Creation and Evaluation of Extensible Language Resources for Maltese

Angelo Dalli

Maltilex Project
Department of Computer Science & AI
University of Malta
Msida MSD 02, Malta
adall@cs.um.edu.mt

Abstract

The creation of Language Resources is a labour intensive process whose difficulty is further compounded when minority languages are concerned (Cunningham, 1999). This paper discusses the creation of an extensible set of Language Resources for Maltese developed by the Maltilex Project at the University of Malta (Rosner et. al., 1999), together with quality evaluation mechanisms for minority languages.

1. Introduction

Maltese is the native official language of Malta, spoken by most of Malta's approximately 370,000 inhabitants and by a significant number of people living in Maltese communities in Australia, UK, USA and Canada. Maltese has evolved independently of Arabic since the thirteenth century and has had significant influences from Sicilian and Italian together with recent additions from English. According to Mifsud, "Maltese is a mixed language with a Semitic (in particular Arabic) substratum, a Romance superstratum and an English adstratum" (Mifsud, 1995). Due to the relatively small number of speakers Maltese is classified as a minor language, although it is not presently endangered.

Maltese was also influenced greatly from the Siculo-Arabic language spoken in Sicily during the Middle Ages. Maltese is usually considered to be a variety of North-African vernacular Arabic with an independent Latin-based orthography. These features make Maltese a quasi-independent language bridging the two different cultures of North Africa and Southern Europe, reflecting Malta's geographical position and sociological history.

The Southern European and English influence are important for the development of the computational lexicon. While it is widely accepted that Maltese has an essentially Arabic system of morphology and syntax (Aquilina, 1973), "the influence of Sicilian and Standard Italian has been primarily on the lexicon (including phraseology) and has led to certain changes in morphology (in the forms themselves, not in the range of categories distinguished). As far as English is concerned, Maltese has borrowed a number of lexical items from English, though these are not in general integrated into the overall system of the language, as the Siculo-Italian loans are." (Comrie, 1987).

This essentially means that techniques developed for the identification and treatment of purely Arabic words can still be applied to Maltese, but any computational lexicon would need the additional capability to handle Romance and English words. This aspect is considered to be an advantage since techniques that do not over-specialise by relying excessively on the Arabic morphology and syntax of Maltese can be readily applied to other languages with minor modifications. On the other hand, techniques that have been developed for the processing of English and Romance languages will not work efficiently or not at all on Maltese.

1.1. Maltilex Project

Maltese did not have any form of large-scale computerised Language Resources prior to the initiation of the Maltilex Project. This unfortunate fact was turned into an advantage since it permitted a modern approach that conforms to and builds upon the existing guidelines, standards and best practices developed by other projects and international programmes like TEI, MULTEXT, XCES-EAGLES, ISLE, and PAROLE-SIMPLE (Zampolli, 2000; EAGLES, 2000; Bertagna, 2000).

Initially a full-scale computational lexicon for Maltese was created. Due to the limited amount of resources available for the lexicon creation, we have looked at different intelligent means of reducing the workload on the linguist by shifting more work onto intelligent automated systems that can perform the bulk of the manual work needed to create Language Resources for minor languages from scratch.

Most Language Resources that are currently available for research and development can be currently classified as a heterogeneous collection of different proprietary formats and databases with minimal means, if any, of interoperability with other Language Resources making it hard to extend their usefulness beyond the life of their originating projects (Cunningham, 1999). This is an even more serious issue for minority languages, since fewer people will be willing to utilise Language Resources of these languages if there is no commonly accessible metadata description of these Language Resources that enables established tools to be used in an interoperable manner.

Certain existing initiatives such as the Open Archives Initiative (OAI) and GATE already solve many of the problems that arise in ensuring interoperability and metadata descriptions of services and content (OAI, 2001; Wilks et. al., 1998). These two solutions still have two main disadvantages in using proprietary data formats and protocols, although a conversion layer that will make both systems interoperate with other implementations is easy to create. Both OAI and GATE are quite suitable for the implementation of different Language Resources. At the Maltilex Project we have developed a weakly supervised machine learning technique, called the Lexicon

Structuring Technique (LST), based on Bioinformatics and Data Mining principles that enabled us to largely automate the initial creation of the computational lexicon for Maltese (Dalli, 2002).

From an initial corpus of around 3,000 different Maltese texts containing over 2 million different word forms, a lexicon consisting of around 80,000 different lemmas (headwords) was created. Using the learning techniques described in (Dalli, 2002) a lexicon of around 60,000 unique word forms was obtained. The lexicon serves as the cornerstone for all other Language Resources that are being developed at the Maltilex Project. Due to the radical difference in the approach we had to create a relational database system to store our Language Resources from scratch to accommodate LST.

Relational database technology is used to create a core set of tables that define a core computational lexicon that contains basic orthographic and phonological information on the words in the lexicon. LST automatically groups word forms under one or more lemmas automatically. Every lemma gets assigned a headword using an exemplar taken from the word forms grouped under the lemma. The main advantage of using lemmas rather than individual word forms themselves as the basic unit of reference is enhanced flexibility. Every lemma can be assigned different semantical relationships and can optionally store some individual word forms explicitly and generate the rest of the word forms that conform to some known rules.

The core table fields can be mapped almost directly to the XCES data representation standard defined by the EAGLES-ISLES projects (EAGLES, 2000; Bertagna, 2000).

The core can be extended indefinitely through the use of a special extension API that automatically registers new functions and creates new additions to the underlying linguistic database. This allows virtually any kind of applications, data and services to be added to the linguistic core.

1.2. Web Based Interoperable Language Resources

The current trend of using web services to integrate different information repositories and services across the Internet led us to consider a more flexible and open standard for Interoperable Language Resources based on industry standard web services protocols and mechanisms. Interoperability is achieved through two means - flexible XML export methods and a Simple Object Access Protocol (SOAP) based server (Box et. al., 2000).

XML is the natural choice for storing and representing linguistic data due to its simplicity and compatibility with a variety of existing systems. One of XML's main drawbacks is that pure XML databases are usually limited in their performance. Relation database records are thus used to store linguistic data efficiently. The database information is then converted transparently to XML format using transformation tables. Standard SQL selection queries can be used to filter the export data efficiently prior to conversion to XML. Alternatives like XSL can transform XCES data to other XML formats as needed, but generally result in great performance penalties for huge amounts of data. An XCES compliant tagset for Maltese was developed by the Maltilex Project for this purpose (Gatt, 2001).

Language Resources can be encapsulated using a SOAP server that provides a transparent, web-accessible layer to existing implementations (Box et. al., 2000). In addition, SOAP easily supports the Web Services Description Language (WSDL) that is used to provide an XML-based metadata description of the services offered by the SOAP server automatically (Christensen et. al., 2001).

The SOAP server provides XML-based interactions between different Language Resources and related applications over the HTTP protocol. Language Resources can be imported and exported in XML format and converted into efficient relational records transparently. Server-side processing can also be utilised effectively to reduce the load on the client.

Figure 1 illustrates the encapsulation of a Language Resource (that might be either a data repository or some other kind of processing application or service) by a SOAP server. A conversion layer provides the necessary transformations needed to hide the Language Resource's actual implementation details by converting the proprietary implementation formats to a standard core format expected by the SOAP server. The conversion layer also performs this transformation in reverse, to facilitate updates of the Language Resource by other linguistic applications or projects.

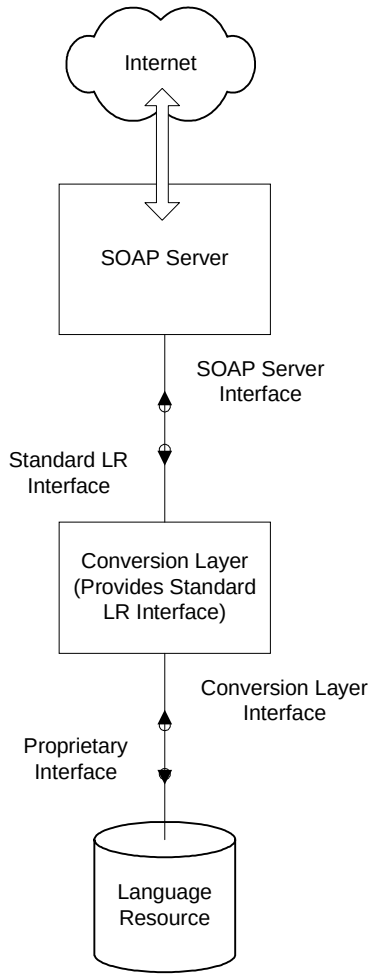

Figure 1 SOAP Server Encapsulation of a Language Resource

SOAP offers many advantages over other web-based protocols since it is slated to be one of the fundamental mechanisms to enable web services over the Internet. Most major programming languages and environments already support SOAP directly, so implementation issues should be minimal.

1.3. Web Services Model

Different Language Resources will obviously have different data entry and processing needs that force different layouts and database schemas to be used to store linguistical information for different languages.

Language Resources are usually accessed through some published API for a particular programming language. If more than one language needs to be supported, different versions of the same API have to be created. Due to development and target language constraints, the different versions of the same API may not cover exactly the same functionality. It is thus desirable to separate the actual API from the implementation language so that one definition is enough for all implementation languages. Also, small projects may not have enough resources to deal with more than one major programming language, making its content difficult to access.

This problem can be solved by adopting a Web Services Model where different data and processing components of Language Resources are modelled by services that are accessed in a language-transparent manner. Every service offers a set of associated algorithms and functions that are applied to relevant linguistic data. The granularity of every service mainly depends on a choice of implementation style. However it is desirable to keep the number of services down to a manageable value. A good heuristic is to model major components or object packages in the API as a service while representing minor objects and object methods associated with the component as service functions (Dalli, 2001).

WSDL provides a standard means of creating Internet-accessible metadata descriptions of the Language Resource being abstractly represented by the SOAP server.

WSDL is used to describe the services and features provided by the Language Resource in a standard manner, significantly reducing the development time for new applications and related information extraction and analysis programs. Additionally, client applications using WSDL are shielded from the server implementation, greatly simplifying maintenance and upgrade of existing facilities.

1.3.1. Sample WSDL Description

WSDL provides a comprehensive means of describing the mechanisms that should be used to access and process content pertaining to a specific Language Resource. A set of abstract operations – that can either return unprocessed or processed information – are bound to some network protocol and finally assigned to some physical address to create a WSDL port. A series of WSDL ports are then packaged together to form a web service.

For example, in the Maltilex Project, we used the function getEtymology to return an array of short language code that determines the etymological source of a particular word. A summarized version of the WSDL definition for getEtymology that includes additional port and service definition parts is presented below:

```
<?xml version="1.0"?>
<definitions name="LRCore"

 targetNamespace=
  "http://mlex.cs.um.edu.mt/IELD/
  LRCore.wsdl"
 xmlns:soap="http://schemas.
  xmlsoap.org/wsdl/soap/"
 xmlns:tns="http://mlex.cs.um.edu.
  mt/IELD/LRCore.wsdl"
 xmlns="http://schemas.xmlsoap.org/
  wsdl/">

       <service name="LRCoreService">
    <documentation>Maltese Computational
Lexicon Core Service Port
        </documentation>
 <port name="LRCorePort"
       binding="tns:LRCoreBinding">
       <soap:address
location= "http://mlex.cs.um.edu.mt/
IELD/LRCore" />
        </port>
       </service>

       <binding name="LRCoreBinding"
    type="tns:LRCorePortType">
        <soap:binding style="document"
     transport="http://schemas.
      xmlsoap.org/soap/http" />

        <operation name="getEtymology">
         <soap:operation soapAction=
      "urn: getEtymology" />

          <input>
               <soap:body use="encoded"
                namespace=
                "http://soapinterop.org"
               encodingStyle="http://schemas.
      xmlsoap.org/soap/encoding/" />
          </input>

  <output>
               <soap:body use="encoded"
                namespace=
                "http://soapinterop.org"
               encodingStyle="http://schemas.
               xmlsoap.org/soap/encoding/" />
    </output>

        </operation>
       </binding>

       <message
    name="getEtymologyInput">
        <part name="headword"
type="xsd:string" />
       </message>

       <message
    name="getEtymologyOutput">
        <part name="return"
          type="tns:ArrayOfstring" />
```

```
      </message>

      <portType
    name="LRCorePortType">
       <operation name="getEtymology"
         parameterOrder="headword">
        <input message=
     "tns:getEtymologyInput" />
        <output message=
     "tns:getEtymologyOutput" />
       </operation>
      </portType>

</definitions>
```

The function getEtymology is thus defined as a function contained in the LRCorePort port that forms part of the LRCoreService web service. The function's input and output definition states that the input takes a single string parameter (headword) and returns an array of strings as output.

Any extensions to the actual service implementations can be simply registered and accessed by changing the WSDL file. The different service ports can also be implemented on a distributed system of servers by changing the URLs in the port binding definition.

1.4. Universal Description, Discovery, and Integration

Recent developments like the Universal Description, Discovery and Integration (UDDI) International Registry facilitate the development of flexible and secure but easily accessible Language Resources (Ariba et. al., 2000). UDDI essentially is a machine accessible directory of different web services provided around the world.

The WSDL description itself is an XML document, enabling consistent and interoperable exchange of Language Resource metadata descriptions to take place. A UDDI repository of WSDL descriptions of different Language Resources around the world will enable the creation of an International Language Resource Directory that aids in the dissemination of metadata descriptions across different research projects. Wide industrial support for WSDL metadata descriptions and UDDI facilitate development and standardization processes while ensuring stability and commitment on the part of large number of business and institutions world wide (Curbera et. al., 2001).

Figure 2 illustrates the relationship between WSDL and UDDI. WSDL is used to describe the Language Resources while UDDI is used to create a globally accessible registration for these services. Essentially UDDI provides an easy means for Language Resources to be discovered and utilised automatically by research projects around the world. The low learning curve and minimal costs involved in integrating these technologies into existing projects makes the proposed system highly attractive for small and medium sized projects that have limited available resources.

2. Evaluation

An important aspect in the creation of any new Language Resource is the validation and quality assurance processes that need to be undertaken continuously to ensure a high quality end result.

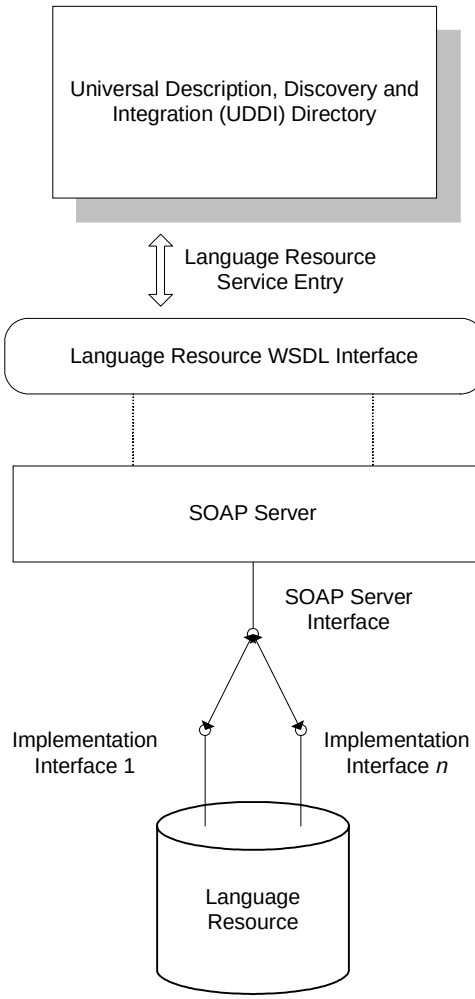

Figure 2 Language Resource Universal Description and Discovery Mechanism using UDDI, WSDL and SOAP

Two statistical quality and evaluation measures were developed at the Maltilex Project. Both measures are able to use a statistical sample that is evaluated against traditional printed or written information instead of existing computerized information. As in the case of Maltese, the ability to use evaluation measures to validate and compare the quality of Computational Language Resources against traditional printed linguistical information is very important to minority languages since these often have no prior computational Linguistic Resources available.

A statistical lexicon quality measure based on the cluster-based F-measure uses the information theoretic concepts of precision and recall together with statistical sampling techniques of existing non-computational data to provide an external measure of lexicon quality (Steinbach et. al., 1999).

An evaluation measure that gives a quantitative value for the estimated language coverage is also presented. The evaluation measure is used to gauge the progress and maturity of the LR creation project on a continual basis.

2.1. Quality Evaluation

In the Maltilex Project the development of the computational lexicon is based on the notion of having a

set of lemmas which consist of clusters of related word forms in their full form, with every lemma being classified under some unique headword. This definition is highly related to clustering systems (with lemmas being analogous to clusters), so a brief overview of existing cluster quality evaluation measures is presented. There are two main ways of evaluating the resulting cluster quality which are summarised in (Steinbach et al., 1999) as follows:

- Internal Quality Measure – Clusters are compared without reference to external knowledge against some predefined set of desirable qualities.
- External Quality Measure – Clusters are compared to known external classes.

Internal quality measures are generally either subjective or else not applicable to most linguistical work since the existence of such a quality measure would mean that better results can be produced by applying this quality measure in conjunction with some optimisation algorithm.

The two main external quality measures are entropy based measures (Shannon, 1948) and the F-measure (Rijsbergen, 1979; Larsen and Aone, 1999).

Entropy based quality measures assert that the best entropy that can be obtained is when each cluster contains exactly one member. The class distribution of the data is calculated by considering the probability of every member belonging to some class. The entropy of every cluster *j* is calculated using the standard entropy formula:

$$E(j) = -\sum_i p_{ij} \log(p_{ij})$$

where p_{ij} denotes the probability that a member of cluster *j* belongs to class *i*. The total entropy E^* is then calculated as:

$$E^* = \frac{1}{n} \sum_{j=1}^{m} n_j \cdot E(j)$$

where n_j is the size of cluster *j*, *m* the number of clusters, and *n* the total number of data points.

The F-measure treats every cluster as a query and every class as the desired result set for a query. The recall and precision values for each given class are then calculated using information retrieval concepts. The F-measure of cluster *j* and class *i* is given by:

$$F(i, j) = \frac{2 \cdot r(i, j) \cdot p(i, j)}{r(i, j) + p(i, j)}$$

where *r* denotes recall and *p* is the precision. Recall *r* and precision *p* are defined as:

$$r(i, j) = \frac{n_{ij}}{n_i} \qquad p(i, j) = \frac{n_{ij}}{n_j}$$

respectively, where nij is the number of class i members in cluster j, while nj and ni are the sizes of cluster j and class i respectively. The overall F-measure for the entire data set of size n is given by:

$$F^* = \sum_i \frac{n_i}{n} \max[F(i, j)]$$

2.2. Lexicon Quality Measure: L-Measure

Computational lexicons have an additional domain-specific external quality measure available in the form of existing non-computational language dictionaries and written resources.

Dictionaries can be used to compare the results generated by LST or the results inputted into some computerized system against possibly non-computational or written resources that might be the only source of data available in the language. Since every cluster and class correspond to a lemma the number of classes to be considered is expected to number in the thousands for any language of significant size. Furthermore most non-computational Language Resources are not amenable to automated analysis techniques. Thus a modified statistical sampling technique based on the F-measure called the L-measure has been devised to overcome these difficulties.

The L-measure attempts to measure the quality of a given lexicon in relation to other existing lexicons that are possibly non-computational lexicons (i.e. human compiled language dictionaries or word lists), taking into consideration that a full population analysis may not be practical under most circumstances.

2.2.1. L-Measure Definition

Given a lexicon *L* and a set of dictionaries $D = \{D_1 .. D_k\}$ obtain two full form canonical wordlists *W* and *W'* from *L* and *D* respectively. Define *Y* to be the wordlist of words common to both *W* and *W'*, $Y = W \cap W'$. The sample size *S* is defined as $\alpha.|lemmas(Y)|$ where α is some value in the range (0..1) that controls the random sample size and *lemmas* gives a set containing all unique lemma headwords in a given wordlist. Typically α should be set to somewhere between 0.01 and 0.1. For computational lexicons an exact value for the size of *lemmas* can be easily obtained. For non-computational lexicons a unique headword count or estimate will provide a reasonably correct estimate for the size of *lemmas*. It is expected that the sample size will be large enough to assume that the sample is representative of the whole population.

The L-measure of a lemma *j* in *lemmas*(*W*) and lemma *i* in *lemmas*(*Y*) is given by:

$$L(i, j) = \frac{2 \cdot r(i, j) \cdot p(i, j)}{r(i, j) + p(i, j)}$$

where *r* denotes recall and *p* is the precision as defined for the F-measure but where n_{ij} is the number of lemma *i* members in lemma *j*, while n_j and n_i are the sizes of lemma *j* and lemma *i* respectively. The overall L-measure L^* for the entire sample of size *n* is given by:

$$L^* = \sum_i \frac{n_i}{n} \max[L(i, j)]$$

L^* will be in the range [0..1] and is proportional to the lexicon quality. *Y* is used instead of *W'* since lexical word coverage is largely determined by the quality of the corpus used to create the lexicon. While this kind of analysis might be useful in determining the coverage of a lexicon the L-measure is oriented towards measuring quality rather than quantity, independently of the corpus that was used to create the lexicon.

2.3. Lexicon Coverage Measure: C-Measure

A simple count of the number of word forms in *W* and *W'* (as defined for the L-measure) that are common to both and those that are specific to either *W* or *W'* should be

enough to determine the amount of coverage provided by the lexicon.

2.4. C-Measure Definition

Given a lexicon L and a set of dictionaries $D = \{D_1 .. D_k\}$ obtain a full form canonical wordlist W' from D. The sample size S is defined as $\alpha.|lemmas(Y)|$ where α and lemmas are defined as for the L-Measure. Define W as the canonical wordlist of size S obtained by randomly selecting S word forms from L.

Count all the wordforms generated from the canonical wordlist and classify them according to the following list:

1. The wordform is related to a lemma in $W \cap W'$ and found in both W and W'.
2. The wordform is related to a lemma in $W \cap W'$ and is found in W but not in W'.
3. The wordform is related to a lemma in $W \cap W'$ and is found in W' but not in W.
4. The wordform is not related to a lemma in $W \cap W'$ and is found in W but not in W'.
5. The wordform is not related to a lemma in $W \cap W'$ and is found in W' but not in W.

Denote the total number of word forms according to their type, as listed above, divided by the total number of word forms, as $C_1 .. C_5$. The C-measure is given by:

$$\chi_1 C_1 + \chi_2 (C_2 + C_4) - \chi_3 (C_3 + C_5)$$

where χ_1, χ_2 and χ_3 are weights in the range [0..1]. The confidence in the dictionaries D increases as the sum $C_2 + C_4$ decreases. The recommended values for the weights are thus $\chi_1 = 1.0$ and $\chi_3 = \gamma (1 - (C_2 + C_4))$ and $\chi_2 = \gamma (C_2 + C_4)$, where γ is a value in the range (0..1] that gives a confidence rating for the corpus quality regarding spelling mistakes and related errors that may creep in a lexicon. Assuming a 0.05% error rate, γ would be set to 0.95. The range of values for the C-measure range from a value of 1 for perfect coverage to $-\gamma$ for no coverage at all.

3. Conclusion and Future Work

This paper has presented an outline of a system of interoperable and extensible Language Resources based on a Web Services Model using WSDL and UDDI, supplemented by brief examples from the actual implementation by the Maltilex Project. Various issues concerning the attractiveness of this model to minor languages are also discussed.

Two statistical quality and coverage measures – the L-Measure and C-Measure – are also highly suitable for the evaluation of Language Resources of minor and endangered languages that might not have any prior computerized Language Resources available to act as a gold standard.

The logical extension of this work is the actual adoption of a web services based model to Language Resources and the refinement of the L-Measure and C-Measure for better evaluation of Language Resources where the available data is sparse.

3.1. Acknowledgements

My thanks goes to Mike Rosner for providing me with helpful advice throughout and to the University of Malta for their support.

4. References

Aquilina, Joseph. 1973. The structure of Maltese: a study in mixed grammar and vocabulary. University of Malta.

Ariba Inc. International Business Machines Corporation. Microsoft Corporation. September 2000. UDDI Technical White Paper. http://www.uddi.org.

Bertagna, Francesca. Calzolari, Nicoletta. Lenci, Alessandro. Zampolli, Antonio. 2001. The Multilingual ISLE Lexicon Entry (MILE). ISLE Computational Lexicons Working Group Report, Italy.

Box, Don et. al. 2000. Simple Object Access Protocol 1.1. W3C Note. http://www.w3.org/TR/SOAP.

Christensen, Erik et. al. 2001. Web Services Description Language 1.1. W3C Note. http://www.w3.org/TR/wsdl.

Comrie, Bernard. 1987. The major languages of South Asia, the Middle East and Africa. Routledge.

Cunningham, Hamish. 1999. A Definition and Short History of Language Engineering. Journal of Natural Language Engineering, Cambridge University Press, 5: 1-16.

Curbera, Francisco. Ehnebuske, David. Rogers, Dan. June 2001. Using WSDL in a UDDI Registry. UDDI Working Draft Best Practices Document version 1.05.

Dalli, Angelo. 2001. Interoperable Extensible Linguistic Databases. IRCS Workshop on Linguistics Databases, University of Pennsylvania.

Dalli, Angelo. 2002. Biologically Inspired Lexicon Structuring Technique. Human Language Technology (HLT) 2002, San Diego, California.

Expert Advisory Group on Language Engineering Standards (EAGLES). 2000. Corpus Encoding Standard for XML. Vassar College, New York. Equipe Langue et Dialogue LORIA/CNRS, France.

Gatt, Albert. 2001. An XCES compliant tagset for Maltese. Maltilex Project, University of Malta.

Larsen, B. Aone, C. 1999. Fast and Effective Text Mining using Linear-Time Document Clustering. KD-99, San Diego, California.

Mifsud, Manwel. 1995. Loan verbs in Maltese: a descriptive and comparative study, pg. 25.

Open Archives Initiative (OAI). 2001. The Open Archives Initiative Protocol for Metadata Harvesting. http://www.openarchives.org.

Rijsbergen, C. J. 1979. Information Retrieval. Butterworth, London.

Rosner, Michael et. al. 1999. Linguistic and Computational Aspects of Maltilex. Proc. of the ATLAS Symposium, Tunis.

Shannon, Claude. 1948. The mathematical theory of communication. Bell Systems Technical Journal, 27:379-423, 623-656.

Steinbach, Michael. Karypis, George. Kumar, Vipin. 1999. A comparison of document clustering techniques. Technical Report #00-034, University of Minnesota.

UDDI Consortium. 2001. UDDI Version 2.0 Data Structure Reference. http://www.uddi.org/pubs.

Wilks, Yorick. Gaizauskas, Robert. Cunningham, Hamish. 1998. GATE: General Architecture for Text Engineering. Sheffield University, UK.

Zampolli, Antonio. 2000. Extensions of PAROLE & SIMPLE resources: National Projects. SIMPLE: From Monolingual to Multilingual Resources Workshop, Athens.