Using Technology Transfer to Advance Automatic Lemmatisation for Setswana

Hendrik J. Groenewald
Centre for Text Technology (CTexT)
North-West University
Potchefstroom 2531, South Africa
handre.groenewald@nwu.ac.za

Abstract

South African languages (and indigenous African languages in general) lag behind other languages in terms of the availability of linguistic resources. Efforts to improve or fast-track the development of linguistic resources are required to bridge this ever-increasing gap. In this paper we emphasize the advantages of technology transfer between two languages to advance an existing linguistic technology/resource. The advantages of technology transfer are illustrated by showing how an existing lemmatiser for Setswana can be improved by applying a methodology that was first used in the development of a lemmatiser for Afrikaans.

1 Introduction

South Africa has eleven official languages. Of these eleven languages, English is the only language for which ample HLT resources exist. The rest of the languages can be classified as so-called “resource scarce languages”, i.e. languages for which few digital resources exist. However, this situation is changing, since research in the field of Human Language Technology (HLT) has enjoyed rapid growth in the past few years, with the support of the South African Government. Part of this development is a strong focus on the development of core linguistic resources and technologies. One such a technology/resource is a lemmatiser.

The focus of this article is on how technology transfer between two languages can help to improve and fast track the development of an existing linguistic resource. This is illustrated in the way that an existing lemmatiser for Setswana is improved by applying the method that was first used in the development of a lemmatiser for Afrikaans.

The rest of this paper is organised as follows: The next section provides general introductory information about lemmatisation. Section 3 provides specific information about lemmatisation and the concept of a lemma in Setswana. Section 4 describes previous work on lemmatisation in Afrikaans. Section 5 gives an overview of memory based learning (the machine learning techniques used in this study) and the generic architecture developed for machine learning based lemmatisation. Data requirements and the data preparation process are discussed in Section 6. The implementation of a machine learning based lemmatiser for Setswana is explained in Section 7, while some concluding remarks and future directions are provided in Section 8.

2 Lemmatisation

Automatic Lemmatisation is an important process for many applications of text mining and natural language processing (NLP) (Plisson *et al*, 2004). Within the context of this research, lemmatisation is defined as a simplified process of morphological analysis (Daelemans and Strik, 2002), through which the inflected forms of a word are converted/normalised under the lemma or base-form.

For example, the grouping of the inflected forms 'swim', 'swimming' and 'swam' under the base-form 'swim' is seen as an instance of lemmatisation. The last part of this definition applies to this research, as the emphasis is on recovering the base-form from the inflected form of the word. The base-form or lemma is the simplest form of a word as it would appear as headword in a dictionary (Erjavec and Džeroski, 2004).

Lemmatisation should, however, not be confused with stemming. Stemming is the process whereby a word is reduced to its stem by the removal of both inflectional and derivational morphemes (Plisson *et al*, 2004). Stemming can thus be viewed as a "greedier" process than lemmatisation, because a larger number of morph-

Proceedings of the EACL 2009 Workshop on Language Technologies for African Languages – AfLaT 2009, pages 32–37,
Athens, Greece, 31 March 2009. ©2009 Association for Computational Linguistics

emes are removed by stemming than lemmatisation. Given this general background, it would therefore be necessary to have a clear understanding of the inflectional affixes to be removed during the process of lemmatisation for a particular language.

There are essentially two approaches that can be followed in the development of lemmatisers, namely a rule-based approach (Porter, 1980) or a statistically/data-driven approach (Chrupala, 2006). The rule-based approach is a traditional method for stemming/lemmatisation (i.e. affix stripping) (Porter 1980; Gaustad and Bouma, 2002) and entails the use of language-specific rules to identify the base-forms (i.e. lemmas) of word forms.

3 Lemmatisation in Setswana

The first automatic lemmatiser for Setswana was developed by Brits (2006). As previously mentioned, one of the most important aspects of developing a lemmatiser in any language is to define the inflectional affixes that need to be removed during the transformation from the surface form to the lemma of a particular word. In response to this question, Brits (2006) found that only stems (and not roots) can act independently as words and therefore suggests that only stems should be accepted as lemmas in the context of automatic lemmatisation for Setswana.

Setswana has seven different parts of speech. Brits (2006) indicated that five of these seven classes cannot be extended by means of regular morphological processes. The remaining two classes, namely nouns and verbs, require the implementation of alternation rules to determine the lemma. Brits (2006) formalized rules for the alterations and implemented these rules as regular expressions in FSA 6 (Van Noord, 2002), to create finite state transducers. These finite state transducers generated C++ code that was used to implement the Setswana lemmatiser. This lemmatiser achieved a linguistic accuracy figure of 62,17%, when evaluated on an evaluation subset of 295 randomly selected Setswana words. Linguistic accuracy is defined as the percentage of words in the evaluation set that was correctly lemmatised.

4 *Lia*: Lemmatiser for Afrikaans

In 2003, a rule-based lemmatiser for Afrikaans (called *Ragel* – "*Reëlgebaseerde Afrikaanse Grondwoord- en Lemma-identifiseerder*") [Rule-Based Root and Lemma Identifier for Afrikaans] was developed at the North-West University (RAGEL, 2003). *Ragel* was developed by using traditional methods for stemming/lemmatisation (i.e. affix stripping) (Porter, 1980; Kraaij and Pohlmann, 1994) and consists of language-specific rules for identifying lemmas. Although no formal evaluation of *Ragel* was done, it obtained a disappointing linguistic accuracy figure of only 67% in an evaluation on a random 1,000 word data set of complex words. This disappointing result motivated the development of another lemmatiser for Afrikaans.

This "new" lemmatiser (named *Lia* – "Lemma-identifiseerder vir Afrikaans" [Lemmatiser for Afrikaans]) was developed by Groenewald (2006). The difference between *Ragel* and *Lia* is that *Lia* was developed by using a so-called data driven machine learning method. Machine learning requires large amounts of annotated data. For this purpose, a data set consisting of 73,000 lemma-annotated words were developed. *Lia* achieves a linguistic accuracy figure of 92,8% when trained on this data set. This result confirms that the machine learning based approach outperforms the rule-based approach for lemmatisation in Afrikaans.

The increased linguistic accuracy figure obtained with the machine learning based approach motivated the research presented in this paper. Since *Ragel* and the rule-based Setswana lemmatiser obtained comparable linguistic accuracy figures, the question arises whether the application of machine learning techniques, together with the methodology and architecture developed for *Lia*, can also be utilised to improve on the linguistic accuracy figure obtained by the Setswana rule-based lemmatiser.

5 Methodology

5.1 Memory Based Learning

Memory based learning (Aha *et al*, 1991) is based on the classical *k*-NN classification algorithm. *k*-NN has become known as a powerful pattern classification algorithm (Daelemans *et al*, 2007), and is considered the most basic instance-based algorithm. The assumption here is that all instances of a certain problem correspond to points in the *n*-dimensional space (Aha *et al*, 1991). The nearest neighbours of a certain instance are computed using some form of distance metric (X,Y). This is done by assigning the most frequent category within the found set of most similar example(s) (the *k*-nearest neighbours) as the category of the new test example. In case of a tie amongst categories, a tie-breaking resolution method is used.

The memory based learning system on which *Lia* is based, is called TiMBL (Tilburg Memory-Based Learner). TiMBL was specifically developed with NLP tasks in mind, but it can be used successfully for classification tasks in other domains as well (Daelemans *et al*, 2007).

5.2 Architecture

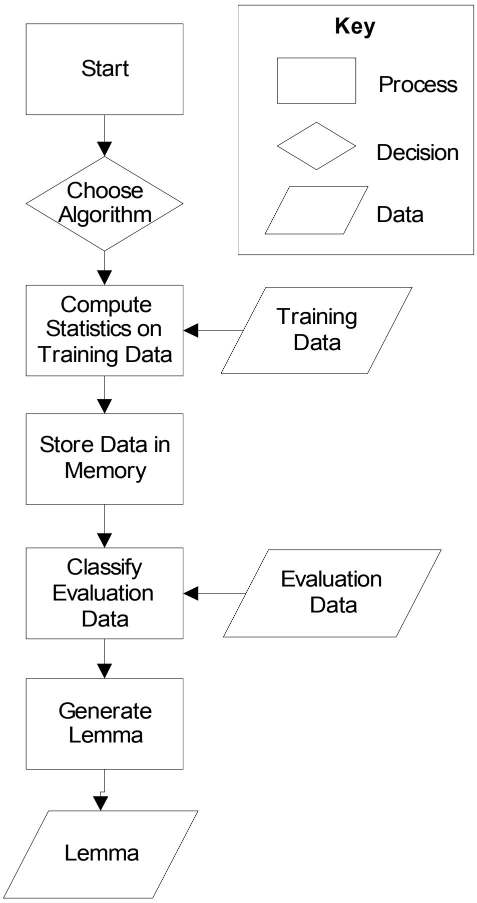

Figure 1. Generic Architecture of the Machine Learning Based Lemmatiser.

The architecture presented in this subsection was first developed and implemented for *Lia*, the machine learning based lemmatiser for Afrikaans. This same architecture was used for the development of the machine learning based lemmatiser for Setswana. The first step in this “generic” architecture consists of training the system with data. During this phase, the training data is examined and various statistical calculations are computed that aid the system during classification. This training data is then stored in memory as sets of data points. The evaluation instance(s) are then presented to the system and their class is computed by interpolation to the stored data points according to the selected algorithm and algorithm parameters. The last step in the process consists of generating the correct lemma(s) of the evaluation instance(s), according to the class that was awarded during the classification process. The generic architecture of the machine learning based lemmatiser is illustrated in Figure 1.

6 Data

6.1 Data Size

A negative aspect of the Machine Learning method for developing a lemmatiser is that a large amount of lemma-annotated training data is required. Currently, there is a data set available that contains only 2,947 lemma-annotated Setswana words. This is the evaluation data set constructed by Brits (2006) to evaluate the performance of the rule-based Setswana lemmatiser. A data set of 2,947 words is considered to be very small in machine learning terms.

6.2 Data Preparation

Memory based learning requires that lemmatisation be performed as a classification task. The training data should therefore consist of feature vectors with assigned class labels (Chrupala, 2006). The feature vectors for each instance consist of the letters of the inflected word. The class labels contain the information required to transform the involved word form from the inflected form to the lemma.

The class labels are automatically derived by determining the character string (and the position thereof) to be removed and the possible replacement string during the transformation from word-form to lemma. This is determined by firstly obtaining the longest common substring between the inflected word and the manually identified lemma. Once the longest common substring is known, a comparison of the remaining strings in the inflected word form and the lemma indicates the strings that need to be removed (as well as the possible replacement strings) during the transformation from word form to lemma. The positions of the character string to be removed are annotated as *L* (left) or *R* (right).

If a word-form and its lemma are identical, the class awarded will be *“0”*, denoting that the word should be left in the same form. This annotation scheme yields classes like in column four of Table 1.

Inflected Word-Form	Manually Identified Lemma	Longest Common Substring	Automatically Derived Class
matlhwao	letlhwao	hwao	Lma>le
menoganya	menoga	menoga	Rya>
itebatsa	lebala	ba	Lit>lRtsa>la

Table 1. Data Preparation and Classes.

For example, Table 1 shows that the class of "matlhwao" is *Lma>le*. This means that the string "ma" needs to be replaced by the string "le" (at the left hand side of the word) during the transformation from the inflected form "matlhwao" to the lemma "letlhwao". Accordingly, the class of the word "menoganya" is *Rya>*, denoting the string "ya" should be removed at the right-hand side of the inflected form during lemmatisation. In this particular case, there is no replacement string. Some words like "itebatsa" undergo alterations to both sides of the inflected form during lemmatisation. The class *Lit>lRtsa>la* indicates that the string "it" must be replaced at the left-hand side of the word with the letter "l", while the string "tsa" should be replaced with the string "la" at the right-hand side of the word.

An example of the training of data of the lemmatiser is shown in Figure 2. The data is presented in C4.5 format (Quinlan, 1993) to the memory based learning algorithm, where each feature is separated by a comma. The algorithm requires that every instance must have the same number of features. In order to achieve this, it was decided that each instance should contain 20 features. 20 features were chosen, since less than 1% of the words in the data contained more than 20 letters. All instances were formatted to contain 20 features by adding underscores to the words that contained less than 20 features. The result of this process is displayed in Figure 2.

```
_,_,_,_,_,_,_,_,_,_,t,s,o,g,a,t,s,o,g,a,0
_,_,_,_,_,_,_,_,_,_,e,d,i,m,o,l,a,n,y,a,Rnya>
_,_,_,_,_,_,_,_,_,_,_,_,_,d,i,n,y,e,p,o,Ldi>
_,_,_,_,_,_,_,_,_,_,t,s,i,s,e,d,i,t,s,e,Ltsisedi>Rse>la
```

Figure 2. Training Data in C4.5 Format.

7 Implementation

Each of the 2,947 lemma-annotated words in the evaluation data of the rule-based Setswana lemmatiser was formatted in C4.5 format. The data was then split up further into a training data set, consisting of 90% of all the data, with an evaluation set consisting of 10% of all the data. A machine learning based lemmatiser was trained (by utilising default parameter settings) and evaluated with these two datasets. This lemmatiser obtained an accuracy figure of 46.25%. This is a disappointing result when compared to the linguistic accuracy figure of 62.71% obtained with the rule-based Setswana lemmatiser when evaluated on the same data set. Algorithmic parameter optimisation with *PSearch* (Groenewald, 2008) resulted in an improved accuracy figure of 58.98%. This represents an increase of 12.73%, but is still less than the accuracy figure obtained by the rule-based lemmatiser.

Error analysis indicated that in some cases the class predicted by TiMBL is conspicuously wrong. This is evident from instances shown in Table 2, where the assigned classes contain strings that need to be removed that is not present in the inflected forms.

Inflected Word	Correct Class	Assigned Class
tlamparele	*Re>a*	Lmm>bRele>a
phologileng	*Rileng>a*	Regileng>a

Table 2. Instances with Incorrectly Assigned Classes.

Inflected Word	Assigned Class	Class Distribution
tlamparele	*Lmm>bRele>a*	0 0.934098 Re>a 1.82317 Rele>a 0.914829 Lmm>bRele>a 1.96103
phologileng	*Regileng>a*	Rileng>a 3.00014 Relang>a 1.24030 Regileng>a 4.20346

Table 3. Instances Containing Additional Class Distribution Information.

For example, the class assigned to the second instance in Table 2, is *Regileng>a*. This means that the string "egileng" must be replaced with the character “a” at the right-hand side of the word. However, the inflected word “phologileng” does not contain the string “egileng”, which means that the assigned class is sure to be incorrect. This problem was overcome by utilizing the TiMBL option (+v db) that adds class distribution in the nearest neighbour set to the output file. The result of this is an additional output that contains the class distribution information shown in Table 3. The class distribution information contains the nearest classes with their associated distances from the involved evaluation instance.

A post-processing script that automatically recognises this type of incorrectly assigned class and replaces the incorrect class with the second most likely class (according to the class distribution) was developed. The result of this was a further increase in accuracy to 64.06%. A summary of the obtained results is displayed in Table 4.

Method	Linguistic Accuracy
Rule-based	62.17%
Machine Learning with default parameter settings	46.25%
Machine Learning with optimised parameter settings	58.9%
Machine Learning with optimised parameter settings and class distributions	64.06%.

Table 4. Summary of Results.

8 Conclusion

The best results obtained by the machine learning based Setswana lemmatiser was a linguistic accuracy figure of 64.06%. This represents an increase of 1.9% on the accuracy figure obtained by the rule-based lemmatiser. This seems to be a small increase in accuracy compared to the 25.8% increase obtained when using a machine learning based method for Afrikaans lemmatisation. The significance of this result becomes evident when considering the fact that it was obtained by training the machine learning based Setswana lemmatiser with a training data set consisting of only 2,652 instances. This data set is very small in comparison with the 73,000 instances contained in the training data of *Lia.*

The linguistic accuracy figure of 64.06% furthermore indicates that a machine learning based lemmatiser for Setswana that yields better results than a rule-based lemmatiser can be developed with a relatively small data set. We are confident that further increases in the linguistic accuracy figure will be obtained by enlarging the training data set. Future work will therefore entail the employment of bootstrapping techniques to annotate more training data for improving the linguistic accuracy of the machine learning based Setswana lemmatiser.

The most important result of the research presented in this paper is, however, that existing methodologies and research can be applied to fast-track the development of linguistic resources or improve existing linguistic resources for resource-scarce languages, a result that is especially significant in the African context.

Acknowledgments

The author wishes to acknowledge the work of Jeanetta H. Brits, performed under the supervision of Rigardt Pretorius and Gerhard B. van Huyssteen, on developing the first automatic lemmatiser for Setswana.

References

David W. Aha, Dennis Kibler and Marc K. Albert. 1991. Instance-Based Learning Algorithms. *Machine Learning*, 6:37-66.

Jeanetta H. Brits. 2006. *Outomatiese Setswana Lemma-identifisering ‘Automatic Setswana Lemmatisation’*. Master’s Thesis. North-West University, Potchefstroom, South Africa.

Gregorz Chrupala. 2006. Simple Data-Driven Context-Sensitive Lemmatization. *Proceedings of SEPLN 2006.*

Walter Daelemans, Antal Van den Bosch, Jakub Zavrel and Ko Van der Sloot. 2007. *TiMBL: Tilburg MemoryBased Learner*. Version 6.1, Reference Guide. ILK Technical Report 07-03.

Walter Daelemans and Helmer Strik. 2002. Actieplan Voor Het Nederlands in de Taal- en Spraaktechnologie: Prioriteiten Voor Basisvoorzieningen. *Report for the Nederlandse Taalunie*. Nederlandse Taalunie.

Tomaž Erjavec and Saso Džeroski. 2004. Machine Learning of Morphosyntactic Structure: Lemmatising Unknown Slovene Words. *Applied Artificial Intelligence*, 18(1):17-40.

Tanja Gaustad and Gosse Bouma. 2002. Accurate Stemming of Dutch for Text Classification. *Language and Computers,* 45 (1):104-117.

Hendrik J. Groenewald. 2007. *Automatic Lemmatisation for Afrikaans*. Master’s Thesis. North-West University, Potchefstroom, South Africa.

Hendrik J. Groenewald. 2008. *PSearch 1.0.0*. North-West University, Potchefstroom, South Africa.

Wessel Kraaij and Renee Pohlmann. 1994. Porter's Stemming Algorithm for Dutch. *Informatiewetenschap 1994: Wetenschaplike bijdraen aan de derde STINFON Conferentie*. 1(1):167-180.

Joel Plisson, Nada Lavrac and Dunja Mladenić. 2004. A Rule-based Approach to Word Lemmatization. *Proceedings C of the 7th International Multi-Conference Information Society IS 2004*, 1(1):83-86.

Martin Porter. 1980. An Algorithm for Suffix Stripping. *Program,* 14 (3):130-137.

John R. Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, USA.

RAGEL. 2003. Reëlgebaseerde Afrikaanse Grondwoord- En Lemma-identifiseerder 'Rule-based Afrikaans Stemmer and Lemmatiser. http://www.puk.ac.za/opencms/export/PUK/html/fakulteite/lettere/ctext/ragel.html.> 11 January 2009.

Gertjan Van Noord. 2002. Finite State Utilities.
< http://www.let.rug.nl/~vannoord/Fsa/>.
12 January 2009.