence for number and person, the distinction between strong and weak verbs, etc.), the gender system (most noticeable in inflection on pre-nominal adjectives), the genitive and the pronominal system. For example, in Dutch the simple present tense conjugates for person, whereas no conjugation is seen in Afrikaans. Similar other systematic differences related to loss of inflection are discussed at length by, amongst others, [10] and [13].

Whereas large differences related to inflection could be observed, many similarities with regard to word-formation exist between Afrikaans and Dutch (with a few exceptions, such as reduplication as a productive word-formation process in Afrikaans). [15] convincingly argues that the influence of Dutch on Afrikaans could be observed very distinctly in the "copying" of the word-formation system in Afrikaans. A number of systematic differences, mainly due to phonological processes discussed above, could also be observed; compare for instance *-atie* vs. *-asie* (e.g. Du. *organisatie* and Afr. *organisasie* 'organisation'), *-air* vs. *-êr* (e.g. Du. *primair* and Afr. *primêr* 'primary'), or *-ist* vs. *-is* (e.g. Du. *propagandist* and Afr. *propagandis* 'propagandist').

3.3. Lexical level

With regard to the lexicon of Afrikaans, it can be assumed with some certainty that 90-95% of all lexical items are of Dutch origin [16]; see also [10]. Because of the spelling, phonological and morphological changes discussed above, many of these lexical items (and their associated word-forms) are not graphologically identical anymore (although some are), while semantic changes can be observed in another part of the lexicon. For purposes of this research, we distinguish between identical cognates (i.e. etymologically related words from two different languages), non-identical cognates, false friends, and non-cognates.

Identical cognates are word-forms that are graphologically identical in Dutch and Afrikaans, and could by and large be ascribed to linguistic inheritance. Compare for example words like *kelder* 'cellar', *olie* 'oil', and *straat* 'street'. In some cases inflected forms are identical (e.g. the plural *kelders* 'cellars'), and in other cases there might be shared inflected forms (e.g. the plural of *olie* in Dutch is either *oliën* or *olies*, while only the latter form is used in Afrikaans). However, in the majority of cases the inflected forms are not identical (e.g. Du. *straten* vs. Afr. *strate* 'streets'; compare also the diminutive forms of all the above-mentioned examples).

Non-identical cognates are lexical items that are etymologically related in Afrikaans and Dutch, but that differ graphologically in a systematic way. Inflected word-forms (i.e. paradigms) make up the majority under this category; compare for example Afr. *bestuur* 'drive' with five cognates (all inflected forms of the verb) in Du. *besturen*, *bestuurt*, *bestuurd*, *bestuurde*, *bestuurden*.

False friends are words with the same form, but with different meanings in the two languages due to semantic broadening, semantic narrowing or referent changes. Compare for example Du. *amper* 'almost not' (as in *Hij heeft de trein amper gehaald* 'He almost did not get onto the train') vs. Afr. amper 'almost' (as in *Hy het die trein amper gehaal* 'He almost got onto the train'). According to the Dutch speaker, the referent was lucky to catch his train, while according to the Afrikaans speaker, the referent was a few seconds too late. (On 2009/10/03, the Dutch-Afrikaans Google Translate (translate.google.com) incorrectly translated the Du. *amper* to Afr. *amper*, while correctly translating Afr. *amper* to Du. *bijna* in the above example.)

Non-cognates are defined, for purposes of this research, as graphologically unrelated word-forms referring to the same referent. Compare for example many words related to ethnobiological nomenclature, such as Du. *gnoe* vs. Afr. *blouwildebees* 'gnu', Du. *baobab* vs. Afr. *kremetart* 'baobab', or Du. *eucalypthus* vs. Afr. *bloekom* 'eucalyptus'.

4. A Dutch-to-Afrikaans convertor

In the design of D2AC, we specifically opt for a highly modular design in order to facilitate easy adaptations and changes during experimentation – specifically for linguists/language students, who might not have a good command of Perl. For example, all regular expressions are entered in separate text files in the format:

```
<SearchString><tab><SubstitutionString>
```

The linguist therefore only needs to edit these files, without worrying about syntax (except basic regular expression meta-characters, quantifiers and back-referencing variables), flow control, data managing, etc. D2AC consists of a series of Perl scripts, with various input and data files. See the flowchart in Figure 1 for a basic overview of D2AC (sample text provided between brackets).

D2AC takes as input a list (`List.D2AC.Ndl.txt`) of Dutch tokens to be translated, each on a separate line (i.e. tokenised text). For the current implementation we assume no erroneous input text (e.g. spelling errors), as well as no proper names, acronyms or abbreviations. As output, a list (`List.D2AC.Afr.txt`) is printed with each token on a separate line, together with a tag to indicate the nature of the output. Four tags are being used:

- `<<D2ALex>>`: If a word was translated as a false friend or non-cognate (in some cases also as a non-identical cognate; see discussion of `D2ALex.txt` below);
- `<<AfrLex>>`: If a word was translated as an identical cognate (see discussion of `AfrLex.txt` below);
- `<<Translated>>`: If a word was translated by means of the grapheme or morpheme rules; and
- `<<Untranslated>>`: If the input text could not be converted at all.

D2AC relies on two lexicons, viz. `D2ALex.txt` and `AfrLex.txt`. The former is a bilingual list, with Dutch tokens (each on a separate line) and potential Afrikaans translation equivalents (tab separated), aimed mainly at covering false friends and non-cognates (but also some cases of non-identical cognates). Where translation alternatives exist, these are separated by two forward-slashes (e.g. `<boerderij> <boerdery//plaas>`). Note that, for purposes of our current research, disambiguation is done manually; in future this could be improved by various other automatic means, such as context-dependent frequencies. For our current implementation this list was compiled manually from available lists on the internet, as well as from language learning books (specifically [11], [12] and [18]). The list currently consists of 2 696 Dutch entries.

The other lexicon, `AfrLex.txt`, is used for look-up purposes only in order to identify identical cognates. Our assumption is that false friends will be covered sufficiently by `D2ALex.txt`; hence, look-up in this list follows only after translation from `D2ALex.txt`. For purposes of this research we used the full lexicon of *Afrikaanse Speltoetser 3.0* [19], consisting of 350 943 Afrikaans entries. The lexicon includes proper names, but excludes abbreviations and other symbols.

Conversion rules in the system are executed by two conversion modules, viz. `MorphModule.pm` and `G2GModule.pm`. For both modules, a linguist can define customised variables in the file `Variables.txt`, such as diphthongs, consonants, vowels, fricatives, etc.

The module `MorphModule.pm` is used to handle systematic differences between Afrikaans and Dutch, which can be observed on the morphosyntactic level. Regular expressions in this module are based on the rules defined by a linguist in `MorphRules.txt`, and typically include rules like `<lijk> <lik>`, or `<atie> <asie>`. In principle, only morphs and allomorphs are included in `MorphRules.txt`.

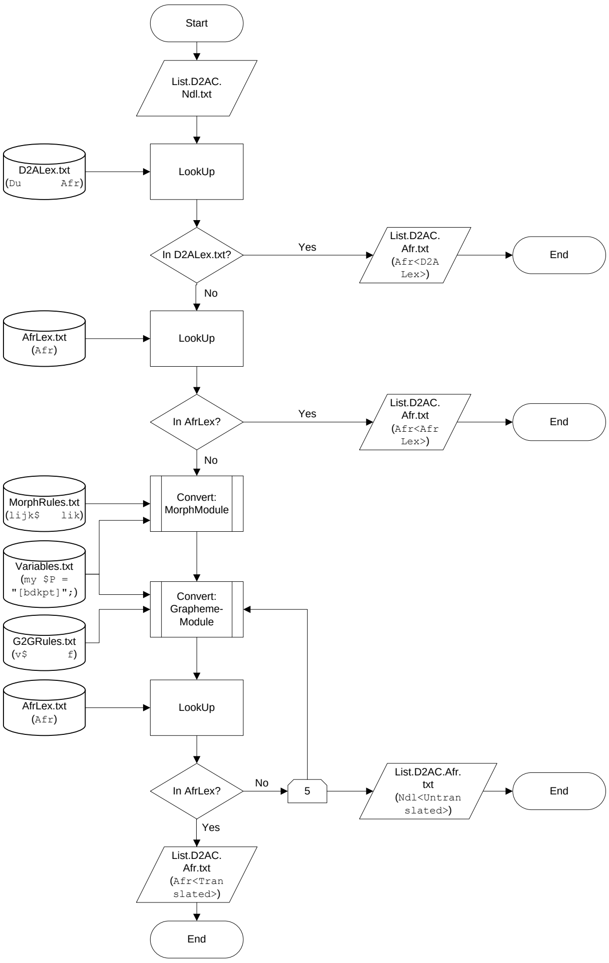

Figure 1: Basic flow-diagram of D2AC

Similarly, the module `G2GModule.pm` converts Dutch graphemes to Afrikaans graphemes in a systematic way, based on the rules included in `G2GRules.txt`. In principle, `G2GRules.txt` should only contain rules that apply on a sub-morphemic level, such as clusters of vowels or consonants (e.g. `<auw>` `<ou>`, or `<sch>` `<sk>`).

Although these two modules are in practice very similar, the distinction between the two was linguistically motivated: for purposes of linguistic research, we wanted to keep differences on the orthographic and morphosyntactic levels separate. It was assumed that this would enable us a better grip on: (1) the number of times certain rules could be executed. For example, `MorphModule.pm` is executed only once, while `G2GModule.pm` could iterate up to five times. This decision was based on the assumption that a Dutch word would only have one morpheme (or combination of morphemes) to convert, while, on the other hand, a single word could contain multiple graphemes for conversion (e.g. Du. *schrijf* vs. Afr. *skryf* 'write'); and (2) rule-ordering, which is one of the most difficult aspects of rule-based systems. The rules in the transfer component are currently (mostly) ordered according to their level of specificity: more specific rules are applied before more general rules.

5. Results

5.1. Experiment 1

For the first evaluation experiment, 500 words were randomly extracted from the 5 000 most frequent words in the *Spoken Dutch Corpus* [20]. Capitalised words (e.g. proper names and acronyms), abbreviations, and interjections were removed and replaced with other randomly selected words from the CGN frequency list. The resulting wordlist was manually translated, taking care to ensure that all possible translation alternatives were added. D2AC was then used to translate the 500 Dutch words and the output was compared with the manual translations. A breakdown of the results is given in Table 1.

	# tags assigned	**# correct tags**
`<D2ALex>`	58	58
`<AfrLex>`	138	138
`<Translated>`	162	146
`<Untranslated>`	142	0
TOTAL	**500**	**342**

Table 1: Word-level evaluation

The evaluation shows that 68.4% of the words in the list (i.e. 342 words) were translated correctly into at least one of the given translation alternatives. The conversion modules of D2AC (`MorphModule.pm` and `G2GModule.pm`) translated 162 words (tagged as `<translated>`) of which 16 resulted in incorrect translations (e.g. Du. *trokken* 'pulled' translated to Afr. *trokke* 'trucks'; Du. *viel* 'fell' translated to Afr. *wiel* 'wheel'). The relatively high precision (90.1%) of the conversion modules is due to the look-up in `AfrLex.txt`, since rules are applied iteratively until the resulting string is found in `AfrLex.txt`.

In order to determine how best to improve D2AC, we analysed the 142 words that D2AC left untranslated (see Table 2). A total of 96.5% of untranslated words could be ascribed to three causes: Firstly, more than 27% of untranslated words are due to the order of the rules in the transfer component. These words contain orthographic differences, which should have been handled by the rules that are currently included in `G2GRules.txt`. For example, Du. *dochters* should have been translated as Afr. *dogters* 'daughters' by the same rule that translates Du. *achtien* to Afr. *agtien* 'eighteen'. However, *dochters* is left untranslated, while *achtien* is translated correctly.

Cause	**# of untranslated words**	**% of untranslated words**
Flemish word	1	0.7%
Not in `MorphRules`	4	2.8%
Rule-ordering	39	27.5%
Not in `D2ALex`	43	30.3%
Ambiguity of *-en*	55	38.7%
TOTAL	**142**	**100%**

Table 2: Error analysis of word-level evaluation

Secondly, almost a third of the untranslated words (30.3%) are non-cognates or false friends, which should have been included in `D2ALex.txt`. The fact that so many frequently used Dutch words in this category are left untranslated shows that `D2ALex.txt` is not comprehensive enough. The list will need to be expanded, and (freely available) resources, other than the internet or language learning books, will need to be explored for this purpose.

Thirdly, the majority of untranslated words (38.7%) end in the ambiguous string *-en,* which can be converted in one of four ways: (1) it can be retained (e.g. Du. *rijkswapen* vs. Afr. *rykswapen,* 'state weapon'); (2) the *-n* can be deleted (e.g. Du. *vluchtelingen* vs. Afr. *vlugtelinge* 'refugee'); (3) the *-en* can be deleted (e.g. Du. *verschijnen* vs. Afr. *verskyn* 'appear'); or (4) the *-en* can be replaced with *-s* (e.g. Du. *kinderen* vs. Afr. *kinders* 'children'). In addition to solving the ambiguity of *-en,* morphonological changes also need to be kept in mind when removing *-en* from Dutch words (e.g. Du. *herstellen* should be changed to Afr. *herstel* 'repair', and Du. *stimuleren* should be changed to Afr. *stimuleer* 'stimulate').

Since *-en* is a highly productive Dutch morpheme (used, amongst others, to indicate plural and the infinitive form of the verb), it is handled in the `MorphModule.pm`, which is executed only once. In practice, this means that `MorphModule.pm` will change Du. *uitgekomen* 'came out' to **uitgekoom*, since the rule handling the morphonological change is more specific and therefore occurs before the rule that simply removes *-en.* The resulting string will then be sent to `G2GModule.pm` (which will not be able to apply any rules to **uitgekoom*) before it is looked-up, but not found, in the Afrikaans lexicon; the untranslated Du. *uitgekomen* is therefore printed to the output. However, if the input string had been sent back to `MorphModule.pm`, the next rule would have removed the *-en*, resulting in the correct Afrikaans word *uitgekom.* We therefore hypothesised that the large number of untranslated words ending in *-en* is a result of the architecture of the system, and not of the quality or comprehensiveness of the rules.

To test this hypothesis, we moved the entries in `MorphRules.txt` to `G2GRules.txt` (thereby ensuring that all the rules in the system are applied iteratively), and evaluated the resulting system (D2AC$_{comb}$) on the same test set. The results (see Table 3) show a marked improvement: D2AC$_{comb}$ was able to translate 71% of the words into at least one correct translation equivalent, resulting in an error reduction of 8.2% (i.e. 71% accuracy, compared to D2AC's accuracy of 68.4%). The precision of the conversion modules (i.e. `G2GModule.pm` only) also improved ever so slightly from 90.1% in D2AC to 90.8% in D2AC$_{comb}$.

	# tags assigned	# correct tags
`<D2ALex>`	58	58
`<AfrLex>`	138	138
`<Translated>`	175	159
`<Untranslated>`	129	0
TOTAL	**500**	**355**

Table 3: Word-level evaluation with D2AC$_{comb}$

In spite of these improvements, D2AC$_{comb}$ is still not able to translate words ending in *-en,* since these cases still amount to 52 instances (i.e. 3 less than with D2AC). Another improvement by D2AC$_{comb}$ over D2AC could be observed with regard to rule ordering, with 10 fewer words left untranslated; nonetheless, it would still be necessary to re-examine the order in which the rules are executed to ensure optimal accuracy of the system.

5.2. Experiment 2

Despite the fact that our current research is not aimed at providing a fully-fledged Dutch-to-Afrikaans machine translation system, we did a rudimentary, informal experiment to get an impression of how D2AC compares to another available solution, the Dutch-Afrikaans Google Translate (GT), when translating sentences. To do this comparative evaluation, we selected 26 random sentences (totalling 251 words) from the development test set that was used to evaluate the Dutch-English machine translation system in the METIS II project [21]. Four different translators prepared reference translations, which were used to calculate BLEU scores [22] for the translations produced by D2AC and GT. The results of the evaluation are shown in Table 4.

	D2AC	GT
% of 1-gram matches	54.7	63.0
% of 2-gram matches	29.1	35.7
% of 3-gram matches	19.6	25.3
% of 4-gram matches	12.9	17.5
BLEU	0.2519	0.3162

Table 4: Results of sentence-level evaluation

It is not surprising that GT achieves a higher BLEU score than D2AC in the automatic evaluation, since D2AC was only developed to do lexical transfer and not fully-fledged machine translation. However, given the fact that D2AC currently contains no rules to handle syntactic differences, it was anticipated that GT would have a much higher percentage of 3- and 4-gram matches than D2AC. The considerable difference between the percentages of 1- and 4-gram matches (of both D2AC and GT) might indicate that phenomena related to word-order could be an important aspect when doing machine translation from Dutch to Afrikaans (contrary to what we have expected initially).

A human assessment of the translation output confirms that D2AC is not able to handle syntactic differences (such as word-order changes and the double negation), in addition to leaving a large number of Dutch words untranslated. The fact that D2AC is limited in producing all possible translation alternatives also has a negative effect on the system's translation quality. Nonetheless, the same problems are observed in the GT translations. In addition, GT seems to translate Dutch compounds ineffectively, since compounds are consistently split into constituents before being translated (e.g. Du. *vervoersituaties* is incorrectly translated to **vervoer situasies* instead of the correct Afr. *vervoersituasies* 'transport situations').

While acknowledging that this was a very small-scale evaluation, and based on the fact that the translation quality of D2AC compares well with that of GT, we come to the conclusion that the conversion of Dutch to Afrikaans using rule-based techniques is worthy of further investigation and development, most probably also for purposes of fully-fledged machine translation.

6. Future work

Through our experiments on converting Dutch to Afrikaans, we confirmed that handling of especially verb conjugations, false friends and non-cognates require close attention. Most of these issues could be addressed by refining and/or extending the bilingual translation list (`D2ALex.txt`) automatically – for instance by using the CELEX database to complete verb paradigms and plural forms of nouns, and by using the approach of [17] to extend the list of false friends. In addition, attention should also be paid to rule-ordering, addition of a few extra rules, as well as iteration of rules and the effect thereof on the greediness of D2AC.

Work in the near future will include not only optimisation of D2AC, but also reversal of the process to develop an Afrikaans-to-Dutch convertor (A2DC) in order to experiment with technology transfer between these two languages. Although this will not be a trivial process, we have now learned enough lessons through our current research to motivate further effort and investigation. In the end, we believe that this will give us insight in the nature of closely-relatedness between languages, so that the approach could be extended to other (resource-scares/South African) languages.

7. Acknowledgements

Gerhard van Huyssteen is jointly affiliated with the Centre for Text Technology (CTexT) at the North-West University, Potchefstroom. Support by NWU is hereby acknowledged.

Part of this research was made possible through a research grant by the South African National Research Foundation (FA207041600015).

We would like to extend our gratitude to the following people who were involved in various aspects of the project: Kirsten Arnauts, Veronique de Gres, Shanna Pettens, Martin Puttkammer, Carla-Mari van den Heever, and Daan Wissing. All fallacies remain ours.

8. References

[1] Rayner, M., et al., “Recycling lingware in a multilingual MT system”, In Burstein, J., and Leacock, C. (eds.), *From research to commercial applications: making NLP work in practice,* ACL, Somerset, 1997, pp 65-70.

[2] Scannell, K., “Machine translation for closely related language pairs”, *Proc. of the LREC2006 workshop on strategies for developing MT for minority languages,* Genoa, 2006, pp 103-107.

[3] Arnold, D., et al., *Machine translation: an introductory guide,* NCC Blackwell, London, 1994.

[4] Hajič, J., Hric, J., and Kuboň, V., “Machine translation of very close languages”, *Proc. of the 6th conf. on applied NLP,* Seattle, 2000, pp 7-12.

[5] Corbí-Bellot, A.M., et al., “An open-source shallow-transfer machine translation engine for the Romance languages of Spain”, *Proc. of the 10th an. EAMT Conf.,* Budapest, 2005.

[6] Dyvik, H., “Exploiting structural similarities in machine translation”, *Computers and the Humanities*, 28:225-234, 1987.

[7] Fat, J.G., *T2CMT: Tagalog-to-Cebuano machine translation,* MS Thesis, De La Salle University, Manila, 2004.

[8] Tantuğ, C., Adalı, E., and Oflazer, K., “Machine translation between Turkic languages”, *Proc. of ACL 2007 companion volume,* Prague, 2007, pp 189-192.

[9] Homola, P., and Kuboñ, V., “Improving machine translation between closely related Romance languages”, *12th EAMT conf.,* Hamburg, 2008, pp 73-77.

[10] De Villiers, M., *Nederlands en Afrikaans,* Nasou, Goodwood, 1978.

[11] Ehlers, D., and Van Beek, P., *Oranje boven. Nederlands voor Zuid-Afrika*, Protea Boekhuis, Pretoria, 2004.

[12] Jansen, E., and Olivier, G., *Praktiese Nederlands,* Academica, Pretoria, 1986.

[13] Scholtz, J. du P., *Wording en ontwikkeling van Afrikaans,* Tafelberg, Kaapstad, 1980.

[14] Van Schoor, J.L., *Die grammatika van Standaard-Afrikaans,* Lex Patria, Kaapstad, 1983.

[15] Van Marle, J. “De derivationele morfologie: een vergeten hoofdstuk uit de geschiedenis van het Afrikaans”, In Olivier, G., and Coetzee, A.E. (reds.), *Nuwe perspektiewe op die geskiedenis van Afrikaans,* Southern Boekuitgewers, Halfweghuis, 1994, pp 90-101.

[16] Carstens, W.A.M., *Norme vir Afrikaans*, J.L. van Schaik, Pretoria, 2003.

[17] Mitkov R., Pekar V., Blagoev D., and Mulloni A., “Methods for extracting and classifying pairs of cognates and false friends”, *Machine translation,* 21(1):29-53, 2007.

[18] Veltkamp-Visser, S., *Afrikaans op reis. Taalgids voor de Nederlandssprekende toerist in Zuid-Afrika*, NZAV/SAI, Amsterdam, 1995.

[19] CTexT, *Afrikaanse speltoetser 3.0,* Noordwes-Universiteit, Potchefstroom, 2005.

[20] Nederlandse Taalunie, *Corpus gesproken Nederlands 1.0,* TST-Central, Leiden, 2004, [Web:] http://www.tst.inl.nl/cgndocs/doc_English/topics/index.htm [Accessed on 2009/09/26].

[21] Vandeghinste, V., et al., “Evaluation of a Machine Translation System for Low Resource Languages: METIS-II”, *Proc. of the 6th intern. LREC,* Marrakech, 2008, pp 449-456.

[22] Papineni, K., Roukos, S., Ward, T., and Zhu, W.J., “BLEU: a method for automatic evaluation of machine translation”, *Proc. of the 40th an. meeting of the ACL,* Philadelphia, 2002, pp 311-318.