EDUCATING *LIA*: THE DEVELOPMENT OF A LINGUISTICALLY ACCURATE MEMORY-BASED LEMMATISER FOR AFRIKAANS

Hendrik J. Groenewald
Centre for Text Technology,
North-West University,
2531 Potchefstroom, South Africa
eeihjg@puk.ac.za

Abstract

This paper describes the development of a memory-based lemmatiser for Afrikaans called *Lia*. The paper commences with a brief overview of Afrikaans lemmatisation and it is indicated that lemmatisation is seen as a simplified process of morphological analysis within the context of this paper. This overview is followed by an introduction to memory-based learning – the machine learning technique that is used in the development of the Afrikaans lemmatiser. The deployment of *Lia* is then discussed with specific emphasis on the format of the training and testing data that is used. The Afrikaans lemmatiser is then evaluated and it is indicated that *Lia* achieves a linguistic accuracy figure of over 90%. The paper concludes with some ideas on future work that can be done to improve the linguistic accuracy of the Afrikaans lemmatiser.

Keywords: Natural Language Processing, Machine Learning, Lemmatisation, Afrikaans, Memory-Based Learning

1. Introduction

In 2003, a rule-based lemmatiser for Afrikaans (called *Ragel* – "*Reelgebaseerde Afrikaanse Grondwoord- en Lemma-identifiseerder*") [Rule-Based Root and Lemma Identifier for Afrikaans] was developed at the North-West University and is currently included in a spelling checker for Afrikaans (Afrikaanse Speltoetser 3.0). *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 word-forms in the lexicon of the spelling checker. However, *Ragel* cannot be considered either a "pure" lemmatiser or a "pure" stemmer in the true sense of the word, since it was developed specifically for purposes of spelling checking. In this sense, both derived and inflected word-forms that are not in the lexicon of the spelling checker are analysed by *Ragel*, only until a word in the lexicon is found, whether that word is a lemma or not (e.g. *"ontbossing"* 'deforestation' will be analysed as *"ontbos"* 'deforest' and not necessarily as *"bos"* 'forest'). 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 dataset.

The purpose of this study is to develop a more "pure" lemmatiser for Afrikaans, using an alternative approach (i.e. memory-based learning). It is important that *Lia* [Lemma Identifier for Afrikaans] should achieve a better linguistic accuracy figure than *Ragel*, and the focus and objective are therefore to achieve a linguistic accuracy figure of at least 90%.

The following section presents background information on the problem of lemmatisation for Afrikaans and briefly discusses the inflectional morphemes used in this study. Memory-based learning and the Tilburg Memory-Based Learner (TiMBL) (Daelemans et al., 2004) are briefly introduced in Section 3, before discussing the actual development of *Lia* at length in Section 4. Here the focus will be explicitly on the architecture of the system, and the representation of the data for optimal linguistic accuracy. Section 5 describes the evaluation of *Lia*, with some general concluding remarks in Section 6.

2. Lemmatisation for Afrikaans

Within the context of this study, 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 (i.e. the simplest form of a word as it would appear as headword in a dictionary (Erjavec and Dzeroski, 2004; Hausser, 1999)) by removing inflectional affixes (Bussman, 1996). In this sense, lemmatisation should not be confused with stemming, which is the process whereby the stem of a word is retrieved by removing both inflectional and derivational morphemes from the word (Gearailt, 2005; Manning and Schutze, 1999). Also, it is usually expected of a lemmatiser to produce independent word forms, while a stemmer might also produce dependent forms, such as roots or stems (Plisson et al., 2004).

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. With regard to Afrikaans, there is still no general agreement among Afrikaans linguists on what the list of inflectional affixes should be. For instance, Combrink (1974) rejects the notion of inflection for Afrikaans altogether and describes it as a useless Latinism. On the other hand, linguists such as Du Toit (1982), Van Schoor (1983), and

Carstens (1992) have each defined their own lists of inflectional morphemes for Afrikaans. Although there is some degree of agreement between these lists, differences still exist. For the purpose of this study, we therefore simply accept all the inflectional categories presented by the previously-mentioned three authors. These inflectional categories are:

1 Plural (e.g. the *"-s"* in *"tafels"*, 'tables' and the *"e"* in *"mense"*, 'humans')

2 Degrees of comparison (e.g. the *"-er"* or *"-ste"* in *"kleiner"* 'smaller' and *"kleinste"* 'smallest')

3 Diminutive form (e.g. the *"-jie"* in *"hondjie"* 'puppy')

4 Past Tense (e.g. the *"ge-"* in *"geloop"* 'walked')

5 Past Participle form (e.g. the *"ge- -te"* in *"getrapte"* 'trampled')

6 Infinitive (e.g. the *"-e"* in *"drinke"* 'drink')

7 Attributive (e.g. the *"-e"* in *"pragtige"* 'exquisite')

8 Partitive Genitive (e.g. the *"-s"* in *"pragtigs"* 'exquisite')

Lia, or any lemmatiser for Afrikaans, should therefore be able to remove all affixes in these eight inflectional categories, yielding linguistically correct lemmas. Although it seems easy, Afrikaans lemmatisation proves to be no trivial task; it entails more than just removing the correct affix from the word to obtain the correct lemma. *Lia* has to deal with a number of further complexities, such as:

1 A rule-based lemmatiser will tend remove the suffix *-tjie* erroneously in the case of words like *"jobskraaltjie"* (a grass species) and *"suurpootjie"* (a tortoise specie), because *-tjie* normally indicates the diminutive form. The *-tjie* in these words however does not indicate the diminutive form, as it forms part of the lemma of the word.

2 Words that contain prefixes like *aange-* and *opge-* like in *"aangedryf"* 'drove' and *"opgelaai"* 'picked up' should be lemmatised by only removing the second prefix *-ge-* in the middle of the word.

3 Words that are in the past participle form like *"ingedraaide"* 'screwed in' should be lemmatised as *"indraai"* 'screw in'. This can be confusing, because it is differs from the lemmatisation method described under (2) above.

4 Words that are in the past participle form that start with *onge-* are not lemmatised according to the manner that other past participle form words are lemmatised. Only the suffixes *-de* or *-te* should be removed during lemmatisation. *"Ongenooide"* 'uninvited' must accordingly be lemmatised as *"ongenooi"*, instead of the invalid lemma *"*onnooi"*.

5 Due to morphonological processes, some words like *"paaie"* 'roads' are not lemmatised by just removing the *-e* that indicates the plural form; a *-d* should also be appended at the end of the word during the transformation to the lemma.

The next section describes the approach taken in this research to train *Lia* to produce grammatically correct lemmas for Afrikaans words.

3. Memory-Based Learning

Previous experience with *Ragel* proved that it is quite difficult to define expert rules for accurate lemmatisation of Afrikaans word-forms. It was therefore decided to take an alternative computational approach in developing *Lia*, namely a machine-learning approach, using memory-based learning algorithms. Based on Mitchell's definition of machine learning (Mitchell, 1997), our basic assumption in this study can be formulated as follows:

> *Lia* is said to learn from a database of correctly lemmatised words (i.e. **E**xperience), with respect to lemmatisation (i.e. **T**ask) and the percentage of correctly lemmatised words (i.e. **P**erformance Measure), if its performance at lemmatisation (**T**), as measured by the percentage of correctly lemmatised words (**P**), improves as the size of the database of correctly lemmatised words is increased (**E**).

This implies that *Lia* will improve (learn) with more and more experience (i.e. a larger and better database of correctly lemmatised words), so that predictions about new cases can be made based on the outcomes of similar cases in the past (Aloaydin, 1997). In order to foster such learning, we decided to follow a memory-based learning approach to train *Lia*.

Memory-based learning is based on the classic k-Nearest Neighbour (k-NN) algorithm, which is a powerful, yet basic classification algorithm. The assumption here is that all cases of a certain problem can be represented as points in an n-dimensional space, where the nearest-neighbour points can be computed using a distance formula $\Delta(X,Y)$. The class (category) of a new case is assigned by considering the classes that are most common with the nearest neighbours of the new case (Daelemans et al., 2004). It has been proven in the past that memory-based learning could be used with great success for natural language processing (NLP) tasks such as lemmatisation (Daelemans and Strik, 2002; Baldwin and Bond, 2003; Gustafson, 1999). A possible reason for this is that each instance is viewed as equally important during the classification process. (Daelemans et al., 1999).

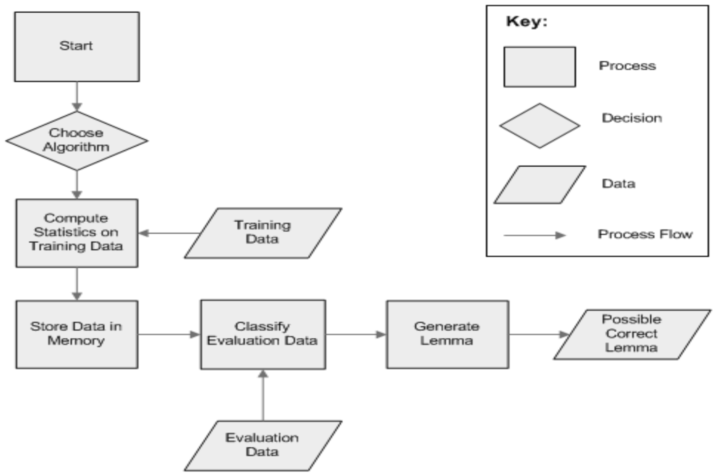
The memory-based learning system on which *Lia* is based, is called TiMBL (Tilburg Memory-Based Learner). TiMBL was specifically developed with NLP tasks in mind, but it can be used successfully for classification tasks in other domains as well (Daelemans et al., 2004).

4. *Lia*: Lemmatiser for Afrikaans

Architecture

The first step in the architecture of *Lia* consists of training the system with data. During this phase, the training data is examined and various statisti-

Figure 1. The architecture of Lia



cal 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.

Data

As was mentioned earlier, machine learning systems improve with experience. In the case of *Lia*, this *"experience"* is based on the amount of data used during training. The assumption here is that the more data *Lia* has access to during the training phase, the better the linguistic accuracy will be. The annotation of training data is, however, a labour-intensive, time-consuming process, especially for resource-scarce languages such as Afrikaans. The training data for this project was extracted from the lexicon of a spelling checker for Afrikaans that consists of 350,000 words (Afrikaanse Speltoetser 3.0). All the words that correspond in form to the inflectional forms defined for this project were extracted. For example, both the words *"geel"* 'yellow' and *"geslaap"* 'slept' were extracted during this process, because both words begin with the possible prefix *"ge-"*. The lemma of *"geslaap"* is *"slaap"* 'sleep', but the word *"geel"* 'yellow' is already a lemma. However, it is important to also train *Lia* with lemmas such as *"geel"* 'yellow', since *Lia* should not only learn how to lemmatise, but also when to lemmatise words and when not to. This extraction yielded 110,000 words, of which approximately 30% do not contain inflectional morphemes.

Defining the format of the classes was an important part of the data-construction phase. The logical way to go about the problem is to use grammatically motivated classes. For example, the class of the word *“hondjie”* ’puppy’ should then have been *-jie*, implying that the suffix *-jie* should be removed from the word to lemmatise it. This approach turns out to be problematic in some cases, such as *“beeldskone”* ’beautiful’ where the correct lemma is *“beeldskoon”*. The linguistically correct class of *“beeldskone”* is *-e* (attributive), but simply removing an *-e* at the right-hand side of *“beeldskone”* will leave us with *“beeldskon”* which is not a valid lemma. This problem was overcome by using non-grammatically motivated classes as described in the next two paragraphs.

The extracted data is annotated manually by providing the lemma for each instance, after which the class of the instance is then automatically awarded on the basis of a comparison between the word and the correct lemma by means of a Perl script. The classes are 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. The positions of the character string to be removed are annotated as *L* (left), *R* (right) and *M* (middle). If a word-form and its lemma are identical, the class awarded will be *“0”*, denoting the word should be left in the same form. This annotation scheme yields classes like in column three of Table 1.

Table 1. Data preparation for *Lia*

Extracted Word-Form	**Manually Identified Lemma**	**Automatically Derived Class**
Geel ’yellow’	*Geel* ’yellow’	*0*
Geslaap ’slept’	*Slaap* ’sleep’	*Lge>*
Hondjie ’puppy’	*Hond* ’dog’	*Rjie>*
Bote ’ships’	*Boot* ’ship’	*Rte>ot*
Omgedraaide ’turned over’	*Omdraai* ’turn over’	*MgeRde>*

The class of *“geslaap”* ’slept’ will be *Lge>*, where the *L* implies that the inflectional prefix *“ge-”* should be removed on the left-hand side of the word to lemmatise it. Accordingly, the class of the word *“bote”* ’boats’ will be *Rte>ot*, denoting the *“te”* at the right-hand side of the word should be replaced by *“ot”*. Words in the past participle form, for instance *“omgedraaide”* ’turned over’, will receive the class *MgeRde>*, meaning that the *“-ge-”* and the *“de”* should be removed respectively at the middle and at the right-hand side of the word.

This method of class assignment eliminates the generation of incorrect lemmas like *“beeldskon”*, but in turn, it produces 311 different classes which also further complicates the lemmatisation process. An example of *Lia*’s training data is shown in Figure 2. The data is presented to TiMBL in C4.5 format (Quinlan, 1993), where each feature of each instance is separated by a comma.

The data is presented in a format that ensures equal amounts of features for each instance as this is required by TiMBL. To do this, it was assumed that the longest possible word to be analysed by *Lia* would consist of not more than 38 characters.[1] Accordingly, all instances were fitted to this format and underscores were added to words shorter than 38 characters, as can be seen in Figure 2. Further experiments will be done to determine the optimal amount of features, because too many or too few features have a negative influence on *Lia*'s accuracy. Too many features also increase the classification time.

Figure 2. Training data in C4.5 format (right aligned without feature positioning)

_,g,e,e,l,0
_,g,e,s,l,a,a,p,Lge>
_,h,o,n,d,j,i,e,Rjie>
_,b,o,t,e,Rte>ot
_,o,m,g,e,d,r,a,a,i,d,e,Mge>Rde>

The training data was at first left-aligned, but this resulted in very low accuracy figures. We then realised that, since the majority of inflectional affixes are suffixes (only one inflectional prefix *"ge-"* occurs in Afrikaans, which can also be inserted between the preposition and stem in so-called particle verbs), the training data should be right-aligned. A remarkable increase in the accuracy figures was achieved by the right-alignment of the data. Right-alignment ensures that the suffix part of every word is always at the same feature position, which is not the case if the data is left-aligned.

A common mistake that *Lia* initially made was that the classes of words like *"geabsorbeerde"* 'absorbed' (class: *LgeRde*) was confused with the classes of words like *"verdofde"* 'dimmed' (class: *Rde>*). The reason for this is that the letters of the inflectional prefix *ge-* was at different feature positions for different instances when the data was right-aligned. The same confusion was experienced with words that were in the past participle form. The success achieved by right-alignment of the data lead us to define the concept of "feature-positioning", in order to reduce the amount of confusion experienced.

Figure 3. Training data in C4.5 format (right aligned with feature positioning)

g,e,_,e,l,0
g,e,_,s,l,a,a,p,Lge>
_,h,o,n,d,j,i,e,Rjie>
_,b,o,t,e,Rte>ot
o,m,_,_,_,_,_,g,e,_,d,r,a,a,i,d,e,Mge>Rde>

Feature-positioning implies that all words containing the possible prefix *"ge-"*, is treated like *"geslaap"* 'slept' in Figure 3, or alternatively like *"omgedraaide"* 'turned-over' when *"-ge-"* is inserted in a participle verb. Feature-positioning ensures that similar features are always aligned at the same feature-positions and thereby eliminates any confusion that may arise. The accuracy gained by the use of feature-positioning is presented in the next section.

A dataset consisting of 56,000 words was randomly extracted from the original dataset of 110,000 words.[2] This dataset was annotated as described above, then manually checked by linguists, after which it was used to train *Lia* for evaluation purposes.

5. Evaluation of *Lia*

Table 2. Comparison of the results obtained with Right-Aligned data vs. Feature-Positioned Right-Aligned data

	Right-Aligned	**Right-Aligned with feature-positioning**	% **Error Reduction**
Dataset 1	88.9027	90.9285	18.2549
Dataset 2	89.3118	90.8945	14.8080
Dataset 3	89.2051	91.3036	19.4397
Dataset 4	88.4225	91.1242	23.3358
Dataset 5	89.4185	91.7823	22.3390
Dataset 6	88.6893	91.3925	23.8995
Dataset 7	88.8672	90.8929	18.1958
Dataset 8	88.3514	90.6261	19.5277
Dataset 9	89.2228	91.3569	19.8020
Dataset 10	88.6893	90.7862	18.5391
Average	**88.9081**	**91.1088**	**19.8141**

The IB1 algorithm was used in this section to verify if an accuracy figure of 90% is attainable. IB1 is the basic instance-based algorithm used in TiMBL and its operation is similar to the basic k-NN algorithm. The algorithm parameters used were determined through the use of the software package *Paramsearch 1.0* (van den Bosch, 2005). *Paramsearch* provides a (possibly optimal) set of algorithm parameters that are expected to do well on the task at hand. The parameters that *Paramsearch* yielded were:

Distance Metric: Modified Value Difference Metric
Feature Weighting: Information Gain
Nearest Neighbour Count: 11
Class voting weights: Inverse Linear

Table 2 shows a comparison of the linguistic accuracy figures for the cases where the data is right-aligned, compared to the cases where feature-positioning is used. The evaluation was done by means of ten-fold cross-validation. This means that the available data is split into ten equally sized parts. Each of the parts is then used as an evaluation set while the remaining nine sets are used as training data. The results for each set are displayed in Table 2, together with the resulting percentages of error reduction obtained when using feature-positioning. The error reduction is measured as the percentage of errors that was saved by using feature-positioning data.

As was stated in the introduction, one of the aims of this study is to develop a lemmatiser for Afrikaans, with an accuracy score of at least 90%. Table 2 shows that this objective is indeed achieved by the introduction of right-aligned, feature-positioned data, which results in an average accuracy figure of 91.1088%. Table 2 also indicates that the use of right-aligned, feature-positioned data results in an average error reduction of 19.8141%.

6. Conclusion

The evaluation shows that an average linguistic accuracy of 88.9801% is obtained by training *Lia* with 56,000 words. A further improvement to 91.1088% is achieved by using feature-positioned data. The objective of this paper, namely obtaining an accuracy score of at least 90%, was successfully reached. Compared to the 67% accuracy figure for *Ragel*, this indicates that memory-based learning provides a suitable alternative to a rule-based approach considering the problem of lemmatisation for Afrikaans. This also confirms the conviction of Streiter and De Luca (2003) that example-based approaches (such as memory-based learning) offer an effective processing strategy for resource-scarce languages.

However, there is still much that can be done to improve the results obtained. Future work includes experimenting with different ways of data representation to see if further improvements in linguistic accuracy can be achieved. Memory-based learning algorithms are also very sensitive to changes in their parameter settings; experiments will therefore be done to determine the algorithm and optimal combinations of parameter settings to deliver the best performance for this particular task. We will also investigate why certain combinational settings deliver better results than other.

7. Acknowledgements

I want to thank Proff. Gerhard B. van Huyssteen, Albertus S.J. Helberg and Antal van den Bosch for their useful comments, support and various opportunities granted. I also wish to thank the National Research Foundation (NRF) for their financial support of the project (NRF Project: Afrikaans Text Technology Modules GUN: FA_20040429000591).

Notes

1. Less than 0,1% of the words in the training set consist of more than 38 characters.

2. Section 5 indicates that 56,000 words are enough data for obtaining the desired linguistic accuracy.

References

Afrikaanse Speltoetser 3.0, Thesaurus 1.0 and Hyphenator, Potchefstroom: CTexT, North-West University, 2005.

E. Aloaydin. *Introduction to Machine Learning*. Cambridge: MIT Press, 2004.

T. Baldwin and F. Bond. *A Plethora of Methods for Learning English Countability*. Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, 2003.

H. Bussman. *Routledge Dictionary of Language and Linguistics*. London: Routledge, 1996.

A. Carstens. *Basiskursus: Aspekte van die Afrikaanse Taalkunde 'Aspects of Afrikaans Linguistics'*. Bloemfontein: Patmos, 1992.

J. G. H. Combrinck. *Soek: Afrikaans se fleksie 'Wanted: The inflectional morphemes of Afrikaans'*. Taalkunde –'n Lewe 'Linguistics – a life'. Cape Town: Tafelberg, 1974.

W. Daelemans and H. Strik. *Het Nederlands in de taal- en spraaktechnologie: prioriteiten voor basisvoorzieningen*. 'Dutch in language and speech technology: priorities for basic provisions'. Dutch Language Union, 2002

W. Daelemans, A. van den Bosch and J. Zavrel. *Forgetting Exceptions is Harmful in Language Learning*. Machine Learning, 34(1):11–43, 1999.

W. Daelemans, A. Van den Bosch, J. Zavrel and K. Van der Sloot. *TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide*. ILK Technical Report 04-02, 2004.

P. J. du Toit. *Taalleer vir Onderwyser en Student 'Language learning for Teacher and Student'*. Pretoria: Academica, 1982.

T. Erjavec and S. Dzeroski. *Machine Learning of Morphosyntactic Structure: Lemmatising Unknown Slovene Words*. Applied Artificial Intelligence 18(1):17–40, 2004.

D. N. Gearailt. *Dictionary characteristics in cross-language information retrieval*. Technical report UCAM-CL-TR-616. Cambridge: University of Cambridge Computer Laboratory, 2005.

J. Gustafson, N. Lindberg and M. Lundeberg. *The August Spoken Dialogue System*. Proceedings of Eurospeech, 1999.

R. Hausser. *Foundation of Computational Linguistics: man-machine communication in natural language*. Berlin: Springer, page 516, 1999.

W. Kraaij and R. Pohlmann. *Porter's Stemming Algorithm for Dutch*. in Informatiewetenschap 1994: Wetenschaplike bijdraen aan de derde STINFON Conferentie, pages 167-180, 1994.

C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. Cambridge: The MIT Press, 1999.

T. M. Mitchell. *Machine Learning*. Boston: McGraw-Hill, 1997.

J. Plisson, N. Lavrac and D. Mladenic. *A rule based approach to word lemmatisation*. Proceedings of the 7th International Multi-conference Information Society. Ljubljana: Institut Jozef Stefan, pages 83-86, 2004.

M. Porter. *An Algorithm for Suffix Stripping*. Program 14(3):1300-137, 1980.

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

J. L. van Schoor. *Die Grammatika van Standaard-Afrikaans 'The Grammar of Standard Afrikaans'*. Cape Town: Lex Patria Publishers, 1983.

O. Streiter and E. W. de Luca. *Example-based NLP for Minority Languages: Tasks, Resources and Tools*. Proceedings of TALN 2003. Batz-sur-Mer, 11-14 June 2003.

A. van den Bosch. *Paramsearch 1.0 beta patch 24*. (2005).