Towards the Implementation of a Refined Data Model for a Zulu Machine-Readable Lexicon

Ronell van der Merwe, Laurette Pretorius, Sonja Bosch
University of South Africa
PO Box 392, UNISA 0003, South Africa
vdmerwer@unisa.ac.za, pretol@unisa.ac.za, boschse@unisa.ac.za

Abstract

The development of a machine-readable lexicon for Zulu requires a comprehensive data model for representing lexical information. This article focuses on the modelling of the verb structure in Zulu, and more specifically verbal extensions and deverbatives due to their complex recursive morphological structure. Moreover, we also show how the machine-readable lexicon, based on the proposed data model, may be embedded in an update framework. This framework also includes a finite-state morphological analyser, its guesser variant and electronically available Zulu corpora as a potential source of new lexical information. Finally implementation issues are considered in order to ensure accurate modelling and efficiency in the lexical repository.

1. Introduction

Comprehensive machine-readable (MR) lexicons remain one of the most basic language resources in human language technologies and natural language processing applications. The quest for comprehensiveness entails among others, the inclusion of all the lexical information and structure contained in paper dictionaries, and the sourcing of lexical information from electronically available Zulu corpora via an update framework. In particular, we aim at clarifying the role of the MR lexicon as a component in the update framework, as well as its relation with the other components, viz. the finite-state morphological analyser (ZulMorph), its guesser variant and electronically available corpora (cf. Pretorius and Bosch, 2003, Bosch et al., 2008). For lesser-resourced languages the development of such a resource is challenging and time consuming, and requires the optimal use of available corpora, enabling technologies and implementation approaches.

Firstly, we briefly discuss the components and processes that constitute the update framework for building and maintaining a comprehensive MR repository of lexical information for Zulu.

Secondly, we refine a comprehensive data model for such a resource, first proposed by Bosch et al. (2007:142), where it was argued that “well-defined recursion is the correct and intuitive way to capture certain linguistic information such as verbal extensions …” The importance of allowing recursion within entries in MR lexicons for the South African Bantu languages, in order to facilitate accurate modelling, is supported by Weber (2002:8). He states that “lexical databases should accommodate (1) derived forms having multiple senses and (2) derived forms ... of the bases from which they are derived”. In the original data model, verbal extensions were treated as the main source of recursion. In this paper, the data model is extended by including deverbatives, the modelling of which is also based on recursion.

Finally, we consider the suitability of various implementation approaches with specific focus on the recursive nature of the data model.

2. Update Framework

The framework for developing and updating the comprehensive MR lexicon for Zulu is shown in Figure 1. Apart from the MR lexicon, the framework includes a finite-state morphological analyser, its guesser variant, new language data in the form of electronic corpora and linguistic expertise for moderating additions derived from the Guesser.

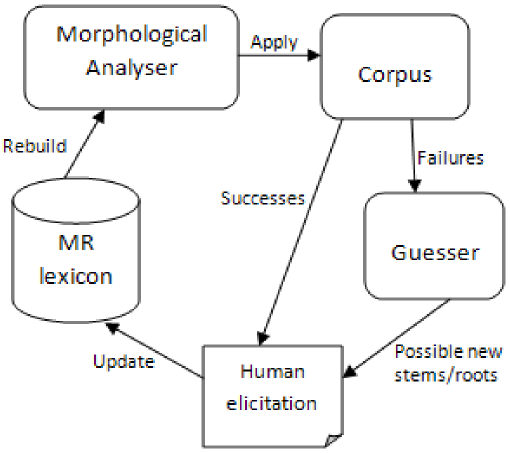

Figure 1: The Lexicon update framework

The main purpose of a comprehensive MR lexicon is to serve as a machine-readable repository of all lexical information. This resource may then be used and reused in various applications. One of its intended applications is to support the development and maintenance of a finite-state morphological analyser, e.g. ZulMorph. The morphological analyser represents only morphosyntactic information in the form of Zulu morphotactics, morphophonological alternation rules and an embedded stem/root lexicon, as automatically extracted from the MR lexicon. The guesser variant of the morphological analyser is designed to identify all phonologically possible stems/roots. This newly found candidate stems/roots are considered for linguistic validity and for inclusion in the MR lexicon.

3. Data model

The development of a comprehensive MR lexicon relies on an underlying data model for the Bantu languages that provides for storage of and access to all the units of information, both rule-based (regular) and idiosyncratic, associated with the attested words in a language. The lexicon is conceptualised as consisting of a collection of entries at the highest level. Each entry then has a tree-like structure for accommodating the relevant lexical information. The top four levels of this tree are shown in Figure 2. Each entry consists of a head (the stem) and a body, which represents general information about the stem such as its phonetic transcription, tone, morphosyntactic information (including part of speech), sense, dialect information, etymology, etc. (cf. Bosch et al., 2007) for a detailed discussion).

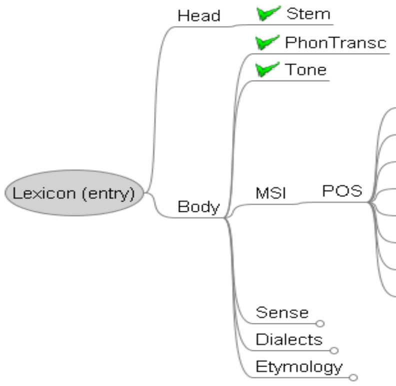

Figure 2: Top levels of the data model.

Figures 3a and 3b provide details of the modelling of the verb, exhibiting two sources of recurrence, viz. verbal extensions and nominal suffixes of deverbatives (more details in 3.1 and 3.2), which results in the generation of complex data such as nested derivational forms and their sense information. These (recursive) characteristics present specific challenges in developing a MR lexicon for the Zulu language. The Kleene * indicates 0 or more occurrences, the Kleene + indicates 1 or more occurrences, | optionality, and √ a leaf node.

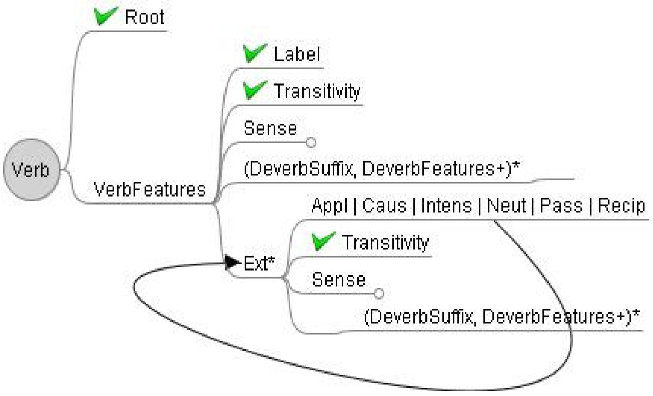
Verb
Root
VerbFeatures
Label
Transitivity
Sense
(DeverbSuffix, DeverbFeatures+)*
Ext*
Appl | Caus | Intens | Neut | Pass | Recip
Transitivity
Sense
(DeverbSuffix, DeverbFeatures+)*

Figure 3a: Verb structure fragment of the data model

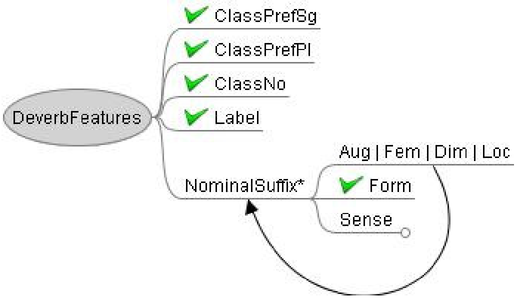

Figure 3b: Deverbative features fragment

The relevant fragment of the data model in the form of a DTD is as follows:

```
<!ELEMENT Entry (Head,Body)>
<!ELEMENT Head (Stem)>
<!ELEMENT Body (PhonTransc*, Tone*, MSI,
Sense+, Dialects*, Etymology?)>
...
<!ELEMENT MSI (POS)>
<!ELEMENT POS (Verb | Noun | Adverb |
Adjective | Ideophone | Enumerative | Pronoun
| AuxVerb)>
...
<!ELEMENT Verb (Root,VerbFeatures)>
<!ELEMENT VerbFeatures (Label, Transitivity,
Sense,(DeverbSuffix,DeverbFeatures+)*,Ext*
)>
<!ELEMENT Ext ((Appl | Caus | Intens | Neut
| Pass | Recip), Transitivity,Sense,
(DeverbSuffix,DeverbFeatures+)*,Ext*)>
...
<!ELEMENT DeverbFeatures  (ClassPrefSg?,
ClassPrefPl?, ClassNo, Label,
NominalSuffix*)>
<!ELEMENT NominalSuffix ((Aug | Fem | Dim |
Loc),Form, Sense, NominalSuffix*)>
...
```

We discuss and exemplify the modelling of two sources of recurrence, viz. verbal extensions and deverbatives.

3.1 Verbal extensions

In Zulu morphology, the basic meaning of a verb root may be modified by suffixing so-called extensions to the verb root. Examples of such extensions are the causative, reciprocal, passive, applied and neuter. We use the verb root *-bon-* 'see' to illustrate the sequencing of these suffixes as well as the associated modification of the basic meaning of the verb root.

(1) *-bon-is-* 'show, cause to see'
-verb.root-caus
(2) *-bon-an-* 'see each other'
-verb.root-recip
(3) *-bon-w-* 'be seen'
-verb.root-pass
(4) *-bon-el-* 'see for'

-verb.root-appl
(5) *-bon-akal-* 'be visible'
-verb.root-neut
(6) *-bon-is-an-* 'show each other'
-verb.root-caus-recip

Verbal extension suffixes modify the basic meaning of the verb root as illustrated in (1) to (6). In certain instances, as illustrated in (6), more than one extension may even be suffixed to the verb root. This structure is modelled in Figure 3 in the `Ext` substructure where recursion is denoted by the arc.

The sequencing of extensions is largely idiosyncratic and attested sequences need to be obtained from corpora and then stored in the MR lexicon. For instance, in a sequence of two or more extensions, the passive is usually last in the sequence. However, in the case of certain verb roots, the reciprocal extension follows the passive, while both sequences are possible in other cases, for example:
(7) *-bon-an-w-* 'seen by each other'
-verb.root-recip-pass
(8) *-bon-w-an-* 'seen by each other'
-verb.root-pass-recip

There are also instances where the applied extension follows the passive (cf. Van Eeden, 1956: 657):
(9) *ya-bulal-w-el-a* > *yabulawela* 'he was killed for'
subj.conc-verb.root-pass-appl-suffix

Similarly, Doke and Vilakazi (1964) list examples (10) and (11) which exemplify the causative and applied extensions in converse sequences, each expressing a slightly different meaning:
(10) *-bon-is-el-* 'look after for'
-verb.root-caus-appl
(11) *-bon-el-is-* 'cause to take care of'
-verb.root-appl-caus-

Existing paper dictionaries of Zulu do not contain exhaustive information on the combinations and sequences of extensions with verb roots. This type of information could, however, be extracted semi-automatically from language corpus resources.

Further examples of verbal extension sequences and how the meaning of the basic root is changed, are given in (12) to (16). This is modelled in Figure 3 by the `Sense` substructure associated with each `Ext`.
(12) *-bon-akal-* "be visible"
-verb.root-neut
(13) *-bon-akal-is-* "make visible"
-verb.root-neut-caus
(14) *-bon-el-an-* "see for/perceive for/take care of each other"
-verb.root-appl
(15) *-bon-el-el-* "treat with consideration"
-verb.root-appl-appl
(16) *-bon-el-el-w-* "be treated with consideration"
-verb.root-appl-appl-pass

For a detailed exposition of Zulu grammar and linguistic terminology, cf. Poulos and Msimang (1998).

3.2 Deverbatives

Deverbatives are formed when a verb (or extended verb) root is suffixed with a deverbative suffix (*o*-, *i*-, *a*-, *e*- and *u*-)[1] and takes a noun class prefix before the verb root. Such derivational affixes cannot combine randomly with any verb root, since they are restricted by semantic considerations.

The data model provides for the capturing of class information, the deverbative suffix, the sequence (recursion) of nominal suffixes, and associated semantic information. This structure is modelled in Figure 3 in the `Nominal Suffix` substructure where recursion is again denoted by the arc. The example in (21) represents the extended root *-bon-is-el-o* with two verbal extensions *-is-* and *-el-* as well as a nominal suffix *-ana*.

Deverbative nouns cannot arbitrarily be formed from any verb root (Van Eeden, 1956:712). At present the resolution of this issue, that is the valid combinations of noun prefixes and (extended) verb roots in the formation of deverbative nouns, is mainly determined from entries in existing dictionaries (attested forms) and occurrences of such combinations in corpora (as yet unlisted forms). The following are examples, based on the root *-bon-* "see":
(17) *isi-bon-i* "mourner (lit. one who looks on at a funeral)"
Class.pref.cl 6/7-verb.root-deverb.suffix
(18) *um-bon-i* "one who sees"
Class.pref.cl 1/2-verb.root-deverb.suffix
(19) *um-bon-o* "apparition, vision"
Class.pref.cl 3/4-verb.root-deverb.suffix
(20) *isi-bon-is-o* "signpost, signal"
Class.pref.cl 6/7-verb.root-caus-deverb.suffix
(21) *isi-bon-is-el-o-ana* "small example"
Class.pref.cl 6/7-verb.root-caus-appl-deverb.suffix-dim

4. Towards implementation

The capturing of data in a rigorous, systematic and appropriately structured way is of utmost importance to ensure that data exchange, in this case between the machine-readable lexical database (MR lexicon) as the source and other applications (for example the morphological analyser), is consistent.

Important considerations in the choice of database implementation include the *type* of data that has to be stored, the different *views* of and *access* to the data that may be required by applications, and the *amount* of data that needs to be stored.

Type: Lexical data is semi-structured (Manning and Parton, 2001) and a chosen database environment for the

[1] *o*- and *i*- are used productively for the formation of impersonal and personal deverbatives respectively, while *a*-, *e*- and *u*- seldom occur and are non-productive.

capturing of lexical information should allow for recursion. In particular, it should allow for multiple occurrences (recursion) of certain substructures such as `Ext` and `NominalSuffix`, resulting in n-depth structures that are reused throughout the representation of such information.

Views and access: For the purposes of rebuilding the morphological analyser (see Figure 1) the preferred view would be the morphological structure of all the word roots in the database while access would be sequential. It should be clear that for other applications different views may be required and access may even be random.

Amount of data: It is estimated that the Zulu MR lexicon should make provision for between 40 000 and 50 000 entries, each of which would have its own specific associated lexical information (see Figure 2).

XML and Unicode are *de facto* standards for mark-up and encoding. Therefore, only Native XML and XML-enabled databases are considered.

An XML-enabled database is a database with extensions for transferring data between XML documents and its own data structures. XML-enabled databases that are tuned for data storage, such as a relational or object-oriented database, are normally used for highly structured data.

A native XML database is one that treats XML documents and elements as the fundamental structures rather than tables, records, and fields (Harold, 2005). Native XML is more suitable for unstructured data or data with irregular structure and mixed content.

Since lexical data are *semi-structured* there are *two* choices: Either endeavour to fit the data into a well-structured database, or store it in a native XML database, designed to handle semi-structured data (Bourret, 2010). In subsequent sections the various options are briefly discussed.

4.1 Native XML databases

Native XML databases (NXD) allow for user-defined schemas. For Zulu such a schema will be based on the proposed data model (given in DTD notation) (Bosch et al., 2007) and its extension (Section 3). This approach to lexical database development was, for instance, also successfully used for Warlpiri, an Indigenous Australian language (Manning and Parton, 2001). Query languages such as XQuery (a W3C standard) and XUpdate, particularly suitable for database-oriented XML, may then be used to query and update the database.

A fragment of the XML code for example (21), according to the DTD in section 3, is as follows:

```
<Verb>
  <Root>bon</Root>
  <VerbFeatures>
    <Label>v</Label>
    <Transitivity>t</Transitivity>
    <Sense>see</Sense>
    <Ext> <!-- Orth. form: bonisa -->
      <Caus>is</Caus>
      <Transitivity>t</Transitivity>
      <Sense>show; cause to see</Sense>
      <Ext> <!-- Orth.form:bonisela -->
        <Appl>el</Appl
        <Transitivity>t</Transitivity>
        <Sense>look after for
        </Sense>
        <DeverbSuffix>o</DeverbSuffix>
        <DeverbFeatures>
          <ClassPrefSg>
            isi
          </ClassPrefSg>
          <ClassPrefPl>
            izi
          </ClassPrefPl>
          <ClassNo>6-7</ClassNo>
          <Label>n:dev</Label>
          <NominalSuffix>
            <Dim>
              <Form>ana</From>
              <Sense>small example</Sense>
            </Dim>
          </NominalSuffix>
        </DeverbFeatures>
      </Ext>
    <Ext>
  </VerbFeatures>
</Verb>
...
```

Using XQuery to return the English translation for *–bon-* in example (21):

```
for $x in doc("verb.xml")/verb
where $x/Root="-bon-"
return $x/VerbFeatures/Sense
```

Result:

```
<?xml version="1.0" encoding="UTF-8"?>
<Sense> see </Sense>
```

4.2 XML-enabled databases

Alternatives to NXDs are XML-enabled databases such as relational and object-orientated databases.

4.2.1 Relational databases

Relational databases are a *de facto* standard for operational and analytical applications (Lin 2008; Naser et al., 2007). Standard query languages (such as SQL) can be used to query the relational database and both commercial and open source software is available for parsing of XML. However, representing the recursive structure of the verbal extensions and the nominal suffixes in a relational database is problematic since the traversal of n-depth structures results in a large number of small tables with "artificially" generated pointers. Arenas and Libkin (2008) investigated the theoretical issues of data exchange in XML documents and indicated that it is not always a clear-cut situation to assume that a set of data in a database shall always produce consistent data, due to the restrictions that exist in a relational database.

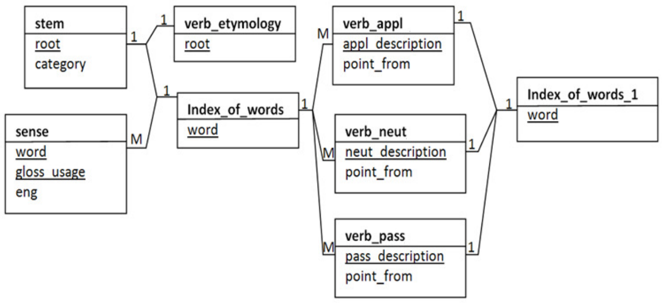

Figure 4: Entity-relationship diagram for Ext-structure

Figure 4 shows a fragment of the entity-relationship diagram (ERD) that was created to represent the Ext-structure in a relational database. Each individual representation is captured in the Index_of_words table and then this instance is represented in the relevant table, e.g. verb_appl or verb_neut. The field point_from will be used as a "pointer" that refers back to the previous element from where the new element is derived. The sense of each new element is captured in the sense table. The ERD is a possible solution for Ext, but rendering DTD compliant XML from this structure and producing consistent data may be time consuming since each individual table needs to be converted to XML and then merged to generate the final XML to be used for archiving and inter-system usage.

4.2.2 Object-oriented databases

An object oriented database supports recursion in the modelling of the verb structure and deverbatives in Zulu. An exposition of the details and full complexity of this approach for the implementation of the data model falls outside the scope of this article. We show only the basic idea by applying it to the verb root and its verbal extensions as follows:

The class VRoot occupies the highest level in the class hierarchy for extended verb roots, followed by six subclasses, viz. VRoot+Appl, VRoot+Caus, VRoot+Intens, VRoot+Neut, VRoot+Pass and VRoot+Recip. In turn, each subclass again has its own six subclasses, viz. VRoot+Appl+Appl, VRoot+Appl+Caus, …, and so on. This hierarchy is dynamically expanded as new extended roots are identified. Unlike an entity in a relational database, an object includes not only information about the relationships between facts within an object, but also information about its relationship with other objects.

A particular verb root (say *-bon-*) and its extensions are then individual objects (instantiations) in the appropriate classes. Each such instantiation has its own unique identity (OID) and inherits specific attributes and methods from its super class/parent object, making it quite intuitive to present the lexical data in an object-oriented database.

In example (6) more than one extension is suffixed to the verb root. The classes of interest in this example are VRoot, VRoot+Caus, and VRoot+Caus+Recip and the instantiated objects are for *-bon-*, *-bon-is-* and *-bon-is-an-*, as shown in Figure 5, where the arrow indicates inheritance of information via the class hierarchy. An example of inherited information is the Etymology of the basic root, while lexical information that is associated with each unique object is the Sense information. Figure 5 illustrates how such recursion is captured in an object-oriented approach.

Naser et al. (2007) describe the "forward engineering" process whereby the object oriented database environment (including inheritance and nesting) is used as input and a corresponding XML schema and document(s) are produced as output. The "two-way mapping" algorithm (Naser et al., 2009) may be used to transform the data in the object-oriented approach, firstly to flat XML, then to the nested XML schema and lastly to the final XML document.

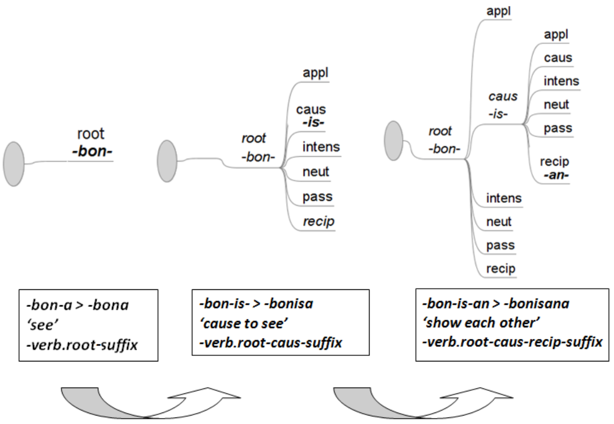

Figure 5: Recursion as objects

5. Conclusion and future work

In this article we described the refinement of a data model for a Zulu MR lexicon, focusing on the modelling of two sources of recurrence in the verb structure, viz. verbal extensions and deverbatives. We also showed how the MR lexicon, based on the proposed data model, may be embedded in an update framework. Finally we proposed Native XML and object-orientated databases as two possible approaches to implementation. Reasons for this may be summarized as follows:

Type of data: Object oriented databases are designed to work well with state of the art object oriented programming languages. Object oriented databases use the same model as these programming languages as they store and index theoretical objects. "Object databases are generally recommended when there is a business need for high performance processing on complex data" (Foster, 2010).

XML Databases offer the same functionality as Object oriented databases. The data is structured in a hierarchical manner except that Native XML databases store XML documents instead of theoretical objects. While this is conceptually the same data storage, XML databases have the added benefit of being able to exchange the data in its native format – no overheads are incurred by transformation and mapping of the data to and from other database structures or representation.

Views and access: Where Object Databases have Object Query Language (OQL), XML Databases have XQuery, which is a W3C standard. Multiple views of the data, as well as appropriate access, are supported via these powerful query languages.

Amount of data: Both native XML and object oriented databases are able to process arbitrarily large documents. Moreover, these approaches are claimed to offer high performance since queries over a well-designed, well-implemented native XML or XML-enabled object oriented database are faster than queries over documents stored in a file system – particularly in cases where queries are significantly more frequent than insertions and updates, which is expected to be the case for a comprehensive, mature and stable MR lexicon.

Work planned for the near future includes

- the development of prototypes for the MR lexicon for Zulu in order to investigate, demonstrate, evaluate and compare the two possible approaches to implementation;
- the development of software support for semi-automating the lexicon update framework;
- the bootstrapping of MR lexicons for various other Bantu languages from the Zulu lexicon.

6. Acknowledgements

This material is based upon work supported by the National Research Foundation under grant number 2053403. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Research Foundation.

7. References

Arenas, M., Libkin, L. (2008). XML data exchange: consistency and query answering, *Journal of the ACM*, 55(2). [Online]. Available: http://portal.acm.org/citation.cfm?id=1346332 (accessed February 2010).

Bosch, S.E., Pretorius, L. & Jones, J. (2007). Towards machine-readable lexicons for South African Bantu languages. *Nordic Journal of African Studies* 16(2), pp.131–145.

Bosch, S., Pretorius, L. & Fleisch, A. (2008). Experimental Bootstrapping of Morphological Analysers for Nguni Languages. *Nordic Journal of African Studies* 17(2), pp. 66–88.

Bourret, R. (2009). XML Database Products:

XML-Enabled Databases. [Online]. Available: http://www.rpbourret.com/xml/ProdsXMLEnabled.htm (accessed March 2010).

Doke, C.M. & Vilakazi, B. (1964). *Zulu-English Dictionary*. Johannesburg: Witwatersrand University Press.

Foster, C. (2010). XML databases – The business case. *CFoster.net*. [Online]. Available: http://www.cfoster.net/articles/xmldb-business-case/ (accessed March 2010).

Harold, E.R. (2005). Managing XML data: Native XML databases - Theory and reality. IBM developerWorks. [Online]. Available: http://www.ibm.com/developerworks/xml/library/x-mxd4.html (accessed March 2010).

Lin, C.Y. (2008). Migrating to relational systems: Problems, methods, and strategies. *Contemporary Management Research*, 4(4), pp.369–380.

Manning, C. & Parton, K. (2001). What's needed for lexical databases? Experiences with Kirrkirr. *Proceedings of the IRCS Workshop on Linguistic Databases*, pp. 167-173. University of Pennsylvania, Philadelphia.

Naser, T., Alhajj, R. & Ridley, M.J. (2009). Two-way mapping between object-oriented databases and XML. *Special Issue: Information Reuse and Integration*. R. Alhajj (Ed.), 33, pp.297–308.

Naser, T., Kianmehr, K., Alhajjb, R. & Ridley, M.J. (2007). Transforming object-oriented databases into XML. *Proceedings of the IEEE International Conference on Information Reuse and Integration*, IRI 2007. pp.600–605.

Poulos, G. & Msimang, C.T. (1998). *A Linguistic Analysis of Zulu*. Pretoria: Via Afrika.

Pretorius, L. & Bosch, SE. (2003). Finite-state computational morphology: An analyzer prototype for Zulu. *Machine Translation – Special issue on finite-state language resources and language processing*, 18, pp.195–216.

Van Eeden, B.I.C. (1956). *Zoeloe-Grammatika*. Stellenbosch: Universiteits-uitgewers.

Weber, D.J. (2002). Reflections on the Huallaga Quechua dictionary: derived forms as subentries. [Online]. Available: http://emeld.org/workshop/2002/presentations/weber/emeld.pdf (accessed February 2010).