Morphology in the Maltese Language
A Computational Perspective

Claudia Borg

Supervisor: Albert Gatt

Co-Supervisor: Ray Fabri

Institute of Linguistics

University of Malta

This dissertation is submitted for the degree of

Doctor of Philosophy

September 2015

UNIVERSITY OF MALTA

~~FACULTY~~/INSTITUTE/~~CENTRE~~/~~SCHOOL~~ of Linguistics

DECLARATION OF AUTHENTICITY FOR DOCTORAL STUDENTS

Student's I.D. /Code 18876 m

Student's Name & Surname Claudia Borg

Course Doctorate of Philosophy

Title of Dissertation/Thesis

Morphology in the Maltese Language:
A Computational Perspective

I hereby declare that I am the legitimate author of this Dissertation/Thesis and that it is my original work.

No portion of this work has been submitted in support of an application for another degree or qualification of this or any other university or institution of higher education.

I hold the University of Malta harmless against any third party claims with regard to copyright violation, breach of confidentiality, defamation and any other third party right infringement.

☑ As a Ph.D. student, as per Regulation 49 of the Doctor of Philosophy Regulations, I accept that my thesis be made publicly available on the University of Malta Institutional Repository.

☑ As a Professional Doctoral student, as per Regulation 54 of the Professional Doctorate Regulations, I accept that my dissertation be made publicly available on the University of Malta Institutional Repository.

☐ As a Doctor of Sacred Theology student, as per Regulation 17 of the Doctor of Sacred Theology Regulations, I accept that my dissertation be made publicly available on the University of Malta Institutional Repository.

☐ As a Doctor of Music student, as per Regulation 24 of the Doctor of Music Regulations, I accept that my dissertation be made publicly available on the University of Malta Institutional Repository.

Signature of Student

15th June 2016
Date

** Tick applicable paragraph.*

11.06.2015

Acknowledgements

Firstly, I would like to thank my supervisors, Albert Gatt and Ray Fabri, for their constant support, inspiration and guidance. I am particularly indebted to Albert for embracing all my proposals and directing them towards fruition. He was a true catalyst for this research. And to Ray, for sharing his immense knowledge of Maltese morphology and all the discussions that ensued. I consider myself lucky to have had both of them guiding me throughout this research.

I am thankful to the colleagues and fellow graduate students at the Institute of Linguistics for their steadfast encouragement. I am particularly thankful to Patrizia Paggio, Lonneke van der Plus, Albert Borg and Nizar Habash for the fruitful discussions and insights to this work. I would also like to thank Michael Spagnol, Luke Galea, Leanne Ellul, Theresa Abdilla, John Paul Grima, John Camilleri, Mike Rosner, Adam Ussishkin and Andy Wedel for their support and interest in my research.

I am grateful to my family and friends for being there when I needed them the most.

The research work disclosed in this thesis was partially funded by the Malta Government Scholarship Scheme (MGSS) grant.

Abstract

This thesis presents the first comprehensive and systematic treatment of Maltese morphology using machine learning techniques. Maltese is considered as a 'mixed' language, reflected in the hybrid nature of the morphological system, which has elements of both templatic systems typical of Semitic languages, and stem-based systems typical of Indo-European ones. The research looked at three different aspects of computational morphology, namely segmentation, relations and labelling.

The segmentation task first explored unsupervised techniques to learn potential stems and affixes. The results were then used as the basis of the relations task, through the clustering of words on the basis of their orthographic and semantic similarity. The clustering technique was also unsupervised and used a metric to measure the disparity or similarity of a group of words so as to improve the clusters. An evaluation of the clusters was carried out using both experts and non-experts. The results of the non-expert group focused on the quality of the clusters, whilst the analysis of the expert responses focused on the differences between the concatenative and non-concatenative word clusters. Morphological labelling of words was viewed as a classification problem and approached using supervised techniques. Initially, the research focused on the classification of verbal inflections, resulting in a sequence of classifiers that represented different morphological properties. Cascade classifiers were then built for the noun and adjective categories, and integrated into a single classification system. The classification of grammatical category was also explored, questioning whether the morphological labels outputted by the different cascades could be used to reinforce the classification of the grammatical category. A final evaluation tested the full classification system on gold standard data from the MLRS corpus.

The research resulted in a morphological classification system for verbs, nouns and adjectives. Although it has not yet achieved a sufficiently high accuracy, it provides the foundations for a more complete morphological analyser with broader coverage. The scope of the research was not merely a technological one, to create a morphological analyser, but rather to investigate the hybridity of the morphological system in Maltese and how this impacts the results of different techniques.

Table of contents

List of figures **x**

List of tables **xii**

1 Introduction **1**

1.1 The Maltese language 1

1.1.1 Main aims and goals of this research 4

1.2 Morphological properties in the Maltese language 5

1.2.1 The verb 6

1.2.2 The noun 7

1.2.3 The adjective 9

1.3 Computational morphology for Maltese:
Challenges and prospects 9

1.4 Outline and contributions of the thesis 11

2 Computational Morphology **14**

2.1 Introduction 14

2.2 Language acquisition 16

2.3 Computational approaches to morphology 21

2.3.1 Rule-based approaches to morphological analysis 21

2.3.2 Data-driven approaches to morphological analysis 23

2.3.2.1 Problem definition 23

2.3.2.2 Morpho Challenge 25

2.3.2.3 Segmentation 27

2.3.2.4 Clustering morphologically related words 36

2.3.2.5 Learning labels for morphemes 40

2.4 Approaches used in Semitic languages 43

2.5 Evaluation . . . 46

2.6 Summary and proposed approach . . . 49

3 Segmenting and Clustering Morphologically Related Words 51

3.1 Introduction . . . 51

3.2 Segmentation . . . 53

3.2.1 Data preparation . . . 53

3.2.2 Extracting affixes . . . 56

3.2.3 Composite suffixes . . . 59

3.2.4 Segmenting words . . . 62

3.3 Clustering technique . . . 63

3.3.1 Segmentation-based clustering . . . 63

3.3.2 Semantic and orthographic similarity . . . 66

3.3.3 Merging clusters . . . 68

3.4 Evaluating clusters . . . 70

3.4.1 Pilot study . . . 71

3.4.2 Participants . . . 72

3.4.3 Materials and design . . . 73

3.4.4 Procedure . . . 73

3.5 Analysis of results . . . 79

3.5.1 Removal of words . . . 79

3.5.2 Quality ratings . . . 83

3.5.3 Inter-Annotator Agreement . . . 84

3.5.4 Hybrid morphology and the clustering technique . . . 87

3.6 Conclusion . . . 90

4 Morphological Labelling using Supervised Techniques 92

4.1 Introduction . . . 92

4.2 Morphological labelling as a classification problem . . . 93

4.2.1 Data sources and preparation . . . 95

4.2.2 Feature specification . . . 98

4.2.3 Segmentation and composite suffixes . . . 101

4.2.4 Machine learning algorithms and experiment setup . . . 103

4.2.5 Data representation . . . 106

4.2.5.1 Composite suffixes . . . 106

4.2.5.2 Feature distribution . . . 109

4.3 Experiments and results . . . 110
4.3.1 The optimal cascade sequence . . . 110
4.3.2 Merging second-tier features . . . 116
4.3.3 Classifying unseen data . . . 121
4.3.4 Performance evaluation of the Cascade Classification System . . . 127
4.3.5 Comparison of different machine learning techniques . . . 130
4.4 Classifying nouns and adjectives . . . 132
4.4.1 Dictionary data extraction . . . 133
4.4.2 Examining morphological properties . . . 136
4.4.3 Extracting basic features . . . 137
4.5 Experiments and results . . . 140
4.5.1 The optimal cascade sequence . . . 140
4.5.2 Classifying unseen data . . . 143
4.6 Conclusion . . . 144

5 Towards a complete Morphological Classification System 146
5.1 Introduction . . . 146
5.2 Category classification . . . 147
5.2.1 Experiment setup . . . 150
5.2.2 Dataset preparation . . . 151
5.2.3 Category classification results . . . 152
5.3 Gold standard evaluation . . . 154
5.3.1 Extracting basic features . . . 156
5.3.2 Choosing the best classification . . . 157
5.3.3 Data preparation for manual annotation . . . 158
5.4 Analysis of results . . . 160
5.4.1 Results for the category classifiers . . . 160
5.4.2 Results for the three cascades . . . 162
5.4.2.1 The verb cascade . . . 163
5.4.2.2 The noun cascade . . . 166
5.4.2.3 The adjective cascade . . . 168
5.5 Conclusion . . . 169

6 Conclusion 171
6.1 Introduction . . . 171
6.2 Summary and main conclusions . . . 171

6.2.1 Principal challenges and themes 174
6.3 Future work 174
6.4 Final conclusion 176

References **177**

List of figures

1.1 The Ġabra online interface 3

3.1 An example of a Trie data structure 57
3.2 The welcome page on the evaluation website 75
3.3 The webpage showing the instructions in textual format 76
3.4 The evaluation webpage 77
3.5 Removal of words 78
3.6 Reinserting words in a cluster 78
3.7 Rating the quality of words 79
3.8 Evaluation matrix 86

4.1 An abstract system of cascade classifiers 95
4.2 The Ġabra online interface 98
4.3 Results of experiments for composite suffix representation 107
4.4 Results for different distribution in suffix and composite suffix data 108
4.5 Results for the Regular cascade sequence 114
4.6 Analysis of the Gender and Person features 115
4.7 Analysis of the Form feature 116
4.8 Results for the Flat cascade sequence 118
4.9 Comparison of the floor benchmarks for the Regular and Flat cascades . . 118
4.10 Comparison of the ceiling benchmarks for the Regular and Flat cascades . 119
4.11 Comparison of the Regular and Flat cascades 119
4.12 Comparison of the classifiers learnt from the different training datasets. . . 120
4.13 Results of the Regular cascade on unseen data 122
4.14 Results of the Flat cascade on unseen data 124
4.15 Comparing the Regular and Flat cascades on unseen data 124
4.16 Results of the Person-Gender cascade 125

4.17 Results of the Number-Person cascade 125
4.18 Results of the Number-Gender cascade 125
4.19 Results of the Subject cascade 126
4.20 Comparison of all the cascade classifiers on the unseen data 126
4.21 Performance of the Regular cascade against a baseline 128
4.22 Performance of the Flat cascade against a baseline 128
4.23 Analysis of accuracy on the Regular cascade 129
4.24 Analysis of accuracy on the Flat cascade 130
4.25 Comparison of machine learning techniques on the Regular cascade 131
4.26 Comparison of machine learning techniques on the Flat cascade 131
4.27 Dictionary entry for *ABBUŻ* 'abuse' 134
4.28 Dictionary entry for *KAMRA* 'room' 135
4.29 Dictionary entry for *AĊĊELERATUR* 'accelerator' 139
4.30 Results for the Adjective cascade 142
4.31 Results for the Noun cascade 143
4.32 Results of the Adjective cascade on unseen data 144
4.33 Results of the Noun cascade on unseen data 144

5.1 Abstract architecture for the proposed morphological classification system 149
5.2 Results for the Basic category classification 153
5.3 Results for the Augmented category classifiers 153
5.4 The spreadsheet given to annotators 159
5.5 F-Measure for category classifiers on gold standard data 161
5.6 Results of the verb cascade on gold standard data 165
5.7 Results of the verb cascade on gold standard data for expert 1 165
5.8 Results of the verb cascade on gold standard data for expert 2 166
5.9 Comparison of the verb cascade on unseen and gold standard data 167
5.10 Results of the noun cascade on gold standard data 167
5.11 Comparison of the noun cascade on unseen and gold standard data 168
5.12 Results of the adjective cascade on gold standard data 169
5.13 Comparison of the adjective cascade on unseen and gold standard data . . 169

List of tables

1.1 Root-based and stem-based morphology examples 2
1.2 Examples of inflection and derivation in the two systems 5
1.3 Verbal inflections for *qasam* 'break' 6
1.4 Verbal inflections for *aċċetta* 'accept' 6
1.5 Verbal inflections with direct and indirect object 7
1.6 Examples of the morphological properties for the noun category 8
1.7 Examples of the morphological properties for the adjective category 9

2.1 Summary of Results for Segmentation Techniques 35
2.2 Summary of results for clustering techniques 40
2.3 Summary of results for labelling techniques 43

3.1 Corpus statistics 55
3.2 Iteration to test stem-affix boundary 58
3.3 Accuracy for splitting of composite suffixes (Keshava and Pitler, 2006) . . . 60
3.4 Accuracy for splitting of composite suffixes (Dasgupta and Ng, 2007) . . . 61
3.5 Initial clusters for the stems *verifika, vverifika and ivverifika* 64
3.6 Initial clusters for the stems *piċċ, piċċa and spiċċa* 65
3.7 Example of semantic cohesiveness values 67
3.8 Description of test clusters used in non-expert evaluation 72
3.9 Average size of the clusters evaluated 74
3.10 Summary of the responses received from the evaluation 80
3.11 Number of words removed 80
3.12 Percentage of words removed 81
3.13 Removal of words in the test clusters 82
3.14 Quality ratings 83
3.15 Inter-Annotator Agreement 85

3.16 Inter-Annotator Agreement excluding outliers 87
3.17 Comparison of CON and NC clusters in expert group 88
3.18 Number of words removed by CON/NC groups 89
3.19 Quality of clusters by CON/NC groups 89

4.1 Example of labelled data from the Morpho Challenge dataset 96
4.2 Distribution of lexemes in Ġabra database 97
4.3 Second-tier features in Ġabra 100
4.4 Segmentation possibilities 102
4.5 An example of the different representation for a split composite suffix . . . 107
4.6 Composite suffix distribution in dataset 108
4.7 Examples of errors in the imperative 109
4.8 Attribute selection using information gain ratio 112
4.9 Analysis of F-Measure to select classifier sequence 113
4.10 Analysis of all classification cascades 120
4.11 List of words extracted from a sample dictionary entry 135
4.12 Number of words per grammatical category 136
4.13 Analysis of morphological properties 138
4.14 Adjectives cascade sequence selection 141
4.15 Nouns cascade sequence selection 142

5.1 An analysis of the distribution between verbs, nouns and adjectives 152
5.2 Metrics for the category classifiers on the gold standard data 161
5.3 Comparison of the augmented category classification 162
5.4 Performance of the augmented category classifier 163

Chapter 1

Introduction

1.1 The Maltese language

Maltese, the national language of the Maltese Islands, is considered as a 'mixed' language with a stratum of Arabic, a Romance (Sicilian, Italian) superstratum and an English adstratum. The Semitic influence is mainly evident in the basic grammatical structure, whilst the non-Semitic aspect of the language is manifested in its lexis (see Brincat (2011) for a historical perspective, and Borg and Azzopardi-Alexander (1997) for a descriptive grammar of Maltese). As a result of the historically different language family sources for its lexis, morphology in Modern Maltese follows two systems: a root-and-pattern or templatic system and a stem-and-affix system (Fabri, 2010; Fabri et al., 2014). The root-and-pattern system is typical of Semitic languages, whereby inflectional and derivational forms are produced through a pattern applied on a set of three or four consonants (also referred to as radicals) in a specific sequence. For example, from the root √nżl one can produce the derivational forms *niżel* (1^{st} verbal form: 'descend'), *niżżel* (2^{nd} verbal form: 'lower') and *nżul* (nominal form: 'descent'). Through the island's historical and cultural influences, Maltese borrowed profusely from Romance (Sicilian and Italian in particular) as well as from English. Some of the borrowing has been assimilated into this grammatical structure. For example, the word *namra* 'attraction' is of Sicilian origin ('innamuratu') which has undergone the gemination of the second consonant to become a verb — *nammar* 'court, attract'. However, a number of loan words follow a purely concatenative morphology, with both inflectional and derivational forms produced through affixation alone. Table 1.1 shows the non-perfective (ipf) of verbs displaying these forms, *qasam* and *gideb* taking a root-based form, and *aċċetta* and *eżamina* taking a stem-based form.

Table 1.1 Root-based and stem-based morphology examples

	qasam √QSM 'break'	*gideb* √GDB 'lie'	*aċċetta* 'accept'	*eżamina* 'ezamine'
1SG	n-aqsam	n-igdeb	n-aċċetta	n-eżamina
2SG	t-aqsam	t-igdeb	t-aċċetta	t-eżamina
3SGM	j-aqsam	j-igdeb	j-aċċetta	j-eżamina
3SGF	t-aqsam	t-igdeb	t-aċċetta	t-eżamina
1PL	n-aqsm-u	n-igdb-u	n-aċċetta-w	n-eżamina-w
2PL	t-aqsm-u	t-igdb-u	t-aċċetta-w	t-eżamina-w
3PL	j-aqsm-u	j-igdb-u	j-aċċetta-w	j-eżamina-w

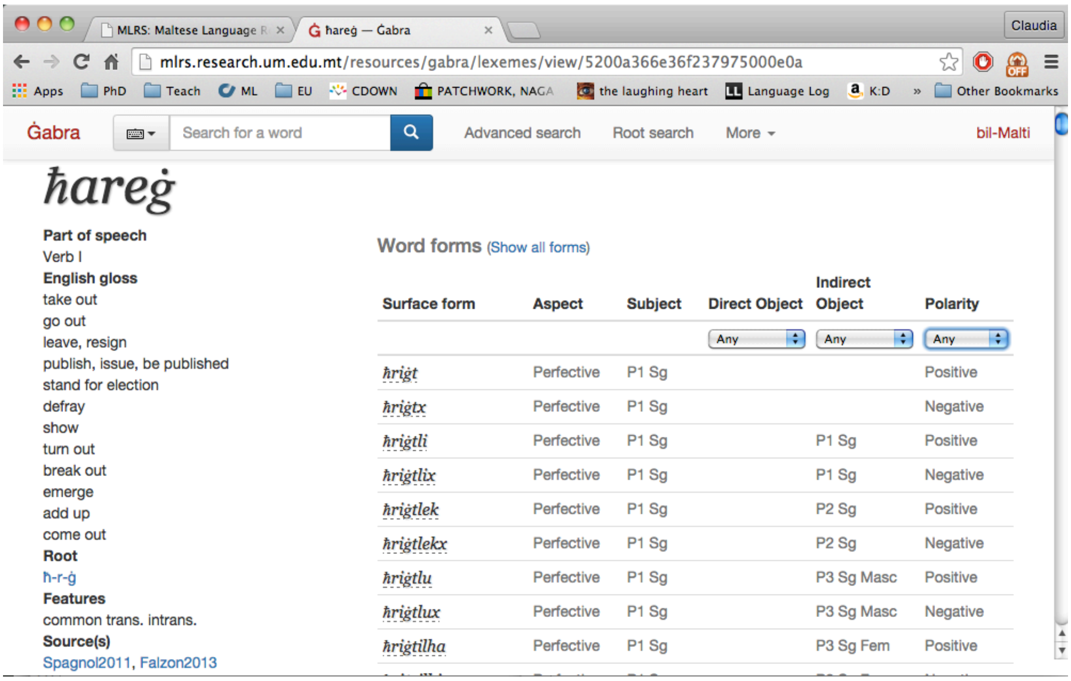
Since its inclusion as an official language of the European Union in 2004, there has been a increased interest in the creation of computational linguistic tools which are urgently required for tasks such as machine translation and parsing. A Maltese national corpus (MLRS — Malta Language Resource Server) was built from various textual sources, including newspaper articles and blogs, parliamentary debates, laws, literature and academic writings. It contains over 125 million tokens and is currently available online[1]. A part-of-speech tagger was also developed (Gatt and Čéplö, 2013) and the corpus was tagged with part-of-speech category. The digitalisation of a Maltese-English dictionary is in its final stages through an ongoing project, and will become available online[2]. However, computational treatments of Maltese are somewhat lacking in many areas, and the lack of basic tools makes it difficult to develop advanced NLP technologies for Maltese in the areas of NL understanding and generation. Morphology is one such basic technology. Evidence for this statement comes from a white paper published by the METANET consortium on the digital status of Maltese (Rosner and Joachimsen, 2012), which concluded that the state of language resources for Maltese is still lagging behind a number of major languages.

There are several linguistic studies that looked at different aspects of Maltese morphology, such as verbal inflection and derivation (Fabri, 2009), the broken plural (Schembri, 2006), the verbal derivational system (Spagnol, 2011), the use of personal pronoun enclitics (Camilleri, 2009), the use of gender in nouns (Farrugia, 2010) and verbal nouns (Ellul, 2015) among others. Schembri's work was used as the basis for a computational system that

[1] http://mlrs.research.um.edu.mt

[2] http://www.maltesedictionary.org.mt

used a neural network to associate the singular with its respective broken plural (Farrugia, 2008). Dalli (2002) produced an initial lexicon for Maltese, and this was further developed by Attard (2005). However, this lexicon did not materialise as an online resource. The analysis of the templatic verbs by Spagnol (2011) was used by Camilleri (2013), who created a computational grammar for Maltese for the Resource Grammar Library (Ranta, 2011), with a particular focus on inflectional verbal morphology. The grammar produced the full paradigm of a verb on the basis of its root. Considering that a verb can have over 1,400 inflective forms, this resulted in a large lexicon of over 4 million words. Since the grammar specifies the production rules to form the inflective wordforms, each word in the lexicon contains its morphological information and its root. This resource is known as Ġabra and is available online[3]. Ġabra allows users to query the database by searching for a specific root or word, and displays all the inflective forms of an entry. Figure 1.1 shows part of the entry for *ħareġ* 'go out'.



ħareġ

Part of speech
Verb I
English gloss
take out
go out
leave, resign
publish, issue, be published
stand for election
defray
show
turn out
break out
emerge
add up
come out
Root
ħ-r-ġ
Features
common trans. intrans.
Source(s)
Spagnol2011, Falzon2013

Word forms (Show all forms)

Surface form	Aspect	Subject	Direct Object	Indirect Object	Polarity
			Any	Any	Any
ħriġt	Perfective	P1 Sg			Positive
ħriġtx	Perfective	P1 Sg			Negative
ħriġtli	Perfective	P1 Sg		P1 Sg	Positive
ħriġtlix	Perfective	P1 Sg		P1 Sg	Negative
ħriġtlek	Perfective	P1 Sg		P2 Sg	Positive
ħriġtlekx	Perfective	P1 Sg		P2 Sg	Negative
ħriġtlu	Perfective	P1 Sg		P3 Sg Masc	Positive
ħriġtlux	Perfective	P1 Sg		P3 Sg Masc	Negative
ħriġtilha	Perfective	P1 Sg		P3 Sg Fem	Positive

Fig. 1.1 The Ġabra online interface

Ġabra is, to date, the best computational resource available in terms of morphological information. Its main limitation is its focus on the root-and-pattern based morphology and it is limited to the knowledge-base in its database. Although the grammar itself is available

[3] http://mlrs.research.um.edu.mt/resources/gabra/

as open source, it is used only to generate all the wordforms of the encoded templates, and is not intended to act as an actual morphological analyser.

1.1.1 Main aims and goals of this research

This research aimed to contribute further in the computational direction, providing further digital resources for the Maltese language, focussing primarily in computational morphology. A morphological analyser is one of the foundational tools for the computational processing of a language, allowing further sophisticated processing tools to infer aspects about the text being processed. To date, there was no morphological analyser for Maltese, and the attempts in this direction dealt with only particular aspects of morphology. This was the first systematic attempt to tackle Maltese morphology at a broader level, laying the foundations for a future morphological processing system for Maltese.

The research also aimed to explore the hybridity of the language, and how this aspect impacts the technological side of morphological processing. This study is the first of its kind, with all previous work in computational morphology for Maltese focusing mainly on the templatic (root-and-pattern) system. Where possible, we explored and evaluated how the performance of the techniques implemented fared on the two morphological systems and provided a discussion in the analysis of the results. The techniques implemented might favour aspects of one system over the other, and through this analysis it was possible to determine what worked best and for which system.

Finally, although the techniques built were only trained and tested on Maltese, the actual framework could be applied to the computational analysis of morphology in any language, thus contributing to the field of computational morphology in general.

In the following section (§1.2) we will give a brief overview of the properties of Maltese morphology that this research dealt with, focusing on verbs, nouns and adjectives. §1.3 will then describe the challenges and prospects for computational morphology for Maltese. An outline of the remainder of this thesis is provided in §1.4, together with a summary of the main contributions of the research carried out.

1.2 Morphological properties in the Maltese language

The main part-of-speech categories that this research dealt with were verbs, nouns and adjectives. In this section we provide the reader with an overview of the morphological properties of each of these categories. For a more complete review of Maltese morphology, see Borg and Azzopardi-Alexander (1997); Grech (1980); Mifsud (1995a); Sutcliffe (1936).

In our work, no distinction was made between the concatenative (stem-and-affix) and non-concatenative (templatic, root-and-pattern) systems. Since the aim of this work was to create a single morphological system that would be able to treat Maltese morphology systematically, the techniques created were applied to both systems in the same way. Similarly, no distinction was made between inflective and derivatational morphological processes. The techniques built were generally not expected to identify the type of word formation or relation, but rather to identify the morphological properties of a word independently from these aspects.

The hybridity and mixture of the two systems makes Maltese morphology an interesting problem to explore from a computational perspective. Table 1.2 provides an example of the difference between the two systems for the inflective and derivational morphological processes. Unlike the initial example in table 1.1 shown previously, where some affixes were common amongst the verbal inflections, the affixation for both the derivational and inflectional processes in the following example differ for both the concatenative and non-concatenative systems.

Table 1.2 Examples of inflection and derivation in the concatenative and non-concatenative systems

	Derivation	**Inflection**
Concatenative *eżamina* ‘examine’	*eżaminatur* ‘examiner’	*eżaminatr-iċi*, sg.f *eżaminatur-i*, pl.
Non-concatenative *gideb* ‘lie’ √GDB	*giddieb* ‘liar’	*giddieb-a*, sg.f. *giddib-in*, pl.

1.2.1 The verb

The verb category is one of the morphologically richest in Maltese, with various morphological properties that feature in the work described in the rest of the thesis. Table 1.3 shows the inflectional paradigm of a non-concatenative verb, whilst table 1.4 shows the paradigm of a concatenative verb. The verb agrees with the subject in terms of *person* (first, second or third), *number* (singular or plural), and *gender* (masculine or feminine). In the case of gender, it is only the third person singular that has gender assigned to it, with the other forms being gender neutral. The verbs are inflected for tense/aspect (perfective vs imperfective) and mood (imperative).

Table 1.3 The verbal inflections for the root √QSM — *qasam* 'break'

Subject	**Perfective**	**Imperfective**	**Imperative**
1.sg.	qsamt	naqsam	-
2.sg.	qsamt	taqsam	aqsam
3.sg.m.	qasam	jaqsam	-
3.sg.f.	qasmet	taqsam	-
1.pl.	qsamna	naqsmu	-
2.pl.	qsamtu	taqsmu	aqsmu
3.pl.	qasmu	jaqsmu	-

Table 1.4 The verbal inflections for the stem *aċċetta* 'accept'

Subject	**Perfective**	**Imperfective**	**Imperative**
1.sg.	aċċettajt	naċċetta	-
2.sg.	aċċettajt	taċċetta	aċċetta
3.sg.m.	aċċetta	jaċċetta	-
3.sg.f.	aċċettat	taċċetta	-
1.pl.	aċċettajna	naċċettaw	-
2.pl.	aċċettajtu	taċċettaw	aċċettaw
3.pl.	aċċettaw	jaċċettaw	-

Verbs in Maltese also take the direct and indirect pronouns (clitics), with table 1.5 providing some examples of such conjugations. A full paradigm can have over 400 forms. Both the direct and indirect object contain information pertaining to the person, number and gender.

Table 1.5 Verbal inflections showing conjugations in different tenses using the direct and indirect object

Subject	Direct Object	Indirect Object	Perfective	Imperfective	Imperative
1.sg.	2.sg.	-	qsamtek	naqsmek	-
3.sg.m.	3.sg.f.	-	qasamha	jaqsamha	-
2.pl.	1.pl.	-	qsamtuna	taqsmuna	aqsmuna
1.sg.	-	2.sg.	qsamtlek	naqsamlek	-
3.sg.m.	-	3.sg.f.	qasmilha	jaqsmilha	-
2.pl.	-	1.pl.	qsamtulna	taqsmulna	aqsmulna
2.sg.	3.sg.m	1.sg.	qsamthuli	taqsamhuli	aqsamhuli
3.sg.m.	3.sg.f.	2.sg.	qasamhielek	jaqsamhielek	-
3.pl.	3.pl.	3.pl.	qsamuhomlhom	jaqsmuhomlhom	-

1.2.2 The noun

Nouns in Maltese can have several morphological properties, however, this discussion will be limited to those properties that feature in our work. Maltese has simple and derived nouns, with words such as *baħar* 'sea' being simple, and *rġulija* 'manliness' from *raġel* 'man' being derived. In the datasets used, there is no specific distinction between these two types of nouns and both are treated in the same manner. As seen in a previous example, nouns can also be derived from verbs — *giddieb* 'liar' from *gideb* 'to lie'. Derivations could also come from adjectives, such as *bjuda* 'whiteness' from *abjad* 'white'.

For this category, the most prominent morphological properties in the data available were *gender* (masculine, feminine and neutral), *number* (mainly singular and plural, with some other properties such as collective, countable and singulative), *diminutive* and *verbal* nouns. The words for some of these morphological properties are formed through different processes — for example, the plural can be formed using either concatenative suffixation or what is referred to as the broken plural, where the word is formed through change in the internal structure of the word. The data used does not distinguish between concatenative and broken plurals, but treats all the plural formations as a single class. In the case of gender, nouns can either be exclusively masculine or feminine, or the feminine form can be produced from the masculine. Table 1.6 provides a number of examples to demonstrate some of the differences in each of the properties.

Table 1.6 Examples of the morphological properties for the noun category

Gender	**Masculine**	**Feminine**	
	alfabet ‘alphabet’, sg.m	(no feminine)	
	(no masculine)	*aljotta* ‘fish soup’, sg.f.	
	tifel ‘boy’, sg.m.	*tifla* ‘girl’, sg.f	
Number	**Singular**	**Plural**	**Collective**
	aljotta ‘fish soup’, sg.f.	*aljotti*, pl.	
	qassata ‘cheese cake’, sg.f.	*qassatat*, pl.	
	omm ‘mother’, sg.f.	*ommijiet*, pl.	
	bieb ‘door’, sg.m.	*bibien*, pl.	
	tifel ‘boy’, sg.m.	*tfal* (broken pl.)	
	barmil ‘bucket’, sg.m.	*bramel* (broken pl.)	
	ħobża ‘a loaf’, sg.f.	*ħobżiet* (definitive pl.)	*ħobż* ‘bread’
	kelma ‘word’, sg.f.	*kelmiet* (definitive pl.)	*kliem* ‘words’
		kelmtejn (dual pl.)	
Diminutive			
	xatt ‘shore’	*xtajta* ‘little shore’	
	dar ‘house’	*dwejra* ‘cottage’	
Verbal Nouns			
	qabeż ‘to jump’	*qbiż* ‘jumping’	
	għallem ‘to teach’	*tagħlim* ‘instruction’	

1.2.3 The adjective

The main morphological properties dealt with in the present work for the adjectives are *gender* (feminine, masculine), *number* (singular, plural) and *agent* (an adjective describing that which is acted upon). The plural of an adjective is gender neutral. Although Maltese does have the comparative form (e.g. *sabiħ* ‘nice, beautiful’, *isbaħ* ‘nicer’, *l-isbaħ* ‘nicest’), the number of examples in our data was too small to be used, and therefore these were left out.

Table 1.7 Examples of the morphological properties for the adjective category

Gender (Singular)		**Number (Plural)**
Masculine	**Feminine**	
ħelu ‘sweet’	*ħelwa*	*ħelwin*
kbir ‘great’	*kbira*	*kbar*
ħafif ‘light’	*ħafifa*	*ħfief*
oħxon ‘gross, fat’	*ħoxna*	*ħoxnin*
abjad ‘white’	*bajda*	*bojod*
Agent		
	waħx ‘ogre, fear’	*waħħaxi* ‘frightening’
	gerriem ‘one who gnaws’	*gerriemi* ‘that gnaws’
	bennien ‘to cradle’	*bennieni* ‘rocking to sleep’

1.3 Computational morphology for Maltese: Challenges and prospects

Computational morphology is a subfield of computational linguistics, mainly concerned with the computational analysis of words in terms of their internal structure and the grammatical information that words or their constituents encode. A morphological analyser is an important tool in natural language processing and is a stepping stone to other more sophisticated tools such as spell checking and parsing. It is also crucial to many other areas. For example, the treatment of morphologically rich languages is currently a focus of work in machine translation; in natural language generation, having the ability to generate morphological forms is absolutely crucial to the production of coherent, fluent text. Consider the following two phrase in Maltese:

(1.1)	*fjura*	*sabiħa*
	flower.N.Sg.**F**	beautiful.Adj.Sg.**F**
	a beautiful flower	

(1.2)	*fjura*	**sabiħ*
	flower.N.Sg.**F**	beautiful.Adj.Sg.**M**
	a beautiful flower	

The first example shows the correct grammatical agreement between the two words, whilst the second examples shows incorrect agreement. If a spell checker is limited to a list of Maltese words, it would not detect the error in the second example since its spelling is correct. In order to detect such errors, the morphological information of words is required so that agreement between the noun and the adjective can be factored into the process of spelling and grammar correction. The previous section described the various morphological properties that Maltese verbs, nouns and adjectives have. Although these were described in isolation, the syntactic structure of a sentence in Maltese must consider these grammatical properties to ensure that the correct agreement is in place. A morphological analyser is used by any computational tool that parses a language to extract information. It can also aid in applications such as machine translation, improving the matching and alignment between two languages. Thus a morphological analyser can be seen as a foundational computation tool which will allow more sophisticated applications to further process the language.

The primary challenge for Maltese morphology is finding a way of dealing with the two morphological systems simultaneously. The hybridity of Maltese morphology makes it interesting to study, not just from a computational or technological point of view, but also as an exploration of how different techniques would fare on the different morphological systems found in the language. Another challenge is the limited resources available. During the initial stages of this project, the only resource available was the MLRS corpus. Over the lifetime of this project, the part-of-speech tagger was developed and the corpus annotated, the lexicon Ġabra was published, and the digitalisation of a Maltese-English dictionary is now in its final stages. The limitation in digital resources placed restrictions on the type of techniques that could be explored. However, the more resources became available and continued to improve over time, the more we could experiment with sophisticated machine learning techniques to aid the creation of a morphological analyser for Maltese.

From a computational perspective, this thesis attempts to examine all three aspects of computational morphology: segmentation, relations and labelling (Hammarström and

Borin, 2011). We approach the first two tasks using unsupervised techniques, starting off from a wordlist extracted from the MLRS corpus and using various techniques to segment and cluster morphologically related words together. The labelling task then shifts to supervised techniques, using the Ġabra and the dictionary datasets to model and test a morphological classification system for Maltese. Although the research carried out focussed solely on Maltese, the techniques used are language-independent, and could be applied to other languages given the necessary resources.

1.4 Outline and contributions of the thesis

The rest of the thesis is structured as follows. Chapter 2 provides an overview of related work, exploring issues pertaining to morphology both from a language acquisition perspective, and a computational perspective. We analyse different techniques used for different languages and consider the treatment of computational morphology in Semitic languages as well. Chapter 3 delves into a number of unsupervised techniques aimed at automatically segmenting words and clustering morphologically related words together. It also discusses the results of an evaluation exercise carried out to assess the quality of the resulting clusters. Chapter 4 introduces morphological labelling as a classification problem, and outlines a number of experiments done to model a morphological classification system for verbs, nouns and adjectives, whilst chapter 5 looks at the classification of a word's category and the extent to which category classification, a task typically performed by a POS tagger, could be performed out of context on the basis of morphological information, and if so, to what extent such morphological labelling would help a POS tagger. An evaluation of the resulting classification system was also carried out using expert linguists and discussed in chapter 5. Chapter 6 concludes by providing a summary of the main contributions of this thesis and outlines the future directions for this research.

The main contributions of this thesis are summarised as follows:

Segmentation We implemented a segmentation technique based on Keshava and Pitler (2006) and Dasgupta and Ng (2007) that proposes potential segmentations for Maltese words. However, since the technique works best for concatenative morphology, the system is not restricted to a single segmentation hypothesis, but rather allows a number of probable segmentations which can be discarded at a later stage by the other tasks.

Clustering We developed a clustering technique that considers both orthographic and semantic similarity of words in an attempt to group together morphologically related words. The technique builds on similar work by Baroni et al. (2002); Schone and Jurafsky (2000, 2001), but proposes a way of measuring the disparity or similarity of a group of words. The metric is then used as a means of improving clusters through a merging and discarding process.

Gold standard dataset of clusters At the time of implementing the clustering technique, there was no lexicon or data that could be used for evaluating the clusters. So an evaluation exercise was carried out that included both experts and non-experts, resulting in manually checked clusters. This dataset could be used in future for further analysis and improvement on the clustering technique, and also from a psycholinguistic point of view to analyse how Maltese native speakers treat derivational words in terms of their conceptual view of related words.

Online Maltese dictionary In the effort to gather data with morphological labels, we automatically extracted and processed a substantial part of a Maltese-English dictionary. The data was essential for this project to consider nouns and adjectives. The extracted data was also used by the dictionary project to further supplement their data.

Morphological Classification System A system of classifiers for verbs, nouns and adjectives was developed which segments words and classifies them both for their part-of-speech category, as well as their morphological features. Although the accuracy of the classification system is not exceptionally high for some of the features, the system performs well overall and can easily be retrained on more, improved data as this becomes available.

Gold standard dataset with morphological labels A small dataset was produced by two linguistics experts to aid the evaluation of the classification system. This is the first gold standard dataset available for Maltese morphology, and can become the basis for a larger dataset in the future.

Apart from tangible results such as software applications and datasets, this work has also addressed the issues of a hybrid morphological system throughout its timeline, offering insight and discussions on how different techniques and ideas perform on the two morphological systems. We did not seek to discard a technique because it does not work as well on one morphological system as it does on the other, but rather modified it in such a way that it could still be useful for both systems. The main goal throughout remained that of finding approaches which would work reasonably well on both morphological systems, and if this did not happen, trying to understand why. At the end of the journey, this thesis did not produce a final morphological analyser that can be used off the shelf. But it does provide a solid foundation through which a final morphological analyser for Maltese is feasible and well within reach, and with further improved datasets the classification system can be retrained and packaged as a service for further language processing applications for Maltese.

This thesis presents the first systematic computational treatment of Maltese morphology using machine learning techniques. Whilst previous approaches were either rule-based or restricted in their scope (for example, focussing only on one sub-system of Maltese morphology), the present work does not restrict itself in this manner.

Chapter 2

Computational Morphology

2.1 Introduction

This chapter provides a review of the state of the art in computational morphology. Computational morphology systems provide the morphosyntacic analysis of words — their output is used either directly by human users or other software systems, such as parsing, grammar correction and other natural language processing tools. The primary focus of this research is on the computational treatment of Maltese morphology. Whilst most studies have been carried out from a linguistic perspective, only a handful attempt to deal with aspects of Maltese morphology from a computational perspective (Attard, 2005; Camilleri, 2013; Cutajar, 1990; Dalli, 2002; Farrugia, 2008; Galea, 1996). Rather than being merely a technical project with the sole aim of producing a morphological analyser for Maltese, this research also looks at possible ways of automatically bootstrapping morphological analysis for Maltese with the resources available at the different stages of this research.

This research unfolded as more resources for Maltese became available, exploring different techniques for different tasks in computational morphology. In terms of data sources available for Maltese, the starting point was the MLRS corpus[1], which contains a vast selection of texts in Maltese covering several genres, such as law, literature, newspapers and academia, with circa 120 million tokens. A part-of-speech tagger was developed and the corpus was tagged (Gatt and Čéplö, 2013) with basic category information. In parallel, a lexicon of Maltese words, called Ġabra[2], was produced, primarily focusing on verbal inflections from a closed-set list of roots documented in Spagnol (2011). This lexicon groups

[1] The corpus is accessible online after a free account registration through the Maltese Language Resource Server website — http://mlrs.research.um.edu.mt/

[2] http://mlrs.research.um.edu.mt/resources/gabra

morphologically related words under one lemma, and provides grammatical feature information for each lexical entry and its related forms (Camilleri, 2013). An ongoing project is currently in its final stages, and aims to produce a Maltese online dictionary[3]. Part of this dictionary project included the automatic extraction of dictionary entries from a scanned version of the Aquilina dictionary (Aquilina, 1987–1990). Apart from extracting the words in the dictionary entries, it was also used to generate various wordforms through the information in each entry. This supplemented the Ġabra lexicon with nouns and adjectives, and is currently being manually checked to be completely integrated with the online lexicon. Our research began with an investigation of simple techniques, exploring ways in which morphologically related words could be grouped together. As the work progressed, we moved from segmentation and clustering of morphologically related words, towards labelling morpheme segments. The backbone motivation throughout our work was to investigate ways in which a morphological analyser could be bootstrapped for a language with a rich and hybrid morphology system such as that found in Maltese.

The literature review begins with a consideration of language acquisition in children. Artificial Intelligence always looks at human behaviour and reasoning for inspiration, and thus it is interesting for this research to look at how children learn that, for example, *walks* and *walked* refer to the same action that occurs at different times. If a morphological analyser was already available for Maltese, this research could have investigated further whether and how the discussed mental models could be replicated computationally, and applied to Maltese. However, the focus remained on the actual production of morphological information that can be derived from words.

Computational approaches to morphology tend to be either rule-based or data-driven. The main focus of this research is on data-driven approaches and aims to explore different techniques and analyse what works and what doesn't in the case of Maltese morphology. A short review is presented on standard rule-based approaches, providing a glimpse of the most common techniques in morphological analysis, together with a short discussion of their advantages and disadvantages. A more detailed review is then presented for data-driven approaches, focusing on segmentation, clustering and morphological labelling tasks. Each task has its specific challenges, and all are relevant for the computational treatment of Maltese morphology.

For a standard and in-depth introduction to the vast field of morphology, see Haspelmath and Sims (2010); Jensen (1990); Nida (1949); Spencer and Zwicky (2001) and others.

[3] http://www.dizzjunarjumalti.com

2.2 Language acquisition

Children are able to learn a language in a natural way, just as they are able to learn how to walk. Although language is a complex system, the seemingly effortless acquisition process suggests that there is an innate capacity to learn a language which is part of the regular developmental process of the brain. Chomsky (1965) labelled this mental capacity the Language Acquisition Device (LAD), which enables children to construct a grammar and generate phrases in the well-formed structure of a language. This innate capacity to learn a language is seen as a biological property of the regular developmental process of the brain. An opposite view is that language is learned through social interaction as a tool for communication. Berko (1958) describes experiments using a set of plausible nonsense words and asking young children to provide a morphological form such as the plural. One of the results is that children are able to generalise acquired rules and apply them to words that they have not encountered before. Berko argues that this is evidence that children apply morphological rules to extend their vocabulary.

There is general agreement among linguists and psycholinguists that we have implicit knowledge in acquiring a language. However opinions differ when it comes to describing this knowledge — how does the mind store language (representation) in the mental lexicon, and how does it access the mental lexicon (processing) when required by the speaker? Does the mind store every word as an individual unit, or does it store morphemes? Similarly, does the mind analyse a word before storing it, or does it leave each entry unanalysed? In other words, when we want to use the word *chairs*, does our processor retrieve the word *chair* and add the plural suffix *-s* to it, or does it simply retrieve *chairs* from our mental lexicon? We briefly look into the different theories that have been debated to gain insight on how our brain deals with morphological processes and whether this could be modelled computationally.

Pinker and Prince (1988) state that language and cognition require rules and symbol-manipulating processes which can generate the same performance as a human, discovering their behaviours and perceptions, processing this information through the elementary symbol-manipulating mechanisms out of which rules are composed. A rule-based system requires a number of fundamental distinctions: (i) lexical items are unique, each an idiosyncratic set of syntactic, semantic, morphological and phonological properties — so words such as *ring* and *wring* are unique even though the sound sequence is phonologically ambiguous; (ii) there is a distinction between a morphological category and its realisation — morphemes can be syncretic and mark different categories, however a category is manifested through the words it represents; (iii) morphology describes the syntax of words whilst phonology describes the predictable features of the sound structure. A rule-based system can thus be seen as an organisation of these principles, feeding the information from one component to the other. Generally, each word will have only one form for each morphological category, and the effect is that when a general rule overlaps a specific rule, the specific rule not only applies, but blocks the general rule from applying.

The connectionist model maps the cognitive and behavioural development model to a computational model, namely artificial neural networks, that can be used to develop and test different theories and model developmental change (Westermann and Mareschal, 2005). A neural net tries to approximately replicate the biological setup of the brain — a web of neurons (processing units) connected through fibres (weighted paths) resulting in some form of output. Rumelhart and McClelland (1986), proponents of the connectionist model, used neural networks to replicate the acquisition of English past tense verbs in children, and simultaneously, strengthen the theory behind the connectionist model. The foundation of this theory is that, although it is evident that some form of mental function occurs to form the past tense, we do not explicitly create rules that produce these formations. Since the knowledge of word formation is implicit, it follows that there is some form of association between the input and the output forms, without specific knowledge of what the association consists of, and thus modelled through the implementation of a neural network. The main issue reflected in the connectionist model is that perhaps the concept of rules has no biological basis, and generalising new instances is based on analogy and similarity. In their experiments, Rumelhart and McClelland state that the learning curve of their neural network is similar to the one found in children as they develop and acquire correct use of the past tense in English.

The connectionist model has been strongly criticised, with various scholars proposing other models. In particular, Pinker and Prince (1988) rebutted the connectionist model on several points, arguing that the neural network model is limited in several aspects, including: (i) it cannot represent certain words due to the structure used as input to the neural net, (ii) it is limited in the number of 'rules' it can learn, and (iii) it learns associations that are not found in any human language. Moreover, the model itself does not provide any insight into morphological and phonological regularities, and the difference between regular and irregular forms. And finally, according to Pinker and Prince, the model does not accurately emulate acquisition in children, since the model's predictions do not match observed data from child language. In further work, Prasada and Pinker (1993) argue that whilst the production of irregular verbs may be generalised by a neural network model, regular verbs should be dependent on production rules. Ling and Marinov (1993), and Ling (1994) offer an alternative symbolic implementation that learns the past tense.

Although there have been several developments in the connectionist model (see Christiansen and Chater (1999); Elman (2005); Knight (1990); Li et al. (2004)), the debate on how children develop morphological processes and learn word formations remains open. Wintner (2010) provides a balanced overview of the computational models from both camps for language acquisition, and identifies the aspects that must be present in models and arguments if they are to be considered a plausible solution to the discussion. To begin with, such systems must be trained and evaluated on dedicated corpora that truly reflect language acquisition in children, such as the CHILDES corpus[4]. Ideally such systems should also be evaluated on more than one language, and rigorously evaluated on real data. Computational models should also focus on learning language from a set of strings, leaving the task of inducing syntactic structure to future research. A plausible model must also explicitly state the formal properties of the class of languages it generates and point out that using unrestricted context-free languages is likely to over-generate. Models should also include different biases that reflect psycholinguistic frameworks, such as item-based learning and rote learning. With these various observations, Wintner proposed a way to better compare and contrast computational models of language acquisition.

In providing a cognitive perspective on language acquisition, Kit (2003) questions what is learnt first: the various cues used for discovering words or the words themselves? He argues that cues are derived and not an innate ability that is within the child prior to the learning process. Thus it is not possible for a child to learn cues without any experience of knowing some words. This strengthens the idea that distributional information (repetition and co-occurring patterns) plays an important role in the lexical acquisition process. (Kit, 2003, p. 36) states that the ***'successes in machine learning of a natural language lexicon indicate that the distributional regularities in the data may play a fundamental role in human lexical acquisition, and it is essential to assume that human infants have an innately endowed mechanism to utilise such statistical information in language learning'***. Although, the mechanisms in human and machine learning are different, this is an interesting observation that supports the bias towards statistical approaches in computational morphology learning.

Bybee (1995) proposes a *network* model for the storage and processing of morphology based on child language development and cognitive organisational principles. The model

[4] CHILDES is the child language component containing child conversational interactions. It can be found online: http://childes.psy.cmu.edu/

takes type frequency into account, arguing that regular verbs have the highest type frequency whilst irregular verbs have the highest token frequency. The model establishes lexical connections between words, built on the basis of phonological and semantic information. The stronger the lexical connection between a number of words, the stronger the morphological relation is — a strong lexical connection between inflected verbs ending with the suffix *s* will increase the likelihood that the suffix will be used when a new word needs to be inflected, therefore making it more productive. In contrast, words with high token frequency have greater lexical autonomy reflected by weaker connections with other items. Thus Bybee argues that high frequency items are learned individually and lower frequency items are learned in relation to existing items through analogy. As learning progresses, the lexical connections are reinforced and generalisations can emerge from the network and fitted into schemas. The productivity of a schema is dependent on the number of restrictions placed on it — e.g. the *strung* pattern cannot be fully productive since it is a verb beginning in consonant clusters and ending in a velar and/or nasal, and most verbs do not meet these conditions; on the other hand, a more generic schema like *-ed* suffix for the past tense can be applied to more words and thus resulting in a higher type frequency (and higher productivity). It is worth noting here that this model suggests that rules and connections are not mutually exclusive, and that rules can be extracted after statistical generalisations are computed.

Baayen (2007) discusses the balance between storage and computation in the mental lexicon. Whereas the connectionist model is completely tilted to favour computational approaches, Baayen brings forward experimental evidence that this is not the case, for various reasons. Similarly, limiting storage only to irregular verbs, as done by the Pinker model, is also not the case. There is evidence that storage is not limited to irregulars, and that regular inflected verbs have 'their own traces' in memory. (Baayen, 2007, p.4) claims that the idea that the past tense is derived in real time from the stem (either through connectionism or through the application of a rule) is also wrong, citing various experiments where subjects where asked to provide either the past or the present tense of a word according to what they were shown. In these experiments, the direction of naming (past to present or present to past) had no influence on the results. On the basis of various experiments, Baayen states that an adequate metaphor for understanding the mental lexicon is still not available. Whilst the connectionist model merges rules and representations, making it neurologically implausible, the symbolic model's divide between regular and irregular processes is too simplistic. Baayen proposes that recent research in the design of an algorithm replicating the structure and properties of the neocortex should be considered and applied to morphological acquisition. This is still a network of neurons similar to the connectionist model, but in which the neurons also have a hierarchical structure and those in the higher hierarchy act as classifiers.

Although not directly related to language acquisition, in further work, Plag and Baayen (2009) focus on derivational morphology and, in particular, the ordering of composite suffixes, both from the perspective of analysis and that of generation. They propose that suffix ordering should include selectional restrictions (affix properties, based on phonological, morphological, syntactic and semantic factors) and processing constraints (limiting the number of combinations of affixes) so as to have a hierarchy of suffixes which determines the order in which these can be encountered. Complexity-based Ordering (CO) is a processing constraint that limits the number of possible combinations of affixes. For example, the suffixes -iz-ation commonly appear in that order but not as -ation-ize (e.g. human-iz-ation). Notwithstanding, there are still cases where suffix ordering is easily 'broken' — for example the suffix -al occurs both at the end of the composite, and also at the beginning in sens-ation-*al* vs. conoli-*al*-iz-ation. Another problem is when the order of two suffixes can be reversed, something referred to as the inside-outside problem, as in 'happinessless' vs. 'hopelessness'. To identify the correct ordering, four 'measures of separability' are used (productivity, token parsing ratio, type parsing ratio, boundary strength), resulting in a ranking mechanism (CO rank) that provides an indication of which combination of suffixes are most likely to occur in which order. The analysis of the hierarchical order of suffixes indicates that suffixes occurring close to the stem are more likely not to occur outside of other suffixes. Plag and Baayen also observe that the mean CO rank is correlated to the productivity of a suffix. The higher its rank, the more productive the suffix is (e.g. *-ism*). The mean CO rank is also used to judge the cost between storage and access vs. decompositional cost in the mental lexicon. Interpreting the results of lexical decision and word naming experiments Plag and Baayen show that storage has an advantage over decomposition. They observe that storage requires less effort than decomposing complex derived words.

The above discussion shows that, even at a psycholinguistic level, morphology can be approached from different angles. The various proposals discussed above support their claims through computational models in support of their arguments. This raises the question of whether a computational model should try to mimic language acquisition development in a child. The opposite might be the case, where the computational model should be designed specifically to deal as best as possible with the described task, rather than designing it to mimic the language model.

2.3 Computational approaches to morphology

The task of a morphological analyser is to provide the morphological structure and analysis of any given word. A number of very successful, knowledge-based morphological analysers are available for a number of languages, relying on manually designed heuristics or rules that require linguistic expertise and are laborious to construct. Moreover, such systems must be continuously updated to cope with language change through the emergence of new words. Roark and Sproat (2007) provide an excellent general overview of the main algorithms used in computational models of morphology and the general operations involved in describing regular morphological phenomena. This section will provide a general overview of two main computational approaches to morphology: rule-based and data-driven. Since we are mainly interested in the latter approach, the review for the rule-based approaches will be relatively concise, mentioning the key techniques and providing a general overview of what is available and why it works. The review for the data-driven approaches is more detailed, highlighting the different approaches used, the type of tasks that have been successfully attempted and the challenges that remain.

2.3.1 Rule-based approaches to morphological analysis

Initial attempts in computational morphology were rule-based, with implementations that were language specific and restricted to describing limited aspects of a language. As the use of computers and computational power increased, more interesting work in computational morphology, both from a theoretical and a computational point of view, began to emerge. Finite state transducers (FST) emerged at the forefront of rule-based techniques since with the possible exception of reduplication (which may require further mechanisms), all morphological phenomena can be expressed through finite-state devices, at the cost of having duplication. Another advantage of FSTs is that they cater for a declarative approach, allowing for a system to both analyse and generate morphological information through the specification of grammar rules.

Chomsky and Halle (1968) specified a traditional phonological grammar using context-sensitive rewrite rules. The rules converted abstract phonological representations into surface forms through a series of intermediate representations. Johnson (1972) found that it was possible to write phonological rewrite rules as regular expressions, thus showing the viability of using FSTs. This idea was overlooked at the time, and rediscovered independently by Kaplan and Kay (1981). However, FSTs were still not applied to morphological analysis at this stage.

Koskenniemi (1983) proposed a novel approach called Two-Level Morphology, which uses a constraint-based model in which rules are applied in parallel and all constraints must hold simultaneously. Rules are expressed as a pair of strings, and can refer to both lexical and surface context at the same time. But rather than focusing on how one string can be derived from another, the rules express mutual constraints on those strings. The system offers a general language-independent model through which language-specific components (lexicon and rules) could be specified. An actual compiler for two-level rules was not available at the beginning, resulting in practitioners having to compile rules to finite-state transducers manually, which was a highly laborious task.

Eventually compilers became available, including a popular implementation called PC-KIMMO (Antworth, 1992). Karttunen and Beesley (2005) provide an overview of the historical development of FSTs and two-level morphology and note that, overall, the formalism was difficult for linguists to master and that it was necessary for the compiler to check for and eliminate conflicting rules. Nonetheless, this was one of the first rule-based systems that could cater for morphologically complex languages, allowing for both generation and analysis within the same grammar specification. The formalism eventually was included as part of the Xerox tools, with the latest documentation by Beesley and Karttunen (2003). Well-known morphological analysers based on FST technology are those by Beesley (1996) for Arabic, Itai and Wintner (2008); Yona and Wintner (2005) for Hebrew, and Minnen et al. (2001) for English. Roark and Sproat (2007) provide an excellent introduction to finite state automata and transducers, as well as a detailed overview of the two-level morphology system.

Rule-based approaches to computational morphology are popular and successful. However, the main drawback remains that expert knowledge is required to design and write the rules, and continuous effort is needed to keep the knowledge base up to date with unseen words. This means that this approach is resource intensive and thus not viable for all languages. Since Maltese became an official language of the European Union in 2004, more resources have been made available. However, this has been a slow process and only recently (more than 10 years later) have funds been allocated for a much-needed national online dictionary[5]. This provides the incentive to work towards the bootstrapping of a Maltese morphological analyser, which might eventually feed into a rule-based technique.

[5] There are small scale online dictionaries for Maltese which are domain specific, e.g. financial terminology

2.3.2 Data-driven approaches to morphological analysis

The data-driven approach aims to learn the morphology of a language, or aspects of it, through statistical observations from the data. Most data-driven approaches treat morphological analysis purely from a computational/engineering perspective, with the main concern being of the type of algorithms designed and the results achieved. Hardly any of the computational literature in this area is based on linguistic theories. However it is still interesting to note the differences and maybe as well the indirect similarities between the computational approaches described below, and the way children learn. Data-driven approaches use corpora, which means that words can be analysed in context. A child's acquisition context is much richer than just the words themselves and contains various cues (visual, prosodic). The learning process occurs through numerous examples over time. The acquisition process exhibits non-linear characteristics, as children acquire a rule, then overgeneralise, and eventually revise and adjust the exceptions. What type of information can be extracted from textual sources, and how such data could be used to increase what can be learnt is also an interesting question for this work, especially since computational resources available for Maltese are limited.

Comparing different techniques or different results is not always possible because the authors often do not carry out an evaluation using the same training and testing data. Sometimes the evaluation principles — for instance, what is considered a correct or incorrect segmentation — could differ from author to author. The main focus of this work is to analyse the different approaches that have been taken and to apply the most promising approaches to morphology learning in a fusion system such as Maltese. For a more complete literature review of unsupervised techniques for morphology learning, see also Goldsmith (2010); Hammarström and Borin (2011).

2.3.2.1 Problem definition

The classical way a morphological analyser works is that, given a word, it produces a labelled output. Most commonly, the word is segmented into morphemes and each morpheme is labelled according to the information that it provides, as in the example below.

(2.1)	*n-*	*approva*	*-hom*	*-lok*
	1.sg.sb-	approve	-3.pl.do	-2.sg.io

'I will approve them for you'

A number of intricate steps are required to produce a labelled form as an output. First, a word is split into the smallest meaningful components — *morphemes*. This process is referred to as **segmentation**. The process of splitting a word into parts is not always straightforward, since there can be morpho-phonological/orthographic alternations in the stem, such as *commerce → commercial*). Sometimes, computational literature also looks at the splitting of a word from an application-oriented point of view, where the stem is going to be used in a specific end-task such as machine translation or information retrieval. In such cases, it might be sufficient for an algorithm to produce a 'partial' stem, e.g. **commerci -al*, providing enough information for the application to carry out its task. This first stage of a morphological analysis system takes a word and returns it split up, ideally in as close an approximation as possible to its morphemes.

Once a word is segmented and the stem is identified, it is possible to group morphologically related words together. The stem of a word provides a link to the lexeme that the wordforms expresses and, through this, the word can be associated with other words sharing the same stem, and by extension the concept. The process of pairing or **clustering** of morphologically related words is of interest in a data-driven approach because it is possible to observe how words are transformed depending on the morphemes applied to the stem. In contrast, a rule-based approach generally does not cluster words since the rules implicitly cluster words by providing information for the generation and analysis of all the transformations. In a data-driven approach, the transforms must be observed through the data itself, and hence it is necessary to associate morphologically related words. In this second stage, the system uses the stem of a word to find other words that are morphologically related to it, thus resulting in a list of words.

The final step is that of **labelling** the morphemes, where each part is labelled according to the morphosyntactic information that it conveys — grammatical meaning for the affixes, and semantic meaning for the stem (some systems choose to associate the lemma of a word). To learn actual labels (e.g. 3SG), an algorithm must be given some labelled examples (training data). A model is built upon these examples, and then tested on unseen data (test data). By comparing the labelled output produced by the algorithm to the original test data, it is possible to evaluate the algorithm with respect to how accurate the labelled output is. This is referred to as supervised learning, since the algorithm is provided with examples that can be used to create the model. When labelled data is not available, an algorithm can output generic 'classes' rather than actual labels (e.g. 'type1'), and all the morphemes sharing that label should share a grammatical feature. This is referred to as unsupervised learning, in which case the algorithm does not have any prior knowledge. The conversion from generic labels to actual labels has to be carried out by a human expert.

Most of the work reviewed below use unsupervised techniques for morphology learning, with particular focus on segmentation and some looking at clustering and labelling. The latter task is usually attempted using semi- or supervised techniques. The remainder of this section will analyse the works carried out and group these papers according to the type of task and technique used. However, it is important to note that the three tasks need not happen in sequence and are not fully dependent on each other. This is especially so in clustering morphologically related words, where the segmentation task is a step that can be bypassed and sometimes occurs as an indirect consequence of the clustering output, thus resulting in an overlap between segmentation and clustering. Similarly, the labelling task might depend on segmentation but does not require clustering of words. Again, labelling might produce clusters indirectly, once the associated stem is identified.

Given the different tasks in computational morphology and the challenges that each task represents, a yearly competition was held called Morpho Challenge. Each year the focus of the competition became more complex, starting from simply tackling segmentation using unsupervised techniques, and moving on to semi-supervised systems to labelling of morphemes. Since the Morpho Challenge provided training datasets for a number of languages, as well as a baseline performance for the tasks concerned, most research in the field refers to the challenge, and some researchers use the datasets to evaluate their algorithms. In the following, we will first describe the Morpho Challenge, and then review the different tasks identified above. We will end the section by looking at the relevant work in Semitic languages that does not necessarily fit in with the three established tasks. The computational treatment of morphology in Semitic languages is of particular interest, since Maltese morphology has both root and pattern (i.e. non-concatenative) and stem-based (i.e. concatenative) morphology. We will, therefore, take stock of the challenges faced in computational morphology for Semitic languages and discuss whether such challenges are relevant to Maltese.

2.3.2.2 Morpho Challenge

The Morpho Challenge[6] is an evaluation challenge aimed at advancing machine learning and particularly unsupervised techniques that can be applied to multiple languages, and intended to identify morphemes and the phenomena that occur in word construction. Morpho Challenge editions were held in 2005, 2007, 2008, 2009 and 2010. Kurimo et al. (2010) report on the editions held until 2009 and outline various open challenges. Apart from evaluating word segmentation, the challenge also evaluated how different techniques

[6] http://research.ics.aalto.fi/events/morphochallenge/

could be applied to the areas of information retrieval and statistical machine translation. This review will not include the results from these two application-oriented tasks. However, one can get an immediate sense of the level of success in the segmentation task from the observation that the overall results throughout the competition, in terms of the highest F-measure, remained below 0.70 (English: 0.687; Finnish: 0.569; German: 0.628; Turkish: 0.621; Arabic Vowel: 0.632; Arabic non-Vowel: 0.608[7]). The majority of the techniques that competed did not display a large disparity in the results. Disambiguating between different alternative analysis of word segmentation and assigning labels to morphemes once the segmentation has been carried out remained an important challenge to the improvement of the results. In 2010 semi-supervised learning was introduced to tackle these challenges; however the results did not improve much for the segmentation task (English: 0.674; Finnish: 0.625; German: 0.508; Turkish: 0.653). The evaluation was carried out against the output of a morphological analyser for each respective language and this output was considered as a gold standard.

Several works reviewed below refer to the Morpho Challenge, especially in terms of evaluation, and often compare their systems to the results published through this competition. However, direct comparison is not always fully possible or obvious. Sometimes it is not clear whether an author would have simply used the datasets available on the Morpho Challenge website[8], or submitted their system to Morpho Challenge. The test sets used for evaluation during Morpho Challenge are not publicly available.

Another issue relates to the actual evaluation metric used. Up until 2009, Morpho Challenge used standard F-Measure, Precision and Recall to compare results achieved by different techniques. In 2010, a new metric was introduced, referred to as EMMA, and more attention was given to the problems encountered in the evaluation of such techniques. The 2010 competition was evaluated using both the standard F-Measure and EMMA. The figures above, and throughout this review, are generally F-Measure for the sake of comparability (when possible).

Apart from Morpho Challenge, where comparable evaluation is possible due to the setup of the task and resources used, comparing the results of the other work reviewed can be a futile exercise. Whilst some evaluations use metrics such as F-measure and Ac-

[7] Arabic orthography generally uses diacritics to mark short vowels, however these are commonly left out in everyday use. The Morpho Challenge provided the same corpus with the vowels included, and without the vowels.

[8] The Morpho Challenge website has a number of datasets available for several languages. It provides a training set for training an algorithm or model, and a development set that is intended to be used during development for parameter tweaking. It also provides datasets for semi-supervised learning with gold standard segmentation and labelling.

curacy, others assess their work on the basis of a more qualitative evaluation. Even when metrics are used, factors such as type of corpora used and what is being learnt make the results difficult to compare side by side. The aspects of evaluation, and the different metrics available and used will be discussed in further detail in §2.5.

Keeping in mind that the main aim of the competition is to facilitate other tasks (information retrieval, machine translation), Kurimo et al. (2010) raise a number of valid observations arising from the Morpho Challenge competitions: (i) No single algorithm was able to perform equally well on all the tasks across the languages, (ii) segmentation is only a small part of the process and finding meaning to the morphemes is a more crucial task, (iii) words taken out of their context could result in ambiguous segmentations, and it is not clear how to evaluate algorithms that return a high number of analyses per word, thus inflating recall. It is clear that although advances were made both in terms of techniques and results over the various competitions, segmentation and labelling are not straightforward tasks and a number of challenges are still relevant in this field.

2.3.2.3 Segmentation

Goldsmith (2001) is one of the focal works on the unsupervised learning of morphology, interest in which reemerged in the late 90s. He proposes an unsupervised algorithm that learns the morphology of a concatenative language by using Minimum Description Length (MDL). The MDL algorithm seeks to describe a group of words using the most compact description possible (both in terms of length and in terms of physical memory space on a computer), and at the same time having the most compact means of extracting that representation. The model is composed of four parts: (i) a probability distribution is assigned to the sample space from which the data is taken, (ii) the data is then compressed using information-theoretic notions, (iii) the length of the description is calculated, (iv) the optimal description is the one for which the sum of the length of the compressed data and the length of the model itself is the smallest. The algorithm continues to iterate until it finds a final optimal solution. The resulting description takes the shape of *signatures*, where each signature represents a set of stems and a set of suffixes that can be combined together – for example the set of stems {*laugh, walk, jump*} and the set of suffixes {*ed,ing,s*}. The principal aim of the technique is to discover automatically the regularity of a language and generalise it in the most efficient way. Once the signatures are proposed, signatures with either only one stem or only one suffix are discarded and those words are then treated as whole when the length of the description is calculated. Moreover, a stem can only be represented in a single signature. Goldsmith also introduces a structure to describe more complex suffixes (e.g. *work.ing.s*). A number of heuristics are used to suggest at which point a word should be split into stem and suffix. For instance, the suffixes in every signature are checked for their initial letter — if they all begin with the same letter, then it is likely that the letter must be part of the stem.

The MDL technique achieves 82.9% accuracy on the manual evaluation of the first 1000 words. Suggested improvements include the possibility to specify the deletion of a character (e.g. the suffix set {*NULL.<e>ed.<e>ing*} would cater for grouping *care, cared, caring* under one signature, through the deletion notation <*e*>). Goldsmith discusses how the signature cannot be mapped directly to a paradigm because in some cases the stems belong to multiple classes (e.g. verbs and nouns both having the suffix -*s*). Some of the limitations for MDL are that it does not cater for compounding and does not handle prefixes. He also discusses the evaluation of the signatures at length, and questions the correctness of some of the proposed segmentations. For example, the words *abet, abetting, abetted* resulted in having the stem set {*NULL.ting.ted*}, when clearly there should be the facility to predict that the *t* is doubled in such cases. Another example raises the difficulty in what should exactly be considered as a stem — in the words *abolish, abolition*, should the stem be *aboli*- or *abol*? Goldsmith decided that both analyses should be considered valid since this is not a clear-cut case. Finally, in a rather controversial concluding point, Goldsmith states that *"knowledge of semantics and even grammar is unlikely to make the problem of morphology discovery significantly easier."* Whilst it is true that the introduction of this information might result in a more supervised technique, other research avenues used this type of information to improve the results of morphology learning. Work on MDL continued and resulted in a reduction of signatures and thus a better representation of a language (Goldwater and Johnson, 2004; Hu et al., 2005); however the principal technique and its aim remained the same: that of representing the regularity of a language in the most efficient way.

Creutz and Lagus (2002) take two different approaches to segmenting words into morphemes, focusing in particular on languages with rich morphology, using an agglutinative language such as Finnish. Their motivation behind segmentation is to "provide a vocabulary of language units that is smaller and generalises better than the vocabulary consisting of words as they appear in text." The first technique combines recursive segmentation and the minimum description length principle, very similar to the process described by Goldsmith (2001), with the main difference being the approach to the segmentation of a word. Goldsmith uses a predefined list of suffixes, whilst here the authors use a 'search' approach which takes into account the number of times a morph has been encountered. The technique also updates the model at regular intervals, resegmenting words if necessary. The second technique uses sequential segmentation and maximum likelihood, which uses the likelihood of the data given the model as a cost function. The first technique developed into Morfessor-Baseline in future work, and became the baseline evaluation technique in the Morpho Challenge competition against which all other participating techniques were evaluated.

Creutz and Lagus (2004) describe an extended version of Morfessor which uses the generative probabilistic model to segment words into morphs, using a Hidden Markov Model (HMM) to model morph sequences. A morph is assumed to belong to one of three categories - prefix, suffix or stem, and part of the procedure is to determine the categories of each morph. Once a morph is assigned to a particular category, it can only appear in a particular sequence — e.g. a suffix cannot appear before a stem. The technique starts with a baseline segmentation model (Creutz and Lagus, 2002), and then initialises the probability distributions for each morph, assigning categories to each. At this point an interim category 'noise' is used to hold morphs that initially cannot be considered prefixes, stems or suffixes. Noise morphs tend to arise as a consequence of over-segmenting rare word forms in the baseline word splitting. The next step looks at eliminating redundant morphs, generally by splitting a morph into two existing morphs, taking the most probable split. Finally noise morphs are removed by being joined with adjacent morphs until a stem can be formed. Through the process of mapping morphs to a category, the segmentation of words is improved considerably over the baseline. Creutz and Lagus point out that the evaluation of segmentation can yield different results according to the application for which the segmentation is aimed — whilst morphological analysers might provide morpheme-based segmentation, applications such as machine translation or speech recognition might leave certain morphemes unsegmented to yield better results in the application itself. This further intensifies the difficulty of the evaluation of segmentation results.

Morfessor is packaged into a suite of a full range of techniques (Creutz and Lagus, 2005, 2007) aimed at morphology learning and also used as the baseline for the Morpho Challenge described above. In Virpioja et al. (2013), Morfessor was further developed, with various computational improvements made to the group of algorithms; support for unicode was also introduced. Morfessor remains the main technique that most other techniques are evaluated against since it offers state of the art results for most languages included in the Morpho Challenge.

Keshava and Pitler (2006) propose a simple and intuitive technique to segment words using transitional probabilities. The approach uses a trie data structure that stores the words found in a corpus and their relative frequencies. A trie data structure is a type of programming structure that can hold data in an efficient manner since strings with multiple overlapping substrings are stored without excessive redundancies. A word boundary is considered as a possibility on the basis of three simple premises: (i) that the stem appears as a valid word in the corpus, (ii) that at the point being considered as a boundary the sequence from the root to that point has more or less the same frequency, indicating the likelihood of a stem, and (iii) that at the point being considered as a boundary there is evidence of the trie branching off into several substrings (which will in turn be considered as potential suffixes). The approach results in a number of potential affixes which are scored according to how many times they have passed through these three conditions. The top N affixes will then be used to segment words in the corpus. Keshava and Pitler suggest that the algorithm is better suited for Indo-European languages with a concatenative morphology. The main advantage of this work is the simplicity of quickly obtaining a list of potential affixes, especially if one is considering a bootstrapping scenario where possibly the list of affixes could be checked and corrected by a linguist. However, there are a number of drawbacks to this technique. First, in calculating a word boundary, the technique only considers those boundaries where the stem has been attested in the corpus. This can be rather restrictive, especially if stem variation occurs in the formation of related words. The technique also does not cater for composite suffixes since it considers a single boundary at a time. Nonetheless, the technique remains simple and plausible for bootstrapping purposes and considered advantageous.

Dasgupta and Ng (2007) extend the work by Keshava and Pitler and include ways of detecting incorrect segmentations (e.g. ‘candid+ate’ vs. ‘candidate’) by relying on relative corpus frequency and suffix level similarity. This is calculated as a ratio in terms of the frequency of the word as a whole, and the frequency of the proposed stem. The segmentation is considered to be possibly correct if the ratio is set to be below a certain threshold. Suffix level similarity is motivated by the observation that if a stem takes a particular suffix, it should also take morphologically similar suffixes. Another extension looks at inducing orthographic rules and allomorphs, allowing the system to detect changes in morphemes at boundaries, and again this is based on relative frequency. The segmentation strategy taken by the authors is to generate all possible segmentations of a word using the induced stems and affixes, and then applying a number of tests until only one candidate segmentation remains. One of the tests described by Dasgupta and Ng is that when a number of segmentations are left, the one/s with the least number of morphemes is then chosen. This is then followed by a scoring heuristic that takes into account the ‘strength’ of the affixes and the stem. The authors do not justify the order of these tests; discarding a potential segmentation due to the number of morphemes prior to checking their statistical relevance might result in discarding a more correct segmentation where, say, a suffix is correctly split into composite suffixes, thus being the reason for having more morphemes. The system is evaluated on a corpus of English and Bengali, achieving F-Measures of 0.874 and 0.851, respectively. It is further evaluated on the English, Turkish and Finnish data from the Morpho Challenge 2005, where it achieved F-Measures of 0.794, 0.662 and 0.652 respectively. The English results, in particular, demonstrate how evaluating the same system on different datasets yields different results — here with an 8% drop in performance. The authors do not attempt to explain the reasons behind such a drop, and merely compare their technique against the other techniques in the Morpho Challenge competition. Nonetheless, the system performs well overall, and either obtains similar results to the best performing system, or better results, which is encouraging given the mostly simple and straightforward techniques set in place to segment words.

The technique proposed in Chan (2006) learns a POS-associated, probabilistic representation of regular morphology through the use of recursive Latent Dirichlet Allocation (LDA) to generate the probabilistic model. The model is built on the basis of three matrices: data, morphological probabilities and lexical probabilities. The system is limited to stem and suffix concatenation, assumes that the segmentations of words and their suffixes are already known and excludes rare suffixes. During the learning phase, the algorithm assigns a probability for a particular suffix belonging to a particular part-of-speech category. In the final model, a suffix is assigned to the category where the probability is the highest, and thus dismisses other possible assignments (e.g. *-s* is assigned to noun plural, and not to a verb). The model is also limited in learning regular morphology and does not distinguish between inflection and derivation. The final output of the system is a recursion tree with each node representing a set of suffixes in their respective grouping, and representing an abstract category. The evaluation is mainly focused on comparing the results with those of Linguistica (Goldsmith, 2001), which is a strange choice since the aim of Linguistica is to describe a language in its most minimal form as a way of arriving at the segmentation of words. In contrast, Chan's primary focus is deriving a form of paradigm, which does not provide a straightforward comparison to Linguistica's signatures. Although the author claims to have achieved the better results, the comparison is not clearly defined and it is not obvious which specific aspects of this approach provide the improvement.

Sirts and Goldwater (2013) proposed Adaptor Grammars (AGMorph), a nonparametric Bayesian modelling framework for minimally supervised learning of morphological segmentation. The model learns latent tree structures over the input of a corpus of strings. In an unsupervised way, the AG takes as input a grammar that is used to specify what the structure of the tree can be. However the authors also introduce a small amount of labelled data and a process to select the best grammar as a way to improve results, thus introducing a semi-supervised approach. This learning process is carried out for each language and data set. The techniques are evaluated on English, Finnish, Turkish, German and Estonian, and compared with Morfessor in terms of results. Overall the techniques perform well, approximately in line with the results achieved by Morfessor.

Poon et al. (2009) propose a log-linear model for unsupervised morphological segmentation which leverages overlapping features such as morphemes and their context. It incorporates an idea from Goldsmith by including exponential priors as a way of describing a language in an efficient and compact manner. The morphological segmentation of a word is viewed as a flat tree, and each leaf of the tree is a potential morpheme. Each leaf is also associated with its n-gram context as a feature. The system is evaluated on two datasets. One is the dataset used by Snyder and Barzilay (2008), which is evaluated on both Hebrew and Arabic. It achieves 0.669 and 0.781 F-Measure, respectively (an increase from 0.63 and 0.72). The same system was also trained in a semi- and fully-supervised way by providing labelled data incrementally. F-Measure increases from 0.759 to 0.809 and from 0.852 to 0.90 for Hebrew and Arabic, respectively, by providing 25% to 100% labelled data. A further evaluation is carried out for Arabic using the Arabic Penn Treebank dataset and the results are compared to Morfessor-CatMAP (Creutz and Lagus, 2007), achieving an F-Measure of 0.777 compared to Morfessor's 0.749. The overall improvement of the results is rather positive; however, since the evaluation is limited to two Semitic languages, it is not clear whether this technique is more geared towards such languages and whether it could be employed on other languages successfully. Also the authors do not discuss issues pertaining to stem variation, especially since the focus is on the segmentation of the words rather than linking a word to its lemma.

Building upon the idea of Poon et al. (2009), Narasimhan et al. (2015) also use a log-linear model, and morpheme and word-level features to predict morphological chains. A chain is a sequence of words that starts from the base word, and at each level through the process of affixation a new word is derived as a morphological variant. For example, the following is a morphological chain which the system attempts to derive: nation → national → international → internationally. The system automatically produces a list of all possible affixes on the basis of the base words present in the corpus. So from the word *paints*, the suffixes *-ints* (stem *pa*), *-ts* (stem *pain*) and *-s* (stem *paint*) are derived. A review of the top 100 suffixes shows that 43 are correct. This is rather low in terms of accuracy, and although such a technique would result in very high recall, its precision rate will probably be low. The conditional probability of each derivational pair is calculated and the system will recursively link words into a chain as shown in the example above. Each pair is also allocated a number of features, including semantic similarity and affix correlation (words which take the same set of affixes demonstrate a certain correlation between those affixes). The log-linear model is then learnt in an unsupervised manner with all the data collected automatically.

The authors report an improvement over Morfessor, AGMorph and the system proposed by Lee et al. (2011) using English, Turkish and Arabic and achieving F-Measures of 0.762, 0.612 and 0.799, respectively. The system does not handle stem variation since the pairing of words is done on the basis of the same orthographic stem and therefore the result for Arabic is rather surprising. This is probably due to the gold standard used in the evaluation, where segmented words do not necessarily give the valid root of the word. An example given by the authors is for the gold segmentation for yntZrwn, given as y-ntZr-wn, even though ntZr is not a valid root. The error analysis for English and Turkish shows that the predicted segmentation tends to under-segment, whilst it over-segments for Arabic. This might support the intuition that an unsupervised system would require some form of parameterised data on how to deal with a particular language. For example, what level of segmentation a probabilistic technique could favour can be a language-specific parameter. This is highly intuitive when comparing languages such as English and Turkish, with the latter having a highly agglutinative morphology. Unfortunately the authors do not consider the use of semantic similarity as a way of finding related words where stem variation occurs. Possibly one could foresee a system that would try to identify stem variation by giving more importance to the semantic similarity between words, whilst allowing minimal orthographic change in the stem.

Lee et al. (2011) incorporate part-of-speech categories with the morphological segmentation problem, arguing that the POS categories should reinforce potential segmentations since certain affixes are associated more frequently with specific categories. Moreover, if the POS categorisation also takes context into consideration, it could be used to determine the correct segmentation of a word. The technique models morpho-syntactic decisions jointly, with the intention of capturing linguistic phenomena such as agreement. The system first generates a list of prefixes, stems and suffixes in a hierarchical manner, and then proposes word types, their segmentations and their syntactic categories. An HMM model is then used to generate tokens and the syntactic classes, followed by a first-order Markov chain which has the dependencies between adjacent segmentations to segment the tokens. The system depends on a number of parameters, such as the number of POS tags possible (here set to 5, which when increased, did not yield improvements). The number of prefixes and suffixes is also limited to two, and the length of the stem cannot be shorter than that of the affixes. Such restrictions might be language dependent. For instance, the last restriction would be detrimental in Maltese, where it is at times possible to find words which have a longer suffix than the length of the stem itself, as shown in the following example:

(2.2)	*ksir*	*-ni*	*-hie*	*-lhom*
	broke	-1.PL.SB	-3.F.SG.DO	-3.PL.IO
	'We broke it for them'			

The system is evaluated on Arabic and compared to the output of Morfessor and Poon et al. (2009). Although the results obtained are an improvement over these two systems, the training of the model was carried out on the full dataset. The norm is that a technique is evaluated on held out data (or unseen data) which would not have been part of the training data. This means that the results presented (F-Measure of 0.862 compared to Morfessor 0.749 and Poon 0.777) could be inflated due to this fact. This continues to highlight the issue of comparability of different systems, even when the same dataset is used. Unless an evaluation is carried out using the same portions of data in training and testing, it is difficult to judge if one system is better than another.

Brychcín and Konopík (2015) build a language-independent high-precision stemmer (HPS), using a combination of lexical and orthographic information, as well as probabilities observed over the training data fed into the system. Words are initially clustered based on their semantic similarity, using the Maximum Mutual Information algorithm, and on their lexical similarity, calculated by using the longest common prefix normalised by the length of the longest word. The clusters are then used by a maximum entropy classifier as training data which calculates a number of probabilities, including suffix length, the probability of being an actual suffix and the probability of an n-gram standing before a suffix. It also allows for a parameter δ to set the aggressiveness of the stemmer, representing the ratio between the length of the stem and the length of the word. HPS was evaluated on a number of languages, including English, Czech, Slovak, Polish, Hungarian and Spanish, and the maximum suffix length was set to three characters for all languages. It was also evaluated against other stemmers, namely GRAS, YASS, Linguistica and a rule-based stemmer (Porter's stemmer) and through different tasks, including information retrieval, which is generally used to evaluate such tools. The evaluation concerning suffix removal was carried out by comparing the results of the system with the words clustered according to the lemmas present in the tagged corpora. With Hungarian and Spanish, rule-based stemmers preformed better than HPS, whilst with the remaining languages HPS performs better in terms of F-Measure. However, one must note that rule-based stemmers for Slovak and Czech were not available. HPS does achieve higher precision than the other techniques, but with lower recall. Unfortunately this work does not take into consideration the technique used by Schone and Jurafsky (2001) (discussed below), which also uses semantic and orthographic similarity to pair up morphologically related words, but uses different algorithms; it was also shown to improve the results over Linguistica (Goldsmith, 2001), one of the techniques evaluated.

Summary of segmentation results

The above review shows that segmentation is a challenging task and that there is no one-size-fits-all solution. Most approaches take an unsupervised approach, aiming to offer language-independent techniques, and finding stems and affixes through probabilistic measures. Table 2.1 provides an overview of the results discussed above. Where available, only the results for the segmentation of English words is provided since this was the language of choice for the majority of the work reviewed. Results for other languages are only included when English was not used (noted in the comments).

Table 2.1 Summary of results for the segmentation techniques reviewed
Type: 'U': Unsupervised; 'LS': Lightly Supervised; 'S': Supervised
Results: 'A': Accuracy; 'F': F-Measure

Citation	Type	Results	Comments
Goldsmith (2001)	U	A: 82.9%	
Creutz and Lagus (2002)	U	A: 49.6%	
Creutz and Lagus (2004)	U	F: 0.769	
Virpioja et al. (2013)	U	F: 0.763	
Keshava and Pitler (2006)	U	F: 0.809	
Dasgupta and Ng (2007)	U	F: 0.794	MorphoChallenge Data
Dasgupta and Ng (2007)	U	F: 0.874	Other data
Sirts and Goldwater (2013)	LS	F: 0.778	
Poon et al. (2009)	U	F: 0.781	Arabic
Poon et al. (2009)	S	F: 0.900	Arabic
Narasimhan et al. (2015)	U	F: 0.762	
Lee et al. (2011)	LS	F: 0.862	Arabic
Brychcín and Konopík (2015)	LS	F: 0.708	

The insights by Goldsmith (2001) provide an indication of the difficulties encountered when analysing what is a correct segmentation. The problem is reduced when a gold standard is available for evaluation — still in this case, at some point, either a machine or a person decided what is the correct segmentation for dubious cases. The segmentation task in the Morpho Challenge competition is a very useful starting point since it provides a gold standard for evaluation, and allows easy comparison across different systems. The simplicity of techniques such as those proposed by Keshava and Pitler (2006) make it ideal for bootstrapping a list of prefixes and suffixes for segmentation of words. However it is clear that a list of affixes alone is not sufficient for the segmentation of words. We have also begun to see some issues with the analysis of Arabic, and how the root-and-pattern morphology requires different approaches when compared to the stem-based morphology of Indo-European languages. The treatment of Semitic languages will be discussed in further detail below in §2.4. The use of multi-lingual resources to map similar morphemes from the same language family, as proposed by Snyder and Barzilay (2008), can be an interesting experiment when considering the hybridity of Maltese — we could envisage such an alignment using Maltese, Arabic, Italian and possibly English. However, it is evident from the results that the techniques by Keshava and Pitler (2006) and by Dasgupta and Ng (2007) obtain amongst the best results, using intuitive techniques which are ideal for a bootstrapping scenario.

2.3.2.4 Clustering morphologically related words

Relying solely on the statistics of hypothesised stems and affixes creates several problems and inaccuracies. Schone and Jurafsky (2000, 2001) use this argument to introduce the notion of semantic knowledge through the application of the Latent Semantic Analysis (LSA) algorithm. The technique is based on several sequential steps starting with potential affixes being identified on the basis of shared orthographic properties. Words that share the same stem are treated as potential variants (e.g. *care* with *care-s*; *car* with *car-s*; also **car-es* with *car*). The LSA is then used to find the semantic relationship between words in a document through a matrix which denotes the frequency of the surrounding context for particular words. Normalised cosine scores are then used to correlate the words which are potentially related. At this stage of the algorithm, it emerges that *cares* and *car* are not semantically related, and, therefore, that the pair **car-es* and *car* should not be ranked high.

Next, the technique looks at orthographic similarity based on the Minimum Edit Distance (MED) algorithm. MED is a way of quantifying how similar two strings are by computing the number of edits required to get from one string to another. The application of MED to compare orthography allows for changes in the stems, accounting for potential stem variation in words. The semantic and orthographic similarity of words is used to rank word pairs, with higher ranking indicating a more probable morphologically related pair of words. Potential signatures are then built and relations are extracted based upon certain conditions. The technique is compared to Linguistica and the evaluation uses CELEX[9] as a dataset. This technique manages to improve the F-measure to 0.84 and the authors conclude that semantic-based and frequency-based approaches could be used as complementary techniques to find morphologically related words.

Yarowsky and Wicentowski (2000) propose a combination of alignment models with the aim of pairing inflected words, covering both regular and irregular morphological processes. The technique relies on (i) a list of part-of-speech tags, (ii) a list of suffixes that can be used for each tag, and (iii) a list of lemmas. The proposed technique estimates a probabilistic alignment between an inflected form and the lemma, which also results in the extraction of the transformation process needed for the lemma to arrive to the inflected form. For example, the lemma *take* requires the following transformations to arrive to *took*: (i) stem change: *ake* → *ook*; and (ii) suffix: $+\epsilon$ (the empty string). This process results in a large table of lemmas, transforms (stem and suffix) and the resulting inflected words together with their part-of-speech tag. Lemmas are then aligned using different models, each providing an individual score for each lemma and ranked according to a final consensus choice. The alignment models are based on similarity in frequency distribution, context similarity, weighted Levenshtein distance and morphological transformation probability. The algorithm is trained and improved by iteratively bootstrapping the model of affixation and stem change probabilities, and re-estimating the alignment over the four similarity measures. At convergence, the model reaches an accuracy of 99.2% in pairing English verbs (lemma, past tense) for the set of lemmas provided with the 'VB' part-of-speech tag. Although the results are promising, Yarowsky and Wicentowski do not clearly specify how the transformations observed can be generalised and applied to unseen words. The main suggestion is that if an unseen word is lemmatized, and its part-of-speech is known, then the table can be looked up to provide the transformation rule. The morphological analysis provided in labelled examples seems to be mainly limited to the part-of-speech tags which have been provided for the algorithm from the beginning. Therefore the high accuracy reached is clearly due to the information provided, such as POS tags, suffixes and lemmas.

[9] CELEX is a hand-tagged, morphologically analysed database of English, German and Dutch words. The latest version is available at https://catalog.ldc.upenn.edu/LDC96L14

Yarowsky and Wicentowski highlight that the suffix-focused transformational model used is not sufficient for handling prefixes, infixes and reduplication, and that it is mainly suited for Indo-European languages.

Baroni et al. (2002) propose a system that attempts to extract morphologically related words without having any prior knowledge or annotation about the language. The system is provided with raw text and returns a list of ranked pairs (structured/unstructured). Two main techniques are used to extract this information: (i) orthographic similarity based on the minimum edit distance algorithm, and (ii) semantic similarity based on mutual information probability. Only word pairs that are in both lists are considered, so as to avoid considering semantically similar words that are unrelated (e.g. blue, green), and orthographic words that are also unrelated (e.g. blue, glue). A final ranking is produced by combining the scores of the weighted orthographic similarity and semantic similarity for each pair. The main advantages of this technique are that it can deal with both concatenative and non-concatenative morphological processes, and that it does not depend on the distributional properties of words and their substrings, and therefore it is able to discover rare morphological patterns. An evaluation of the top 1500 word pairs against the results of the XEROX parsing tools yields a precision of 91%. A simple rule induction program is used to extract various common morphological patterns found in the word pairs for a qualitative assessment.

Can and Manandhar (2012) create a probabilistic hierarchical model to cluster morphological paradigms on the basis of a word's segmentation. However, the definition of a paradigm here is closer to the idea of signatures described in Goldsmith (2001), where a morphological clustering aims at gathering those stems that share the same suffixes. The model uses stems and suffixes to combine words in the same clusters. However, it does not consider prefixes, infixes and circumfixes. A hierarchical structure is learnt through inference, and allows for morphologically similar words to be located close to each other and thus grouped in the same paradigm. Morphological similarity is defined as words having at least one common morpheme, but there is no distinction made on the type of morpheme — it can be either a stem or an affix. Segmentation is then carried out on the basis of the tree. The authors claim that the training of the model had to be limited to 22K words due to memory limitations, and if this could be increased, the results would improve. The system was evaluated against the Morpho Challenge datasets for English and Turkish, reaching an F-Measure of 0.58 and 0.38, respectively. The limitations for this type of model with a language like Turkish result from its rich morphology and high agglutination, something which the system is not set to cater for.

Ahlberg et al. (2014) produce inflection tables by obtaining generalisations over a small number of samples through a semi-supervised approach. The system takes a group of words and assumes that the similar elements that are shared by the different forms can be generalised over and are irrelevant for the inflection process. The focus is kept on where the variation occurs in the strings, and a template is extracted reflecting the paradigm of a set of inflectional word forms. Those having the same template can then be grouped as having the same type of inflectional paradigm. The technique is applied to various languages, including Spanish and German. The resulting work is very much in the same spirit of Camilleri (2013), who took a rule-based approach to inflectional paradigm generation.

de Roeck and Al-Fares (2000) cluster Arabic roots to improve information retrieval in Arabic. The motivation behind this work is that traditional morphological analysers for Arabic have a limited morphosyntactic coverage and that, in practice, there are several orthographic challenges. Unwritten short vowels and different regional spelling conventions are among the main challenges mentioned. A two-stage algorithm is designed, first applying light stemming, followed by calculating a word pair similarity coefficient. Initially, the similarity between pairs is seen as the factor of shared substrings and is based on n-gram overlap. However, the authors observe that two words with minor differences in the root consonants still had a high incidence of being clustered together since the similarity overlap also covers affixes. In order to reduce this error, words were first lightly stemmed by removing a small number of obvious affixes. Since this procedure does not remove all affixes, all substrings were assigned a weight of 1, whilst potential affixes and substrings containing weak consonants were assigned a weight of 0.5 and 0.25, respectively. These weights were arrived at empirically. The pair-string similarity coefficient is then calculated by taking these weights into consideration and by looking at the number of n-grams that overlap between two words. The evaluation focuses on the percentage of correct clusters — i.e. the percentage of clusters that contain only morphologically unrelated words. It does not look into whether a cluster is complete (contains all the word forms for a particular root). However, this might be due to the restricted dataset available and indeed, they emphasise the need for the algorithm to scale up to larger corpora. On average, 91% of the clusters produced are correct. Attard (2005) applied the same technique to Maltese, with the aim of providing the basis for a Maltese lexicon. However, this was done using very limited data and was not developed further.

Summary of clustering results

Clustering of morphologically related words is evidently more successful when additional information, such as part-of-speech category, is provided. Table 2.2 provides an overview of the works reviewed above. For the clustering review, it is more difficult to make a direct comparison between the different works for various reasons: (i) different datasets were used; (ii) different languages (including German, Spanish, Finnish, and Arabic); (iii) different metrics (F-Measure, Precision and Accuracy). Nonetheless, it is still interesting to note the results obtained through unsupervised techniques proposed by Baroni et al. (2002); Schone and Jurafsky (2000, 2001). This is an ideal option for the Maltese language when considering the limited resources available. Using the MLRS corpus to extract semantic relations between words would increase the amount of information available to a system, with the aim of determining morphologically related words in an automatic and unsupervised manner.

Table 2.2 Summary of results for clustering techniques reviewed
Type: 'U': Unsupervised; 'LS': Lightly Supervised; 'S': Supervised
Results: 'A': Accuracy; 'F': F-Measure; 'P': Precision

Citation	Type	Results	Comments
Schone and Jurafsky (2000, 2001)	U	F: 0.84	
Yarowsky and Wicentowski (2000)	LS	A: 99.2%	
Baroni et al. (2002)	U	P: 0.91	
Can and Manandhar (2012)	U	F: 0.58	MorphoChallenge dataset
Ahlberg et al. (2014)	LS	A: 96.4%	German, Spanish, Finnish
de Roeck and Al-Fares (2000)	LS	A: 91%	Arabic; Avg. on 5 datasets

2.3.2.5 Learning labels for morphemes

Van den Bosch and Daelemans (1999) use Memory-based Learning to learn morphological labels on a Dutch corpus, reformulating the problem of labelling as a classification task. A word is analysed letter by letter, with a window of 5 preceding and 5 following letters. Thus a word is represented by as many instances as it has letters. This approach also takes care of the 'segmentation' task through the windowing procedure, by learning labels when morphemes are present in a particular window, and thus indirectly resulting in segmenting the word into actual morphemes. Once the model is trained, unseen instances are classified according to a distance between a new instance and the existing instances in the model.

The distance metric used is a simple orthographic overlap, which practically results in matching the presence of morphemes. A weighted function is then applied to calculate the information gained and instances are labelled according to the closest match. The evaluation is carried out using 10-fold cross validation on regular running text — it looks at the different levels of success, including whether words were segmented correctly and whether they were labelled correctly, with an accuracy rate of 64.6% on unseen words. One of the limitations of the technique is that it can only return one segmentation for a word. However, Van den Bosch and Daelemans point out that it is rare for a word to have ambiguous segmentation in Dutch.

Clark (2002, 2007) also uses Memory-based learning, starting first by learning finite-state transductions between pairs of uninflected and inflected words. The stochastic transducers are then used by the distance function to determine which stored instance is closest to a new unseen instance, based on the conditional distribution of a set of possible outputs. The data used in this work is focused on specific word pairs, such as singular and plural German nouns and Arabic singular and broken plural pairs. Thus the algorithm is not specifically learning labels, but rather how to associate an inflected form with its base form. Although the technique is more sophisticated than Van den Bosch and Daelemans (1999), it is not directly comparable to this work since the data used and the formulation of the machine learning task is substantially different.

Kohonen et al. (2010) extend Morfessor (described in §2.3.2.3, on page 28, Creutz and Lagus (2002, 2004, 2005, 2007)) by introducing semi- and supervised approaches to the model learning for segmentation. This is done by introducing a discriminative weighting scheme that gives preference to the segmentations within the labelled data. The algorithm relies on the bias given by its priors to guide the segmentation, which in turn is affected by a balance between the priors and the model. The introduction of the discriminative weights means that the segmentation will take a strong source of information from the labelled data as the amount of segmentation that is preferred by the gold standard. The weights are optimised on separate held out data. The best improvement over the Morfessor-baseline is a semi-supervised approach where a partial amount of the training data is labelled and weights are learnt as parameters for the level of segmentation allowed by the model. In fact, with just a 100 instances of labelled segmentations, F-Measure improves from 0.61 to 0.65 for English, and from 0.49 to 0.52 for Finnish, with a continued upward trend as more labelled data is added. With a maximum of 10,000 instances in the training data, the F-Measure reaches 0.73 and 0.60 for English and Finnish, respectively. This is the best performing technique for the 2010 Morpho Challenge competition across all languages. However, the focus of Morfessor is mainly on segmentation of words rather than learning labels. In further work, the mapping between segments and morpheme labels was extracted. Once the words in the test data were segmented, the morphemes were assigned the most common label or label sequence. This rather simple approach achieved an F-Measure of 0.77 for English and 0.70 for Finnish in the labelling task.

Durrett and DeNero (2013) use the paradigms found in Wiktionary[10], a crowd-sourced lexical resource that includes inflectional tables for many lexical items in several languages. The technique first looks at rule extraction on the basis of the orthographic changes that take place in the inflection tables. A log-linear model is then used to place a conditional distribution over all valid rules. The model learnt is limited to a particular part-of-speech at a given time, and evaluated on German nouns and Spanish verbs. The aim of the model is to then predict or fill in inflection tables for unseen words. So, rather than learning labels for words, the system learns to generate words on the basis of the paradigm and template that would have been modelled. The reliance on edit distance might not be very successful on a root-and-pattern system, where stem variation impacts the cost of the distance function and, as a result, the extracted rules. Probably for such a system to be successfully applied, the training data would have to be split into two, one to model the concatenative morphological processes, and the other to model the root-and-pattern processes, just as it was split to model particular part-of-speech categories.

Summary of labelling results

The task of morphological labelling of words is generally dependent on the resources available, since labelled datasets are required which can be used to train models and then test them on unseen data. Interest in this type of task is not so high because generally, rule-based approaches perform very well and several languages would already have morphological analysers in place.

Some of the techniques reviewed focus only on particular morphological aspects, such as plural or past tense, thereby reducing the complexity of the learning problem. The work by Kohonen et al. (2010) is the most exhaustive approach, with the added advantage of utilising the Morpho Challenge datasets. However, in the following section we will review more work carried out in morphological labelling in Semitic Languages.

[10] http://en.wiktionary.org

Table 2.3 Summary of results for labelling techniques reviewed
Type: ‘U’: Unsupervised; ‘LS’: Lightly Supervised; ‘S’: Supervised
Results: ‘A’: Accuracy; ‘F’: F-Measure

Citation	**Type**	**Results**
Van den Bosch and Daelemans (1999)	S	A: 64.6%
Clark (2002, 2007)	S	A: 85.8%
Kohonen et al. (2010)	S	F: 0.77
Durrett and DeNero (2013)	S	A: 94.9%

2.4 Approaches used in Semitic languages

The Maltese language has a strong Arabic component, mainly evident in its grammatical structure, with part of the morphological system being root-and-pattern. It is therefore interesting to see the approaches taken for Semitic languages and the issues that these approaches had to face in computational morphology. Wintner (2014) provides an excellent overview of morphological processing in Semitic languages, identifying a number of challenges. The high number of forms that are related to and derived from a root makes it impractical for a lexicon-based approach and, given the complexity of Semitic languages, it is very difficult to implement a morphological analyser to represent the morphological and orthographic rules of the language. If such a grammar is available, it generally produces a number of analyses for a single word, so the results must be disambiguated. It is thus difficult to find a single approach that provides an adequate solution to this challenge. Nonetheless, rule-based approaches have been attempted for a number of Semitic languages, especially using finite-state transducers (see §2.3.1, page 21).

Kiraz (2000) and others extend the two-level morphology of Koskenniemi (1983) to include more than two levels, to allow the system to represent the non-concatenative morphology in Semitic languages. Although Koskenniemi presented a generic morphological system, it was primarily focused to deal with concatenative morphology. By including multiple tiers of representation, the system allows for multiple lexical representations corresponding to the characteristics of non-concatenative morphology, in this case Semitic languages. In a similar vein, Habash et al. (2005) also used a multitape approach for Modern Standard Arabic and spoken dialects, adding information to model the phonology and the orthography, especially for dealing with the requirements of spoken dialects which are sparse in data and use inconsistent orthography. Amsalu and Gibbon (2006) also use FST for Amharic.

The state of the art in Arabic morphological analysis is the Buckwalter Arabic Morphological Analyser (BAMA) (Buckwalter, 2004) and more recently Standard Arabic Morphological Analyser (SAMA) (Graff et al., 2004), described in Habash (2007); Habash et al. (2012); Wintner (2014). Rather than representing the morphological, phonological and orthographic rules directly, the information is held as a database in a large-scale lexicon of base forms, along with tables for prefixes, stems and suffixes, and a list of compatibility rules which specify the combination of stems and affixes. An efficient engine implements the rules as well as a the lexical lookup. During analysis, all possible splits of a word are explored. One of the main limitations of this analyser is that it cannot generate forms, unlike FSTs, whose grammar specification results in both analysis and generation. BAMA/SAMA is the official morphological analyser used by the Linguistic Data Consortium for the Penn Arabic Treebank (Maamouri et al., 2004), a language resource used by most practitioners interested in Arabic disambiguation and parsing.

Machine learning approaches to Semitic languages tend to be applied to part-of-speech tagging, which, due to the complexity of the language, also contains morphosyntactic features. For instance, the Penn Arabic Treebank has over 2,000 tags, whereas a tagset for English would contain just over 50 tags. The techniques used for part-of-speech tagging are ideal for languages such as Arabic, as they take context into consideration. However, due to the rich morphology, data sparseness has an impact on the statistical learning process, and individual words could have several possible tags. Even the BAMA/SAMA outputs multiple potential tags for a given word. Thus the main problem with Arabic morphology is to disambiguate between the potential tags, and choose the correct one in the actual context of the word.

Habash and Rambow (2005) take this approach by using an existing morphological analyser to produce all potential tags for a word, and then applying classifiers (Support Vector Machine) to choose the best tag. The classifiers are learnt for 10 specific morphological features, providing a confidence value for the different characteristics a feature contains. It also uses Viterbi decoding, a technique which allows context to influence the end result (in this case a two-word window prior to and after the current word being analysed). Different approaches are tested as to how to select this analysis. The best approach is to take the product of the number of classifiers agreeing with the analysis by the weighted agreement of the classification confidence measure of the value that agrees with the analysis.

The best accuracy rates reported in this work for the individual feature classifiers are always over 95%, with the lowest being 95.5% for the POS feature and the highest 99.9% for the conjunction and particle features. The accuracy rate of the overall performance of the system is at best 97.6%, the highest accuracy rate achieved over previous work. One of the main disadvantages of this technique is that it is highly dependent on a morphological analyser and a large collection of manually annotated words with morphological features. In fact the same technique was applied to Egyptian Arabic once such resources became available for the language (Habash et al., 2013). These systems were improved in terms of performance, portability and robustness, and an online demo for Arabic and Egyptian Arabic is available online (MADAMIRA Pasha et al. (2014)). Lembersky et al. (2014) use a similar approach for morphological disambiguation in Hebrew, using a combination of classifiers to rank the analysis produced by the MILA morphological analyser for Hebrew (Itai and Wintner, 2008), but report lower accuracy rates (84%), primarily due to the size of the training data.

Marsi et al. (2006) and, more recently, Van den Bosch et al. (2007) apply the Memory-based Learning approach to Arabic (see §2.3.2.5, page 40 for Dutch in Van den Bosch and Daelemans (1999)), integrating morphological analysis and part-of-speech tagging. The algorithm here uses a slightly more complex distance function in order to learn which instances in the model are the closest neighbours to a new unseen instance. This is determined according to the similarity of the pairs based on their conditional probabilities that their features belong to the same class. The k nearest neighbours are found, and a new instance can be classified accordingly. This is a particularly advantageous approach for Arabic, where a word has an average of 6.8 correct analyses, and finding k neighbours allows the system to provide more than one analysis[11]. The evaluation again uses a 10-fold cross validation setup, and a further held out test set. The analysis of the results focuses primarily on the portion of words in the held out test set that were not part of the training set, and therefore completely unseen by the model. In this portion of the data, the system achieves at best an F-Measure of 0.47, which is a fair result, given the complexity of the task and the fact that the metrics take into consideration that the analyser must return all the correct possible analyses of a word.

Stallard et al. (2012) compare unsupervised to supervised techniques for Levantine Arabic applied to machine translation, showing that unsupervised techniques achieve similar or better gains than supervised techniques. The unsupervised technique proposed uses the segmentation process by Lee et al. (2011) discussed above, and then applies maximum

[11] This was one of the drawbacks of the system described in Van den Bosch and Daelemans (1999).

marginal decoding. Maximum marginal decoding takes the possible word segmentations, marginalizing out the irrelevant detail and returns the segmentation that occurs most frequently (Johnson and Goldwater, 2009). The segmentation output produces better F-Scores than Poon et al. (2009) and Morfessor, as well as an improved result on the machine translation task.

Snyder and Barzilay (2008) use a hierarchical Bayesian model on multilingual data as an approach to unsupervised morphology learning, by aligning morpheme patterns across languages, using Hebrew, Arabic, Aramaic and English. The underlying assumption of this work is that there is information related to morphological analysis that is present in the structural commonality of the different languages, and what might be ambiguous in one language, can be clearly marked in another language. A probabilistic model is built on the basis of parallel aligned texts. For these experiments, the authors took the Hebrew bible and its translations and aligned the texts using Giza++. The phrases that occurred at least five times were kept, thus avoiding noisy translations. A monolingual model was built for each language and used as the baseline for morphological segmentation. Moreover, a more sophisticated cross-language model was also built that included character-to-character phonetic correspondences. The cross-language model performed best, increasing F-Measure by 10% and 4% for the Hebrew/Arabic pair over the monolingual model. The cross-language models without the phonetic correspondences performed only slightly better than the monolingual model.

Due to the intrinsic differences in the problem of computational morphology between Semitic and English/Romance languages, it is difficult to set a direct comparison in results. Our interest is more in the types of approaches taken, and particularly, in seeing morphological labelling as a classification problem. Modelling different classifiers for specific morphological properties can be the appropriate approach for Maltese, since it allows the flexibility to focus on those properties where data is available.

2.5 Evaluation

As we have already highlighted, evaluation can be quite problematic in this type of task. To begin with, the task of what to consider a correct word segmentation can be disputed, with several cases highlighted by Goldsmith

> consider the pair abolition and abolish. The words are clearly related, and abolition clearly has a suffix; but does it have the suffix -ion, -tion, or -ition, and does abolish have the suffix -ish, or -sh? It is hard to say.

(Goldsmith, 2001)

To counteract this difficulty, in the manual evaluation Goldsmith decided to have three different categories of incorrect segmentations, which were classified as *wrong* (incorrect analysis), *failed to analyse* (no analysis given), and *spurious analysis* (words not morphologically complex, but analysed as containing a suffix). Of course, this discussion becomes superfluous when using an already segmented dataset for training and testing, as is the case in the Morpho Challenge competition. But it just goes on to highlight the problematic views and decisions that have to be made when a segmented wordlist is not available. In fact, Kurimo et al. (2010) also raise the issue of how to evaluate ambiguous segmentations since words are usually analysed outside of their context. Since Morpho Challenge uses the segmentation and labelling tasks as a foundation for other applications, such as information retrieval and machine translation, they suggest that the segmentation and labelling could be done in context so as to provide better results upstream to these applications.

In the case of Morpho Challenge and similar works where a morphological analyser is available to segment and label the data, evaluations of techniques are generally carried out on withheld data — the test dataset. This dataset would be segmented and labelled by the standard tool for the language, and the technique can be measured for performance against the output of the analyser. In this case, the evaluation is clear-cut, and the most used metrics are accuracy, precision, recall and F-Measure. In order to explain each metric, we will use $truePositives$ to indicate positive examples classified as positive, $falsePositives$ as negative examples classified as positive, $trueNegatives$ are negative examples classified as negative, and $falseNegatives$ are positive examples classified as negative.

Accuracy is the percentage of correct classifications amongst all the tested instances.

$$Accuracy = \frac{truePositives + trueNegatives}{allInstances}$$

Precision measures the correctly classified instances from the instances predicted as positive, as a measure of the quality of the predicted classification.

$$Precision = \frac{truePositives}{truePositives + falsePositives}$$

Recall measures correctly classified instances from all the positive instances present in the testing data, as a measure of quantity of instances being correctly classified.

$$Recall = \frac{truePositives}{truePositives + falseNegatives}$$

F-measure is a metric that uses precision and recall together with a weight α. The role of this weight is to assign relative importance to either precision or recall.

$$F^{\alpha} = \frac{(1+\alpha^2) \cdot (Precision \cdot Recall)}{(\alpha^2 \cdot Precision + Recall)}$$

These metrics are used in evaluating several machine learning and natural language processing techniques. The segmentation task in Morpho Challenge is evaluated by taking a random number of word pairs that have at least one morpheme in common. Precision is calculated as the portion of morpheme-sharing word pairs in the predicted sample in common to the gold standard, and recall is calculated as the portion of morpheme-sharing word pairs in the gold standard sample that is also present in the predicted output.

One of the main problems with this approach is that highlighted by Spiegler and Monson (2010) since it artificially boosts recall when a technique provides several alternative analyses to a word. Spiegler and Monson propose a new evaluation metric — Evaluation Metric for Morphological Analysis (EMMA) — which uses a graph-based assignment algorithm to match the predicted morphemes with those in the gold standard, with an emphasis for unsupervised learning techniques since they do not have access to linguistically motivated morpheme labels. EMMA compares the predicted analyses with those in the gold standard and measures the degree to which the predicted analyses approximate an isomorphism of the gold standard analyses. In order to deal with ambiguous words, EMMA also expects the gold standard to contain a set of analyses for such words, and similarly expects a set of words in the predicted analyses. Among its advantages, EMMA offers an evaluation metric that specifically covers major morphological phenomena, also in terms of how the segmentation task correlates to the natural language processing task. In 2010, Morpho Challenge evaluated all the submitted techniques using both F-Measure and EMMA so as to allow comparability to previous submission as well as to take stock of the challenges raised in the evaluation task.

Machine learning tasks tend to also be evaluated using a N-fold cross validation system. This is carried out by splitting the training data into N equal partitions (commonly 10, thus referred to as 10-fold cross validation) and each partition is split into 90% as training data and 10% as testing data. The technique is iteratively trained and tested, each time adding a fold to its training and testing data, measuring the correctness in prediction with the increase of data as each fold is added. In the end, this is equivalent to using 90% as training data and 10% as testing data, but with the exception of seeing the actual improvement as the amount of data is increased. Accuracy, precision, recall and F-Measure are still used as the evaluation metrics in each iteration, with the final results being the mean values of these metrics.

2.6 Summary and proposed approach

This chapter looked at several aspects of morphology, starting from a short review of language acquisition in children. Psycholinguistic views offer different models as to how language is represented and processed in the mental lexicon, with a connectionist view (Rumelhart and McClelland, 1986) versus a symbolic, rule-based view (Pinker and Prince, 1988). The different models rely on evidence from observations of language acquisition in children, and then attempt to produce a computational model which reflects the same learning process in children. Baayen (2007) proposed that there is a balance between storage and computation and that the brain is more likely to use both aspects rather than one model/system exclusively. The analysis of computational approaches to morphological analysis looks at both rule-based as well as data-driven techniques. Rule-based systems achieve very good results, but require strong linguistic knowledge which needs to be encoded, and tend not to generalise over new words. Data-driven techniques were split according to task: segmentation, clustering and labelling. The majority of the segmentation tasks relied on statistical techniques to determine stems and affixes such as Keshava and Pitler (2006), Dasgupta and Ng (2007) and Creutz and Lagus (2005, 2007). A challenging aspect for the segmentation task was also the evaluation of the proposed segments, discussed in detail by Goldsmith (2001) and also evident through the evaluation process of the Morpho Challenge and other works.

Clustering of morphologically related word-pairs also relied on statistical techniques but added further information to determine relations between words by looking at the orthographic and semantic similarity of words, such as the work described by Schone and Jurafsky (2000, 2001) and Baroni et al. (2002). The approaches to the morphological labelling of word segments generally use machine learning techniques that build models representative of labelled, training data. The models can then be employed on unseen data and therefore the machinery learnt can generalise to new words. Machine learning approaches are also used in Semitic languages, but with the purpose of disambiguating between possible parts-of-speech tags for a single word. Approaches such as Habash and Rambow (2005) for Arabic, and Lembersky et al. (2014) for Hebrew employ classification techniques so that the most appropriate tag can be selected.

This research sought to look at the three different tasks for computational morphology in Maltese. The segmentation task was based on the techniques described by Keshava and Pitler (2006) and Dasgupta and Ng (2007), whilst the clustering techniques were similarly based on Schone and Jurafsky (2000, 2001) and Baroni et al. (2002). The labelling task took the view of seeing morphological properties as possible features that can be modelled and classified. So rather than using classification to disambiguate POS tags following (Habash and Rambow (2005), Lembersky et al. (2014)), the classifiers learn the actual morphological properties of words. The approaches taken for each of these three tasks and the results achieved will be described in the following chapters.

Chapter 3

Segmenting and Clustering Morphologically Related Words

3.1 Introduction

The clustering of morphologically related words is an interesting problem which could provide information for the bootstrapping process of a morphological analyser for Maltese. This chapter focuses on the experiments carried out in the segmentation and clustering of morphologically related words in Maltese using unsupervised techniques. The approach uses a variety of relatedness heuristics based on orthographic and semantic information, combining ideas which have been used successfully in other languages to pair up morphologically related words, and at the same time extending them to consider the clustering of as many morphologically related words as possible.

One of the main research questions explored is whether the techniques employed perform differently on Romance-origin words than on Semitic-origin words. The hybridity of the morphology in the Maltese language makes it ideal to investigate such a question. An indirect benefit from this research is the provision of further information about the historical origin (Semitic/root-based vs Romance/stem-based) of words, possibly using the results to distinguish between the two sets of words.

The second goal of this work is to explore possible evaluation strategies to verify the output of the clustering techniques. While developments in this field have often been aided by the existence of gold-standard lexical resources (e.g. as used for the Morpho Challenge evaluations), no such resources currently exist for Maltese. Indeed, the evaluation method proposed here is intended to partially address this by adopting a crowd-sourcing evaluation strategy. The evaluation uses both linguistic experts as well as non-experts, thereby also laying the grounds for the development of a future gold standard of morphologically related word clusters.

The clustering technique relies on the segmentation of words to identify stems and use them as an initial basis for relating words. Since linguistic tools, such as a lemmatiser or segmenter, are not available for Maltese, the technique follows a similar process to that of Keshava and Pitler (2006) and Dasgupta and Ng (2007), using transitional probabilities to discover affixes in an unsupervised manner. Words are then segmented according to a list of ranked affixes. However, since the purpose of the segmentation is as a stepping stone towards the clustering of words, a certain level of uncertainty is allowed in the segmentation results by allowing the system to suggest more than one possible segmentation. We posit that a segmentation can be reinforced through other processes, so the correct segmentation can be decided at a later stage.

The clustering of morphologically related words uses the output of the segmentation process as the basis for grouping words together. Since stem variation is common in Maltese, it is important to use semantic and orthographic similarity, similar to the techniques used by Schone and Jurafsky (2000, 2001) and Baroni et al. (2002), as part of the heuristics to discern which words should be grouped together, and which groups should be discarded. A portion of the output clusters are evaluated through a crowd-sourcing approach using native Maltese speakers as non-experts, and a smaller number of clusters is evaluated by three linguists. A discussion is then presented on the basis of the results of the evaluation in terms of the quality of the resulting clusters. Through the evaluation done by the expert group, the evaluation also considers the differences in clusters of words of Semitic and of Romance origin.

The rest of the chapter is structured as follows. First we discuss the segmentation technique in §3.2, describing the data, the technique of Dasgupta and Ng (2007); Keshava and Pitler (2006) and the approach to the segmentation of words. The way clusters were formed is described in §3.3, outlining the use of both orthographic and semantic similarity and how clusters are ranked and merged together or discarded. The crowd-sourcing and expert evaluations are described in §3.4, with an in-depth analysis of the results given in §3.5. Apart from the quantitative and qualitative analysis of the clusters, we also analyse and obtain insight from the evaluation results pertaining to the questions on the hybrid morphology of the Maltese language in §3.5.4. The majority of the techniques used for the segmentation and clustering of words were mainly applied on Anglo-Saxon and Romance languages. Although such languages can contain non-concatenative morphological aspects, these are not as prevalent as in Maltese. Comparing clusters of non-concatenative and concatenative morphologically related words side-by-side shows that although the techniques do perform better overall on concatenative words, but both processes have their pitfalls and the techniques can only serve as part of a bootstrapping process in languages where the computational resources are few.

3.2 Segmentation

The segmentation of words is a preprocessing step in computational morphological analysis, whereby a word is split into a stem and affixes (Hammarström and Borin, 2011). The identification of the stem is important to group morphologically related words together. It is also possible to list all possible inflections of a stem, grouping words into paradigms and then providing functional labels for each slot within the paradigm, since a segmenter or lemmatiser for Maltese is currently not available. This research began by exploring possible ways of segmenting words automatically through unsupervised techniques, using the Maltese Language Resource Server (MLRS) corpus[1] (Gatt and Čéplö, 2013) as a starting point. By employing statistical techniques, a list of affixes was extracted and used as the basis for the segmentation of words, which could in turn feed into the clustering of morphologically related words.

3.2.1 Data preparation

The starting point for the segmentation process was the MLRS corpus, which at the time contained over 120 million tokens. The corpus is a collection of texts from different online sources and represents various genres, from academic and literary articles to blogs and newspaper articles. A wordlist was extracted from the corpus, which included frequency counts for each word. Since the corpus was opportunistically built from articles and documents crawled through the Internet, the quality, especially in terms of orthography, was not exceptionally high, with several mistakes in the texts. Another issue of Maltese orthography is the spelling of loan words (especially from English). In an effort to increase the quality, a section of the corpus was spell-checked manually in a separate project. The emphasis of this task was to correct orthographic errors and to identify and mark foreign words. Using the resulting database, these corrections were then propagated to the rest of the corpus. Apart from these corrections, a further 123 rules were written programmatically to make some further corrections, mainly aimed at eliminating incorrect

[1] http://mlrs.research.um.edu.mt

words from the wordlist. Table 3.1 provides the statistics of the number of distinct word types split into different categories according to the correction process. The corrections resulting from the spell-checking project are marked as 'SC', whilst those resulting from the rules written specifically for this work are marked as 'RL'. The categories used are indicative of the process applied. By applying the spell-checking corrections, a number of words were marked as foreign (6), or had different corrections by different annotators (9), so these words were discarded. Words which were corrected were included (B). The remaining words were initially considered as 'Maltese' (A), and eliminated from this category when a word matched a specific rule. For instance, a set of non-Maltese characters were identified and if a word contained such a character it would be placed into category 7. Other simple rules, such as, those applied to words with triple or more letters, would place words into category 5 (e.g.: "sooooo" - multiple 'o's placed for over-emphasis). Some of the rules were aimed at identifying possible errors but without a means of automatically verifying whether these are actually errors or not. Depending on the rules, some words were marked as possible errors but retained (category D), and some words were marked as probable errors and ignored (marked as category 10).

Since the approach uses a completely unsupervised technique, it is important to restrict the wordlist to correctly spelled words. The more errors present in the wordlist, the more these would be propagated in the results. By excluding known and probable errors, function words, numbers, punctuation marks, proper nouns, determiners, foreign words and those words with a token count of less than 10, the resulting wordlist used had 67,434 word types.

Table 3.1 Corpus statistics indicating the number of word types and how these are categorised.

Description	**No. of Word Types**	**Percentage**
Number of distinct word types	529,056	100%
(A) Marked as Maltese	111,870	21%
(B) Manually corrected (SC)	139,961	27%
(C) Rule corrected (RL)	196	0%
(D) Included — requiring manual checking (RL)	48,597	9%
Sub-total	300,624	**57%**
Having Token Count > 9	**67,434**	13%
Categories for discarded words:		(43%)
(1) Function words	173	0%
(2) Proper nouns	93,492	18%
(3) Determiners	195	0%
(4) Punctuations	28	0%
(5) Rule marked errors (RL)	12,076	2%
(6) Marked as foreign (SC)	61,392	12%
(7) Foreign (RL — non-Maltese characters)	5,128	1%
(8) Rule marked foreign (RL)	64	0%
(9) Ambiguous corrections (SC)	49,947	9%
(10) Marked as ignore (RL)	5,937	1%

3.2.2 Extracting affixes

The first step towards segmentation is identifying the most probable prefixes and suffixes which can then be used to segment words. Keshava and Pitler (2006) proposed an unsupervised technique that uses transitional probabilities to identify prefixes and suffixes automatically from a list of words. Words are represented into a Trie data structure — a type of programming structure which can hold data in an efficient manner since strings with multiple overlapping substrings are stored without excessive redundancies. A Trie has a symbolic *root* node which is used to access the data structure and the remaining nodes, each representing a single character. Each node can have several children, with the constraint that each child is unique (no two children can represent the same character). For any node n, representing the character c, the children of n represent the attested subsequent characters in the wordlist. Thus all words starting with 'a' will share this node. A small sample is shown in fig. 3.1, showing how a number of word forms from the word *aċċetta* 'to accept' would be stored in the trie data structure and how a number of nodes are shared among the different words. Each node also holds information about the token count and whether it represents the end of a word (marked in a blue double-line in fig. 3.1).

The technique takes a word and iterates through every character, checking whether this can be considered as a potential boundary between a stem and a suffix. In order to be able to generalise the description, a word is represented as $\alpha AB\beta$, where uppercase A, B represent single characters, the Greek letters α, β represent character sequences, with the possible boundary being examined between A and B. Hence, αA is the potential stem, and $B\beta$ the potential suffix. Table 3.2 shows how the word *aċċettajtx* 'I/You did not accept' is represented as $\alpha AB\beta$ as the boundary check iterates through every character of the word. The sequence $B\beta$ is given a score, reflecting the likelihood of it being a suffix, on the basis that the following conditions are satisfied:

1. αA is itself a word type in the wordlist — In the Trie this would be true if A is a blue double-lined node.

2. $P(A|\alpha) \approx 1$ — The number of words starting with αA is approximately the same as the number of words starting with α. Intuitively this test is again checking the potential stem and that there is a low branching factor at this boundary.

3. $P(\beta|\alpha A) < 1$ — The number of words starting with αA is higher than the number of words starting with $\alpha A\beta$, indicating that substantial branching occurs at the point in the word where the possible suffix begins.

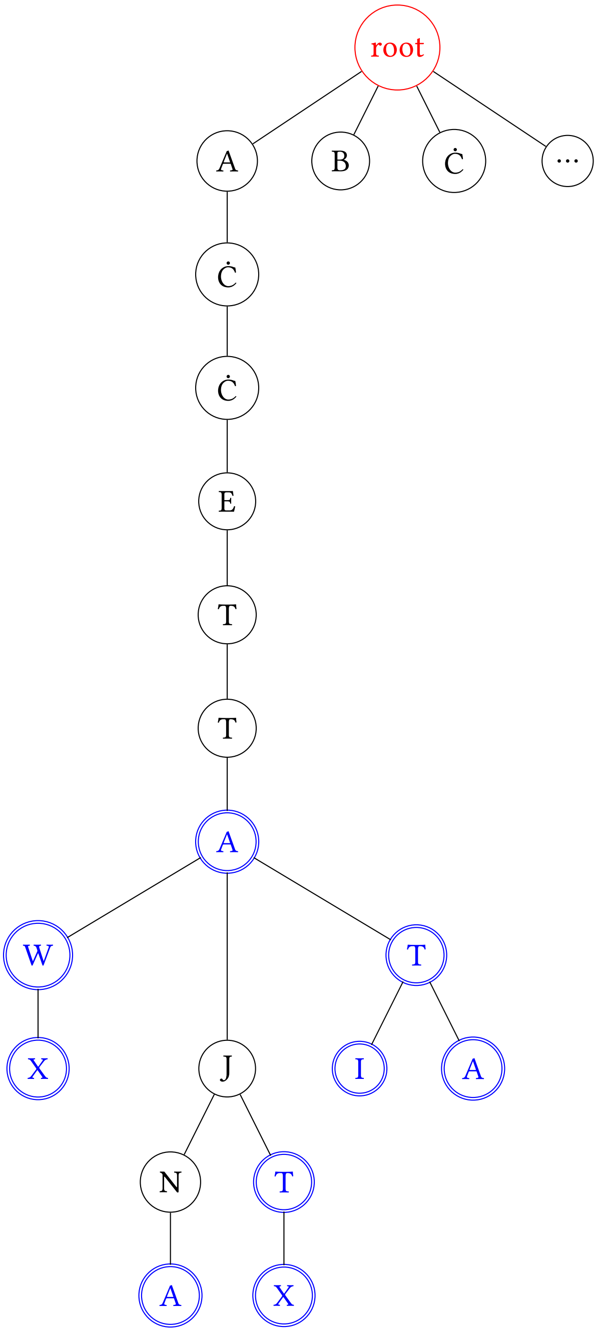


Fig. 3.1 An example of a Trie data structure for the dictionary entries: *aċċetta, aċċettaw, aċċettawx, aċċettajna, aċċettajt, aċċettajtx, aċċettat, aċċettati, aċċettata* — some of the inflective forms of the word *aċċetta* ‘to accept’. The red node is the root node, the black nodes intermediary nodes and traversing any path from the red node to any of the blue double-lined nodes forms a word.

Table 3.2 Iteration for the word *aċċettajtx*, the representation of $\alpha AB\beta$, and the results of the three tests.

Representation	α	A	B	β	Test1	Test2	Test3
a-ċċettajtx	–	a	ċ	ċettajtx	FALSE	TRUE	FALSE
aċ-ċettajtx	a	ċ	ċ	ettajtx	FALSE	TRUE	FALSE
aċċ-ettajtx	aċ	ċ	e	ttajtx	FALSE	TRUE	FALSE
aċċe-ttajtx	aċċ	e	t	tajtx	FALSE	TRUE	FALSE
aċċet-tajtx	aċċe	t	t	ajtx	FALSE	TRUE	FALSE
aċċett-ajtx	aċċet	t	a	jtx	FALSE	TRUE	FALSE
aċċetta-jtx	aċċett	a	j	tx	**TRUE**	**TRUE**	**TRUE**
aċċettaj-tx	aċċetta	j	t	x	FALSE	FALSE	TRUE
aċċettajt-x	aċċettaj	t	x	–	TRUE	FALSE	TRUE

The technique uses these three tests to discover potential boundaries between stems and suffixes. Table 3.2 provides an overview of how the word *aċċettajtx* was processed, showing the results of the three tests. In this example, there is only one boundary where all three tests give an affirmative answer - *aċċetta-jtx*. However, it is also possible that multiple boundaries produce an affirmative result for a word. In this example, the boundary *aċċettajt-x* could also be considered a good boundary. However, in this case it did not pass the second test since there is substantial branching off at the *j* node, due to words such as *aċċettaj-na*. This does not impact the technique negatively since the suffix *x* is encountered several other times when the boundary check would pass all three tests.

For every boundary tested, the possible suffix $B\beta$ is scored according to the results of the three tests: the score is incremented by 19 if all three tests pass, and decremented by 1 if any of the three tests fail. So in the example in table 3.2, the score for the substring *-jtx* is incremented by 19, and the scores of all the other substring are decremented by 1. Each possible suffix has a score associated with it and is updated every time it is encountered in a word. The scores are not arbitrary, but are purposely selected to ensure that any string $B\beta$ has a positive final score only if it passes the three tests at least 5% of the times it was attested ($\frac{1}{1+19} = 0.05$). 1 and 19 can be substituted with any numbers that satisfies ($\frac{y}{x+y} = 0.05$). The scoring mechanism gives a system of ranking potential suffixes, with the most likely suffixes having the highest scores.

This technique was implemented and applied to the MLRS wordlist, resulting in a large list of potential suffixes ranked by their relative scores. A similar process was carried out to extract the prefixes. This was done using exactly the same technique but representing and storing the words back-to-front, and processing them in reverse order. The technique produced a list of prefixes and suffixes and their relative scores through which the affixes are ranked. The higher the score, the more likely that a particular string is indeed an affix. The list produced also contains strings which are not affixes, and therefore a decision must be made to determine the cutoff point. The top 400 ranked suffixes and 200 ranked prefixes were used to segment words. This represented approximately the top 10% of the potential affixes. The cutoff was rather lenient in order to allow composite suffixes into the set, since these are less frequent and therefore rank lower than more frequent suffixes. The technique does not distinguish simple suffixes from composite affixes. Thus, in the example below, both *-ha* and *-hielha* are processed as whole suffixes and no link is made between the presence of *-ha* in both words.

(3.1)	*aċċetta*	*-ha*
	accept.Perf.3SgM	-3SgF-Acc
	he accepted her	

(3.2)	*aċċetta*	*-hie*	*-lha*
	accept.Perf.3SgM	-3SgM-Acc	-3SgF-Dat
	he accepted it for her		

3.2.3 Composite suffixes

The resulting suffix list included several composite suffixes. The ability to segment composite suffixes could be beneficial as this would split a complex suffix into morphemes. Some experiments were carried out, in a similar vein to the strategies used by Keshava and Pitler (2006) and Dasgupta and Ng (2007), to analyse whether splitting composite suffixes automatically is feasible.

Keshava and Pitler's technique is rather simple and uses the ranked list of suffixes. Iterating over the ranked list, when a suffix S_x can be decomposed into two better-ranked suffixes S_i and S_j, S_x is removed from the suffix list since it is represented by the two composites S_i and S_j. This means that the whole suffix xy is removed and only the composite suffixes are kept as two separate suffixes so that, during the segmentation process, the suffix xy is correctly presented as $x + y$. The limitation of this technique is that it only considers a maximum of two composites within a suffix. This technique was applied to the top 400 ranking suffixes in Maltese and resulted in the removal of 228 suffixes, leaving 172 suffixes. However, further analysis of the results showed that this technique is not very reliable for Maltese, as can be seen in table 3.3 below. From the 172 suffixes which made it to the final list, only 60% were correct, with the rest which should have been segmented. On the other hand, a number of suffixes removed from the list (segmented) were either segmented at the wrong boundary (8%) or should have been left whole (23%). The overall accuracy of the technique is rather low (52%) for Maltese and Keshava and Pitler do not provide an analysis of the composite suffixes, so it is not possible to compare the results. However, it is clear that this technique is not well suited for Maltese.

Table 3.3 An overview of the errors in the composite suffix procedure using the technique based on Keshava and Pitler (2006)

Description	Total	Percentage
Total Suffixes	400	
Correct Output (Accuracy)	207	52%
Incorrect:	193	48%
– should have been split	68	17%
– split at the wrong boundary	32	8 %
– should have been left whole	93	23%
Final affix list:	172	
– of which are correct	104	60%

One of the drawbacks of the technique above is that it only caters for one split in a suffix. This restriction is not ideal since Maltese has several suffixes whichcan be split into more than two parts (e.g *-w-hom-x* -PL.NOM-3PL.ACC-NEG). In this analysis, it is crucial to balance between not segmenting at all and over-segmenting, either of which should ideally be avoided. Another strategy for segmenting composite suffixes is that described in Dasgupta and Ng (2007). A suffix $\alpha\beta$ can be considered a composite of two suffixes α and β if the inner part of the suffix (α) is combined with a similar set of words as the whole suffix ($\alpha\beta$). The similarity between the whole and the composite suffix is measured by comparing the number of distinct words they combine with. Thus,

$$Similarity(\alpha\beta, \alpha) = \Pr\left(\alpha \mid \alpha\beta\right) = \frac{|W'|}{|W|} \tag{3.3}$$

where $|W'|$ is the number of distinct words that combine with both $\alpha\beta$ and α (thus taking the intersection of the two sets of words), and $|W|$ is the number of distinct words that combine with $\alpha\beta$. In Dasgupta and Ng, a composite is considered to be correct if the similarity obtained is greater than 0.6. However, it is not clear whether this threshold is arbitrary or whether the value signifies some specific reasoning. Again, it is desirable to be able to consider more that one segmentation in a composite suffix. Thus we extended the principle to consider all possible segmentations, such that if we were considering suffix $\alpha\beta\gamma$, then we would have

$$\Pr\left(\alpha \mid \alpha\beta\right) * \Pr\left(\beta \mid \beta\gamma\right) \tag{3.4}$$

The way that the extension is implemented is that all possible splits are considered and the similarity is worked out for each possibility. This means that given the suffix xyz, the composites considered are $x - yz$, $xy - z$ and $x - y - z$. The most probable split is suggested as being the most likely segmentation of the composite suffix. In practice, this extension did not really impact the results of the segmentation, since larger segments are usually more probable than smaller segments. Different thresholds were tested since the 0.6 threshold used in Dasgupta and Ng (2007) resulted in only 5 composite suffixes. The results are indicated in table 3.4 below.

Table 3.4 The accuracy in the segmentation of the top 400 ranked suffixes using different thresholds, extending the technique of Dasgupta and Ng (2007).

Threshold	0.6	0.5	**0.4**	0.3	0.2	0.1	0.05
Correct suffixes	271	276	288	288	291	274	248
Accuracy	67.7%	69%	**72%**	72%	72.7%	69%	62%
Composite suffixes	5	10	32	42	57	118	188
Correct composites	0	5	22	27	36	58	80
Correct composites	0%	50%	**69%**	64%	63%	49%	42%

The extension over Keshava and Pitler is worth considering since it raised accuracy from 52% to 72%, with the best threshold being at 0.4. Still, more than one fourth of the suffixes were incorrectly segmented. However, the segmentation of suffixes into their composites is not an essential task for clustering morphologically related words. It is more important to identify the stem of a word correctly so that morphologically related words can be grouped together. The importance of composite suffixes in relation to morphological analysis is more important for the labelling task since each composite, when correctly equivalent to a morpheme, can provide an important relation to the relative morphosyntactic label of a word. The task of marking words with their morphosyntactic labels is described later on in chapter 4.

3.2.4 Segmenting words

The main focus of the segmentation task is to aid the morphological clustering of words. The automatically identified affixes can be used to remove affixes from words so as to identify the stem. Although considerable stem variation occurs in Maltese, it is somewhat easier to cluster morphologically related words together on the basis of the stem rather than the whole word. For instance, the pairing up of the words *aċċettaha* and *aċċettahielha* would be simpler once the stem *aċċetta* is identified. In this way, the stem is acting as a common denominator between different words, allowing for the possibility to group words together on the basis of the shared orthography over the stem.

The segmentation process used the top 10% of the ranked suffixes and prefixes identified previously. Since the affix list is automatically retrieved through a data-driven approach, it contains some incorrect affixes. For example, consider the incorrectly identified suffix *-kament*[2] in words such as *politikament* 'politically' and *bażikament* 'basically'. The difference between these two examples is that the stem **politi* is not in the wordlist, whilst the stem *bażi* 'basis' is. For this reason, the suffix *-kament* ends up in the final suffix list even though it is incorrect. If segmentation of words is restricted to where the stem is identified as a valid word in the corpus, it avoids segmenting *politi-kament* incorrectly as the stem *politi* is not found, allowing only the correct segmentation *politika-ment*. However, there is no precise way of predicting that *bażika-ment* is the correct segmentation and *bażi-kament* is the incorrect one as both suffixes and both stems are valid according to the data present. *Bażika* 'basic.3SgF' is an adjective derived from the noun *bażi*, with the latter stem being the most frequent one.

Complications may also arise due to homographs in the wordlist. For instance, the stem *park* is both a part of the verb meaning 'to park' and the noun meaning 'public space'. Thus, for *ipparkja* 'he parked', the method finds two potential segmentations: *i-pparkja* and *ip-park-ja*. A more subtle cause is the presence of consonant gemination in Maltese as part of the derivational process, for example *verifika* 'verification' → *v-verifika* 'he verified', and, if it is preceded by a word ending in a consonant, it takes the epenthetic vowel *i* (i.e. *i-vverifika*). This is only typical in loan verbs. Another problem is over-segmentation, resulting in a stem that belongs to a different (homonymous) lexeme — e.g. *spiċċa* 'to finish' is segmented as *s-piċċ-a*, with the resulting incorrect stem *piċċ* 'pitch'.

As these examples illustrate, the challenge in segmentation is not limited to how a word should be split, but also to determine when a word should *not* be segmented. This

[2] The correct suffix *-ment* is also in the list and is ranked higher. *-ment* is equivalent to the adverbial suffix *-ally* or *-ment* in English.

also arises from the fact that the probabilistic method will tend to favour smaller numbers of segments rather than multiple segments. In order not to be forced to chose a potentially worse segmentation over another one, all possible valid segmentations are considered as input to the clustering process.

A valid segmentation is one in which the affixes are in the top 10% of the ranked lists, and the resulting stem is present in the wordlist. This strategy might leave words unsegmented, especially in cases where stem variation occurs. However this choice is justified since the final aim is to cluster morphologically related words together. Each word was therefore processed through the segmentation technique, and presented as a whole word unsegmented together with any additional valid segmentations to the clustering technique described below.

3.3 Clustering technique

The clustering of morphologically related words looked at automatic techniques through which words could be grouped together on the basis of orthographic and semantic similarities. The aim of this work was to produce word groupings (clusters) that are morphologically related. Each cluster is headed by a word, which ideally, represents the overall concept of the words found within the group, and is referred to as the head or head word of a cluster. However, since this is an automatic process, it does not necessarily mean that it captures the full conceptual meaning of a cluster, especially since no distinction was made between inflectional and derivational words. Such words might end up in the same cluster and the head word might not necessarily represent the concepts present in the cluster.

3.3.1 Segmentation-based clustering

Initially words were grouped together on the basis of common stems. Given a word W, and all its possible segmentations $seg_1 \ldots seg_n$ (including the unsegmented word), we used every stem s_i from the segmentations as a cluster head and grouped together all words incorporating that stem. Thus, from *ivverifika* and its possible segmentations *i-vverifika* and *iv-verifika*, the system derives three clusters headed by each stem. *Ivverifika* is itself a member of every cluster, together with any other word types that have these stems and additional affixes (e.g. *ivverifikajt* ‘I verified’). Table 3.5 shows the words clustered together with these three stems. All three clusters mainly included inflective forms of the verb *vverifika*, whilst the cluster for *verifika* included also some derivational forms (e.g. *verifikar*, verbal noun; *verifikaturi*, noun).

Table 3.5 Initial clusters for the stems *verifika, vverifika and ivverifika*

Head word	Initial word clusters	Size
verifika	ivverifika, ivverifikajna, ivverifikajt, ivverifikar, ivverifikat, ivverifikata, ivverifikaw, verifikar, verifikat, verifikata, verifikaturi, verifikazzjoni, vverifika, vverifikajna, vverifikajt, vverifikat, vverifikata, vverifikaw	18
vverifika	ivverifika, ivverifikajna, ivverifikajt, ivverifikar, ivverifikat, ivverifikata, ivverifikaw, jivverifika, jivverifikahom, jivverifikaw, nivverifika, nivverifikaw, tivverifika, vverifikajna, vverifikajt, vverifikat, vverifikata, vverifikaw	18
ivverifika	ivverifikajna, ivverifikajt, ivverifikar, ivverifikat, ivverifikata, ivverifikaw, jivverifika, jivverifikahom, jivverifikaw, nivverifika, nivverifikaw, tivverifika	12

By allowing the system not to select one definite segmentation for a word, each word can appear in more than one cluster. For instance, *jivverifikaw* was found with the clusters headed by the words *vverifika* and *ivverifika*, but not the cluster *verifika*. This is because the segmentation using the latter as a stem was not possible since the prefix *jiv-* was not in the list of prefixes, whilst *i-* and *ji-* were. Table 3.6 shows the more problematic example of the word *spiċċa* 'he finished' being wrongly segmented as *s-piċċa* and *s-piċċ-a*. The stem *piċċ* (which is a possible spelling of 'pitch/field pitch') is a valid word but with a completely different meaning from *spiċċa*. However, the stem **piċċa* is erroneous due to the presence of noise in the wordlist.

This initial clustering technique produced 21,381 clusters; a number of words were present in multiple clusters, and it was possible for a cluster to be a subset of a larger cluster. The amount of clusters up to this stage was quite large when considering that the starting point was a list of 67,434 words. However, over 10,000 clusters contained just two words, and most of these small clusters were subsets of larger clusters. The largest cluster contained 74 words and, overall, the clusters contained an average of 4 words.

Due to the large number of clusters, the question arose as to whether and how the quality of a cluster could be automatically determined, and how to select those clusters which would best represent a group of morphologically related words. Ideally, clusters should contain as many morphologically related words as possible. The first improvement to the

Table 3.6 Initial clusters for the stems *piċċ, piċċa and spiċċa*

Head word	Initial word clusters	Size
piċċ	piċċa, spiċċa, spiċċat	3
piċċa	piċċaw, spiċċa, spiċċajna, spiċċajniex, spiċċajt, spiċċajtu, spiċċalha, spiċċalhom, spiċċali, spiċċalu, spiċċat, spiċċatilhom, spiċċatlu, spiċċatx, spiċċaw, spiċċawh, spiċċawlhom, spiċċawlu, spiċċax	19
spiċċa	jispiċċa, jispiċċaha, jispiċċalek, jispiċċalha, jispiċċalhom, jispiċċalu, jispiċċaw, jispiċċawh, jispiċċawlhom, jispiċċawlu, jispiċċax, nispiċċa, nispiċċaw, nispiċċawh, nispiċċax, spiċċajna, spiċċajniex, spiċċajt, spiċċajtu, spiċċalha, spiċċalhom, spiċċali, spiċċalu, spiċċat, spiċċatilhom, spiċċatlu, spiċċatx, spiċċaw, spiċċawh, spiċċawlhom, spiċċawlu, spiċċax, tispiċċa, tispiċċalhom, tispiċċalu, tispiċċaw, tispiċċax	37

clusters simply removed clusters which were proper subsets of larger clusters. However, there were two main challenging problems which needed to be addressed in further detail:

Separate Clusters A number of different clusters contained an overlap of morphologically related words or there were separate clusters which were morphologically related. Such clusters should be **merged** together into one cluster. E.g., table 3.5 shows the three initial clusters obtained from processing *ivverifika*. Ideally these three clusters should be merged into a single cluster. This problem is more pressing due to stem variation in Maltese words, resulting in several separate clusters. For instance, for the verb *seraq* 'to steal', the words *jisraq*, *seraq* and *serqu* were all indicated as stems and found in separate clusters due to stem variation. Such clusters should be merged together into a more complete cluster.

Mixed Clusters A number of clusters contain unrelated words, e.g., the cluster for *piċċ* seen in table 3.6. The choice was either to try and remove the unrelated word or completely disregard such a cluster, especially if a large percentage of the words in the cluster are unrelated words.

In order to improve the quality of the clusters, we introduced semantic and orthographic metrics to try and determine automatically whether words in a cluster were morphologically related or not. We also used these measures to compare clusters. This approach is similar to Schone and Jurafsky (2000, 2001) and Baroni et al. (2002). However, rather than limiting the relatedness to a pair of words, we looked at whole clusters of words.

3.3.2 Semantic and orthographic similarity

Latent Semantic Analysis (LSA) is a technique belonging to a much larger family of distributional or vector-space methods (Turney and Pantel, 2010). It analyses the relationship between words by comparing the context in which they appear. The more two words appear in similar contexts, the more semantically related they are. For example, the words *green* and *blue* would be expected to be surrounded by similar words since they are both colours. This is also plausible for morphologically related words, such as *blue* and *bluish*. LSA creates a vector representing the surrounding words for each word present in the corpus. It then gives the semantic similarity of two words by comparing the similarity of their individual vectors. One of the main advantages of using an LSA as a vector-space method is due to its matrix-algebraic characteristics, with the capacity to reduce very high-dimensionality vectors by performing singular vector decomposition. In order to employ an LSA as part of the clustering technique, we chose to use an open-source semantic space library implemented by Jurgens and Stevens (2010).

The semantic space was created using the MLRS corpus with a stoplist consisting of function words, punctuation and some misspelt word types. The resulting semantic space was then used to compute the semantic similarity of word pairs. The LSA returns a value between 0 and 1, where 1 is the limiting case of similarity, implying identity. In fact a value of 1 usually occurs in assessing the similarity of a word with itself. The smaller the value, the less semantic overlap there is between two words.

The purpose of applying semantic values between words is twofold. First, clusters with a high number of unrelated words should be disregarded completely, especially if the words are contained in other clusters. Second, clusters which might be morphologically related but were clustered separately, especially due to stem variation, should be considered for merging. For both cases, we devised a metric that measures a cluster's *semantic cohesiveness*. The intuitive concept behind this metric is that the more semantically related the words within a cluster are, the tighter or more 'compact' the cluster is. Although semantic relatedness does not imply morphological relatedness, the clusters so far were formed through the identification of potential stems. This limits the measuring of semantic relatedness of words to those which have a strong orthographic similarity between them, since this is guaranteed by the stem overlap. Semantic cohesiveness is calculated by taking the standard deviation of the semantic relations between the stem heading a cluster and every word in the cluster.

Let C_i be a cluster headed by stem s_i, and let $Sem_{s_i w_j}$ be the semantic similarity between s_i and $w_j \in C_i$. The cohesiveness, σ_{C_i} is computed as:

$$\sigma_{C_i} = \sqrt{\sum_{w_{1\dots n} \in C_i} (Sem_{s_i w_j} - \mu)^2} \tag{3.5}$$

where μ is the mean pairwise semantic relatedness in C_i. If a cluster contains seven words ($w_{1\dots 7}$), then it will have seven similarity values ($Sem_{s_i w_1} \dots Sem_{s_i w_7}$) associated to each word reflecting the semantic similarity between the stem and the word. The variance and deviation of all the semantic values within one cluster is calculated and the deviation is taken as an indicator of a cluster's semantic cohesiveness. The intuition here is that, the wider the dispersion within the cluster, the less cohesive it is, indicating a weaker similarity between words. Using this metric, clusters such as those in table 3.5 generally score quite high since the words are clearly related to each other. By comparison, the cluster for *piċċ* has a lower score. This can be seen in table 3.7, showing the semantic cohesiveness σ_{C_i} for each of the example clusters illustrated previously.

Table 3.7 Values of the semantic cohesiveness σ_{C_i} for some of the initial clusters indicated by the head word

Head word	σ_{C_i}
verifika	0.1504
vverifika	0.1624
ivverifika	0.2053
piċċ	0.0807
piċċa	0.1376
spiċċa	0.2359

Although the initial clusters were produced on the basis of a common stem, further improvements in the clusters needed to consider the orthographic similarity of words. The Minimum Edit Distance (MED), also known as Levenshtein distance, is a standard cost function which, given two words w_i and w_j, returns the cost required to arrive from one word to an other, calculated on the least possible number of character insertions, deletions and substitutions. If each operation had a cost of 1, the distance between *cat* and *cot* is 1 through substitution. The technique employed a weighted implementation of MED provided in the LingPipe library (Alias-i, 2008), which also allows for negative costs for insertion, deletion and substitution operations, whilst providing a positive cost to the matching operation. Rewarding matches over other operations is a favourable strategy, especially due to stem variation. The negative score from the number of insertions, deletions and substitutions is 'normalised' by the number of matches that two words have. The costs used were set as follows: match 4.0; insertion -6.0; deletion -6.0; substitution -12.0. Substitution in particular was weighted as the sum of the cost of the two operations (insertion and deletion) so as to avoid the unwanted situation where the change of one single character would result in a lower cost but resulting in a completely different meaning. For instance, if substitution is given a lower cost, it will result in *kiser* 'to break' being closer to *kiber* 'to grow' than to *ikser* 'break.IMP'; however, with the proposed costs they would have the same distance.

3.3.3 Merging clusters

The initial clusters contained both overlap in terms of separate clusters with a number of shared and related words as well as separate clusters with different words but morphologically related. The semantic cohesiveness of a cluster can be used as a metric to assess whether two clusters should be merged together. The merging of clusters was carried out in two phases. The first phase iterated through the clusters, creating a ranked list of *potential cluster pairs*. This avoided the problem of merging clusters iteratively in a random order, which may result in C_i being merged with C_j, when a later cluster C_k would have been a better candidate to merge with C_i. The actual merging of clusters was done in a second phase, which used the ranked list of candidate cluster pairs and checked the improvement in the semantic cohesiveness of the new potential cluster prior to merging two clusters. In other words, two clusters were merged only if the semantic cohesiveness of the new cluster is better than the two unmerged clusters.

To create the potential cluster pairs, all clusters were paired up and the semantic cohesiveness for each pair was calculated, providing the possibility to rank the pairs according to the added value that merging two clusters would actually bring. For each candidate clusters C_i and C_j, with stems s_i and s_j respectively, two clusters were considered for merging as cluster C_k taking stem s_i, the ranking of C_k was calculated as follows:

1. **Semantic similarity** $\textsc{Sem}_{s_i s_j} > \alpha$, where s_i is the stem head for C_i, s_j is the stem head for C_j, and α is a predetermined threshold. This means that the semantic simi-

larity of the two stems must be above a certain threshold α, which was empirically determined at 0.4.

2. **Orthographic similarity** $\textsc{Med}_{s_i s_j} >= \beta$, where s_i is the stem head for C_i, s_j is the stem head for C_j, and β is a predetermined threshold. This means that the two stems must share some orthographic similarity, and the threshold β was empirically determined to be 0.
3. **Improvement in the semantic cohesiveness** (semImp) of the merged cluster C_k, where
$$\textsc{semImp}_{C_k} = \sigma_{C_k} - \min(\sigma_{C_i}, \sigma_{C_j}). \tag{3.6}$$
The potential new cluster C_k was considered better when its semantic cohesiveness σC_k is lower than the cohesion for either of the two original clusters. This would indicate that there is an improvement in the new cluster, resulting in lower dispersion.

The technique required the consideration of all three values because the improvement in semantic cohesiveness (semImp) alone was not sufficient to reliably determine whether two clusters should be merged. An 'improvement' in semImp could be registered even when non-related words are introduced through the merging of two clusters. This was primarily due to the fact that some of the inflective variants of a word would have a rather low semantic similarity value, so the overall improvement of the semantic cohesiveness of a merged cluster cannot be considered in isolation. We combine all the three values described above into a single weighted metric, referred to as Combined Value (ComVal):

$$\textsc{ComVal}_{C_k} = \alpha\textsc{Sem}_{s_i s_j} + \beta\textsc{Med}_{s_i s_j} + \gamma\textsc{semImp}_{C_i C_j} \tag{3.7}$$

where the weights were empirically determined as $\alpha = 0.2$, $\beta = 0.2$, and $\gamma = 0.6$.

The first phase produced a list of cluster pairs which were ranked by their relative ComVal. In the second phase, the program iterated through the ranked list to execute the merging of clusters. Since the same cluster appears in the ranked list several times, clusters which were merged were given an integer value to indicate their status. All clusters began with a status of 0 indicating that no operation was so far carried out on that cluster. The status of the two clusters being considered for merging was changed as follows:

1. Both clusters C_i and C_j had a status of 0, indicating that so far neither has been merged — the merging of the two clusters was carried out, and the individual clusters were given a status of 1.

2. Cluster C_i had a status of 0, whilst cluster C_j had a status of 1, indicating that it had already been merged to another cluster — In this case, the $\textsc{ComVal}_{C_i \cup C_j}$ was calculated, and if an improvement was registered (using the same threshold described above), then the clusters were merged. However, the clusters would be given a status of 2 to indicate that these clusters should not be merged any further.

3. Both clusters C_i and C_j had a status of 1, indicating that both clusters had already been merged separately with other clusters. No further merging is considered so as to avoid overly large clusters. The clusters were given the status 3.

The status of the clusters allows the monitoring of the merging process as this progresses automatically. The procedure reduced the number of clusters substantially, from the initial 21,381 clusters, down to 4,524 clusters. Still, the number of clusters was too large to evaluate manually. However, unlike the segmentation task which would require linguistic expertise to evaluate, the clusters can be evaluated by native Maltese speakers if they are given a reasonable idea of what is meant by morphologically related words. The next section describes the evaluation that was carried out, its setup and discusses the results both from a quantitative and qualitative perspective.

3.4 Evaluating clusters

Since there is no large-scale lexical resource against which to evaluate the clusters, it is not possible to obtain an overall measure through a metric such as accuracy or f-measure. It is also unfeasible to rely on human experts to evaluate all the resulting clusters. These limitations have led to the development of two different evaluation scenarios in an attempt to review the clusters from a quantitative and qualitative point of view.

The first evaluation focused on gathering feedback from experts (trained linguists) who were familiar with the notions of morphologically related words. The second evaluation focused on native Maltese speakers who do not necessarily have any linguistic knowledge or background, but have an intuitive understanding of whether words are morphologically related. A small pilot study was carried out before the second evaluation to ensure that the instructions provided were clear and that there were no technical mishaps. The first evaluation is referred to as the expert evaluation, whilst the second evaluation is referred to as the non-expert evaluation.

The non-expert evaluation used crowd-sourcing, a technique that extracts information by soliciting the participation of online users. There are several platforms available that also allow users to be paid for their participation, such as Amazon's Mechanical Turk[3] or Crowdflower[4]. Such systems are becoming common for the evaluation of results in different areas in computational linguistics and artificial intelligence. However, some of these systems have certain geographical restrictions and usually require payment to participants. Thus we developed our own online evaluation protocol, which was then advertised widely among Maltese speakers. This was necessary as the number of Maltese speakers on existing platforms is likely to be very small, given that Maltese is a 'small' language (in the sense that there are just over 400k native speakers in Malta).

The first main goal for the evaluation was to answer the question: *how good are the clusters that the system has produced?* To address this question, participants were asked to give a subjective rating of the quality of a cluster on a likert-type scale. However, the quality of clusters could also be measured more objectively, as a function of the number of morphologically unrelated words they contained. In an ideal scenario, when related words are not found in the cluster, they could also be included to provide a more complete cluster. The exercise of adding and removing words from the existing clusters also had an indirect benefit — that of providing a gold standard dataset of related words. In order to ensure that both aims were reached, each cluster was evaluated by at least three people so that the agreement between participants could also be measured and analysed.

3.4.1 Pilot study

In order to test the setup of the crowd-sourcing system and the clarity of the instructions provided, a pilot study was carried out with 4 non-expert participants. The pilot participants were asked to (i) watch the instructional video and read through the written instructions, and comment if there were points that were not clear; (ii) provide feedback on the functionality and ease of use of the website; (iii) spend around 10 minutes using the system; and (iv) provide general feedback about the task and highlight any difficulties or uncertainties that arose during the task. The aim of this exercise was to understand what type of problems participants might encounter and whether the description of the task was clear.

The main problem encountered by the pilot participants was that they were not sure whether derivationally related words should be left within a cluster or not. This is especially the case for words that fall under the root and pattern morphological system and different forms are found in the same cluster. For instance, a user was uncertain whether

[3] https://www.mturk.com/mturk/
[4] http://crowdflower.com/

twassal 'to deliver (a message)' should be left under the cluster *tasal* 'to arrive'. From a computational perspective, and given that the type of techniques applied to cluster words were unsupervised, it is not feasible to expect the system to distinguish automatically between inflection and derivational relations. Similarly, it is unrealistic to expect non-experts to always distinguish correctly between inflection and derivation, making the evaluation exercise more complex. In fact, to ensure that participants do not limit morphological relations to only inflection, the instructions contained examples that include derivation (e.g. stating that *ġera* 'he ran' and *ġirja* 'race' are related to each other). As a result of the uncertainty that arose during the pilot exercise, two *test* clusters were manually designed to be used as an indication of a participant's strategy in the removal of words from a cluster. Table 3.8 shows the two clusters which included both derivational and unrelated words. The unrelated words were purposely chosen due to their similar orthographic form to the words in the cluster. The cluster *kbirt* 'grow-1P.SG.PERF' is from the root √KBR, and all word formations follow a root-and-pattern morphology. The cluster *abbuż* 'abuse' follows a concatenative morphology.

Table 3.8 Two test clusters introduced to assess a participant's strategy in removing derivational words from a cluster

Head word	Inflections	Derivations	Unrelated
kbirt *grow*	kiber, kbirna, jikber, tikbirx ikbar	kabbar, kburi, tkabbir, kobor	kiser *broke*
abbuż *abuse*	abbuża, abbużat, jabbuża, nabbuża, tabbuża, abbużati, abbużak	abbużiv, abbużiva, abbużar, abbużivament, abużazzjoni	akkużat *accuse*

The pilot study also provided the option to mark a cluster for deletion, intended for those clusters which were very large in size and contained several unrelated words. However none of the participants used this option and they preferred to correct a large cluster and rate it negatively in terms of quality rather than to delete it completely. Therefore this option was left out from the actual evaluation process to keep the interface and task as simple as possible.

3.4.2 Participants

Three linguists with postgraduate training (one at Masters and two at PhD level) took part in the expert evaluation. For the non-expert evaluation, the targeted participants were

Maltese native speakers, who were mainly sourced from the student population at the University of Malta, as well as through social media networks. In total, 248 people visited the website, of which 106 chose to participate in the evaluation.

3.4.3 Materials and design

The clusters presented for the expert evaluation were purposely chosen to represent a mixture of root-and-pattern and stem-and-affix morphology. The clusters were chosen on the basis of their stems. For example, to cover the stem-and-affix morphology, all clusters whose stem has *prova* 'try' as a substring were chosen. To cover root-and-pattern morphology, those clusters which have stems containing the consonants ħ-s-l (capturing the root √ħsl for *ħasel* 'to wash') were included in the evaluation. Each search resulted in more than one cluster, meaning that the same expert might evaluate the cluster for *ħasel* and *nħaslu* (vii.perf.p3pl or vii.imp.p2pl). It also resulted in capturing stems with different meaning — for instance the cluster with the stem *approva* 'approve' was included since *prova* is its substring. The approach of selecting the clusters for the expert group reflected the overall observations of problems encountered in the segmentation and clustering procedures. The goal of the expert evaluation was to have a dataset which could be used in the future to correct or tweak the techniques used. A total of a 101 clusters were chosen through this manual approach, 26 of which (randomly chosen) were evaluated by all 3 experts so that the inter-annotator agreement between them could be measured. The remaining 75 clusters were randomly allocated between the three experts. Experts were also given the opportunity to give textual feedback/comments on each of the clusters.

The non-expert evaluation placed no limit on the number of clusters a participant could evaluate. 300 clusters were randomly chosen, with the aim that every cluster would be evaluated by least 3 participants so that inter-annotator agreement could be calculated. Table 3.9 provides an overview of the size of the clusters. The majority of clusters are rather small, with less than ten words in a cluster. This can be considered as an advantage for the type of crowd-sourcing exercise since it is easier to check a smaller list of words.

3.4.4 Procedure

The evaluation was developed as a website[5] allowing participants to carry out the task from any location over the Internet. The website provided a short introduction and a description

[5] http://mlrs.research.um.edu.mt/cmexperiment/intro.php

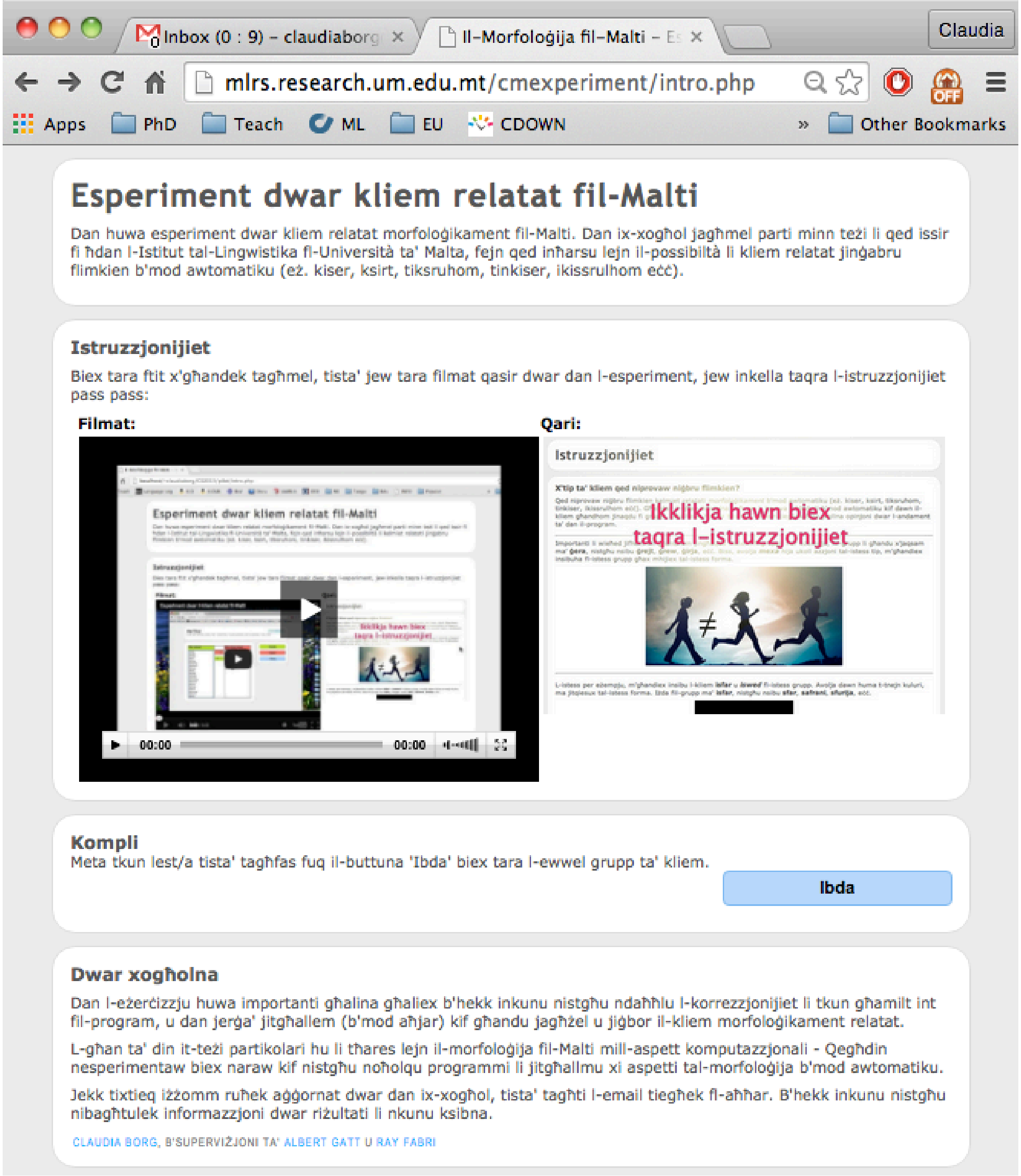
Table 3.9 Average size of the clusters evaluated

Size Range	Expert	Non-expert
< 10	39% (39)	58% (173)
$10 - 19$	30% (31)	26% (79)
$20 - 29$	14% (14)	10% (29)
$31 - 40$	6% (6)	3% (8)
> 40	11% (11)	4% (11)
Total:	101	300
Max Size:	60	88

of the actual task. Since the idea of 'related' words could be rather subjective, a short 3-minute video describing the task was produced, as well as written instructions. The home page for the experiment is shown in fig. 3.2, showing the links to the video and the textual instructions. Figure 3.3 shows the instructions page. The instructions showed a number of examples to illustrate morphological relatedness (e.g. *ġera* 'he ran' and *ġrejna* 'we ran'), distinguishing it from semantic relatedness (e.g. *mexa* 'he walked'). Both the instructional video and the textual instructions were in Maltese since the evaluation was aimed at native speakers.

The main evaluation screen is shown in fig. 3.4. A cluster was presented to participants as a list of words, with the stem of the cluster shown on the top of the list, marked in bold, and stating that the list of words are related to it. This was done in case a cluster contained a majority of unrelated words — in this way, the participant would know which words should be removed and not rely solely on the majority of words. Participants were able to remove words from a cluster, or replace them back in the word list if they changed their mind (fig. 3.5, fig. 3.6). Participants also had to provide a quality rating for a cluster on a likert-type scale, which ranged from *tajjeb ħafna* 'very good' to *ħażin ħafna* 'very bad' (fig. 3.7). Participants also had the option to include additional related words which were missing from the cluster (fig. 3.7). However, in the actual evaluation, participants barely used this function and therefore the analysis below will not include this.

Participants in the non-expert evaluation were continuously exposed to clusters, selected randomly from the pool of 300 clusters, until they would choose to stop, by pressing the 'stop' button. A participant would never see the same cluster twice in one session since the back-end system kept track of the responses by each participant. No registration was


Inbox (0 : 9) – claudiaborg
Il-Morfoloġija fil-Malti – E
Claudia
mlrs.research.um.edu.mt/cmexperiment/intro.php
Apps
PhD
Teach
ML
EU
CDOWN
»
Other Bookmarks
OFF

Esperiment dwar kliem relatat fil-Malti

Dan huwa esperiment dwar kliem relatat morfoloġikament fil-Malti. Dan ix-xogħol jagħmel parti minn teżi li qed issir fi ħdan l-Istitut tal-Lingwistika fl-Università ta' Malta, fejn qed inħarsu lejn il-possibiltà li kliem relatat jinġabru flimkien b'mod awtomatiku (eż. kiser, ksirt, tiksruhom, tinkiser, ikissrulhom eċċ).

Istruzzjonijiet

Biex tara ftit x'għandek tagħmel, tista' jew tara filmat qasir dwar dan l-esperiment, jew inkella taqra l-istruzzjonijiet pass pass:

Filmat:

Esperiment dwar kliem relatat fil-Malti
Ikklikja hawn biex taqra l-istruzzjonijiet
00:00
00:00

Qari:

Istruzzjonijiet
X'tip ta' kliem qed nipprovaw niġbru flimkien?
Ikklikja hawn biex taqra l-istruzzjonijiet
≠

Kompli

Meta tkun lest/a tista' tagħfas fuq il-buttuna 'Ibda' biex tara l-ewwel grupp ta' kliem.

Dwar xogħolna

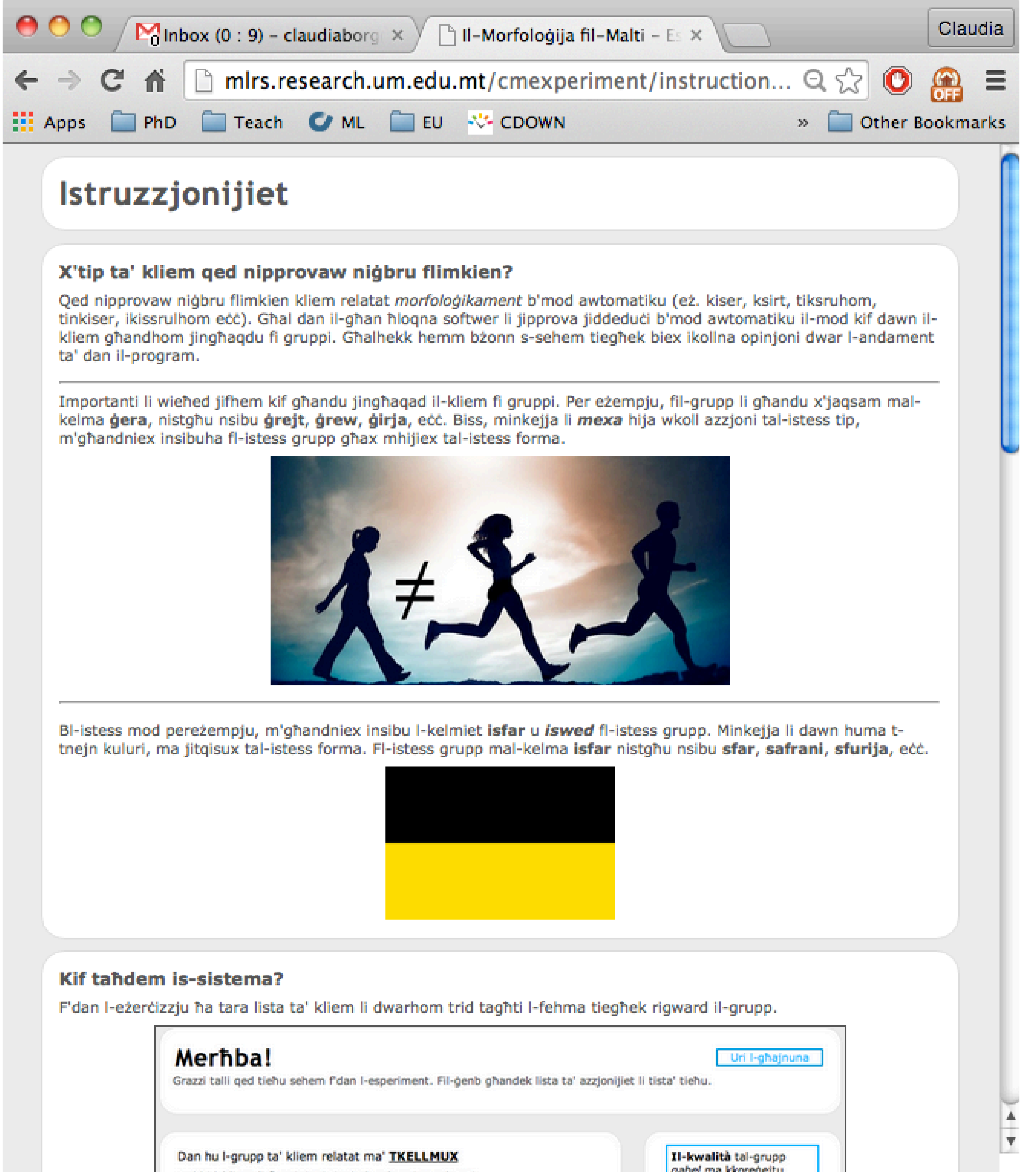
Dan l-eżerċizzju huwa importanti għalina għaliex b'hekk inkunu nistgħu ndaħħlu l-korrezzjonijiet li tkun għamilt int fil-program, u dan jerġa' jitgħallem (b'mod aħjar) kif għandu jagħżel u jiġbor il-kliem morfoloġikament relatat.

L-għan ta' din it-teżi partikolari hu li tħares lejn il-morfoloġija fil-Malti mill-aspett komputazzjonali - Qegħdin nesperimentaw biex naraw kif nistgħu noħolqu programmi li jitgħallmu xi aspetti tal-morfoloġija b'mod awtomatiku.

Jekk tixtieq iżżomm ruħek aġġornat dwar dan ix-xogħol, tista' tagħti l-email tiegħek fl-aħħar. B'hekk inkunu nistgħu nibagħtulek informazzjoni dwar riżultati li nkunu ksibna.

CLAUDIA BORG, B'SUPERVIŻJONI TA' ALBERT GATT U RAY FABRI

Fig. 3.2 The welcome page on the evaluation website


Inbox (0 : 9) - claudiaborg
Il-Morfoloġija fil-Malti - Es
Claudia
mlrs.research.um.edu.mt/cmexperiment/instruction...
Apps
PhD
Teach
ML
EU
CDOWN
Other Bookmarks
≠
Uri l-għajnuna

Istruzzjonijiet

X'tip ta' kliem qed nipprovaw niġbru flimkien?

Qed nipprovaw niġbru flimkien kliem relatat *morfoloġikament* b'mod awtomatiku (eż. kiser, ksirt, tiksruhom, tinkiser, ikissrulhom eċċ). Għal dan il-għan ħloqna softwer li jipprova jiddeduċi b'mod awtomatiku il-mod kif dawn il-kliem għandhom jingħaqdu fi gruppi. Għalhekk hemm bżonn s-sehem tiegħek biex ikollna opinjoni dwar l-andament ta' dan il-program.

Importanti li wieħed jifhem kif għandu jingħaqad il-kliem fi gruppi. Per eżempju, fil-grupp li għandu x'jaqsam mal-kelma **ġera**, nistgħu nsibu **ġrejt**, **ġrew**, **ġirja**, eċċ. Biss, minkejja li ***mexa*** hija wkoll azzjoni tal-istess tip, m'għandniex insibuha fl-istess grupp għax mhijiex tal-istess forma.

Bl-istess mod pereżempju, m'għandniex insibu l-kelmiet **isfar** u ***iswed*** fl-istess grupp. Minkejja li dawn huma t-tnejn kuluri, ma jitqisux tal-istess forma. Fl-istess grupp mal-kelma **isfar** nistgħu nsibu **sfar**, **safrani**, **sfurija**, eċċ.

Kif taħdem is-sistema?

F'dan l-eżerċizzju ħa tara lista ta' kliem li dwarhom trid tagħti l-fehma tiegħek rigward il-grupp.

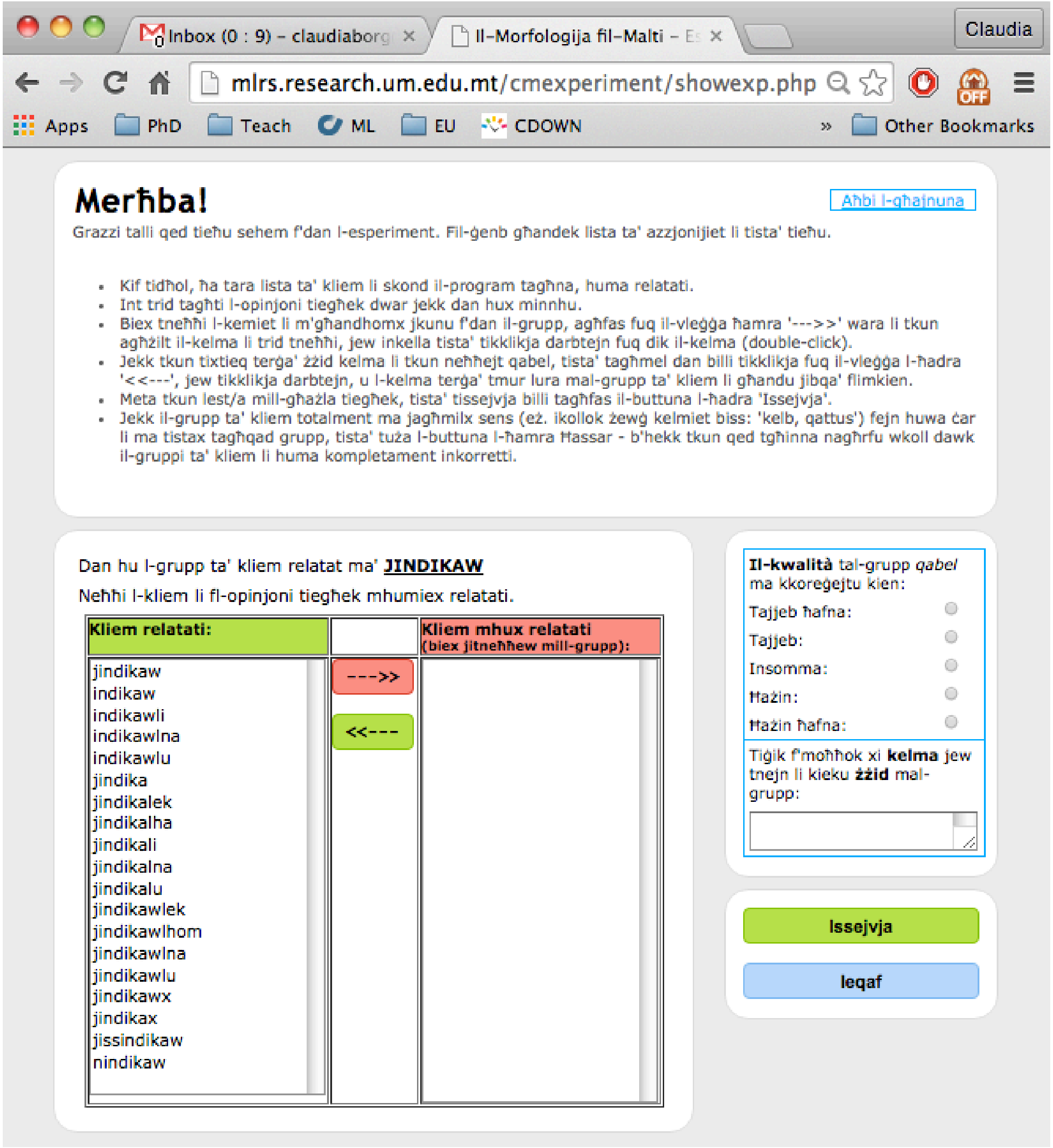
Merħba!

Grazzi talli qed tieħu sehem f'dan l-esperiment. Fil-ġenb għandek lista ta' azzjonijiet li tista' tieħu.

Dan hu l-grupp ta' kliem relatat ma' **TKELLMUX**

Il-kwalità tal-grupp qabel ma kkoreġejtu

Fig. 3.3 The webpage showing the instructions in textual format


Inbox (0 : 9) - claudiaborg
Il-Morfologija fil-Malti - E
Claudia
mlrs.research.um.edu.mt/cmexperiment/showexp.php
Apps
PhD
Teach
ML
EU
CDOWN
Other Bookmarks

Merħba!

Aħbi l-għajnuna

Grazzi talli qed tieħu sehem f'dan l-esperiment. Fil-ġenb għandek lista ta' azzjonijiet li tista' tieħu.

- Kif tidħol, ħa tara lista ta' kliem li skond il-program tagħna, huma relatati.
- Int trid tagħti l-opinjoni tiegħek dwar jekk dan hux minnhu.
- Biex tneħħi l-kemiet li m'għandhomx jkunu f'dan il-grupp, agħfas fuq il-vleġġa ħamra '--->>' wara li tkun agħżilt il-kelma li trid tneħħi, jew inkella tista' tikklikja darbtejn fuq dik il-kelma (double-click).
- Jekk tkun tixtieq terġa' żżid kelma li tkun neħħejt qabel, tista' tagħmel dan billi tikklikja fuq il-vleġġa l-ħadra '<<---', jew tikklikja darbtejn, u l-kelma terġa' tmur lura mal-grupp ta' kliem li għandu jibqa' flimkien.
- Meta tkun lest/a mill-għażla tiegħek, tista' tissejvja billi tagħfas il-buttuna l-ħadra 'Issejvja'.
- Jekk il-grupp ta' kliem totalment ma jagħmilx sens (eż. ikollok żewġ kelmiet biss: 'kelb, qattus') fejn huwa ċar li ma tistax tagħqad grupp, tista' tuża l-buttuna l-ħamra Ħassar - b'hekk tkun qed tgħinna nagħrfu wkoll dawk il-gruppi ta' kliem li huma kompletament inkorretti.

Dan hu l-grupp ta' kliem relatat ma' **JINDIKAW**

Neħħi l-kliem li fl-opinjoni tiegħek mhumiex relatati.

Kliem relatati:		**Kliem mhux relatati** **(biex jitneħħew mill-grupp):**
jindikaw indikaw indikawli indikawlna indikawlu jindika jindikalek jindikalha jindikali jindikalna jindikalu jindikawlek jindikawlhom jindikawlna jindikawlu jindikawx jindikax jissindikaw nindikaw	--->> <<---	

Il-kwalità tal-grupp *qabel* ma kkoreġejtu kien:

Tajjeb ħafna: ○
Tajjeb: ○
Insomma: ○
Ħażin: ○
Ħażin ħafna: ○

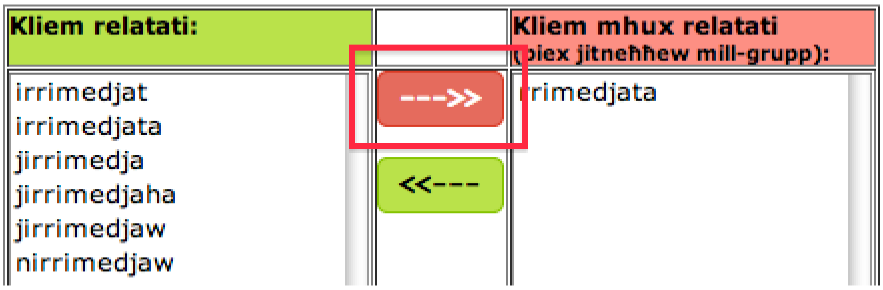
Tiġik f'moħħok xi **kelma** jew tnejn li kieku **żżid** mal-grupp:

Issejvja

Ieqaf

Fig. 3.4 The evaluation webpage showing a cluster and the functionality available to a participant. The instructions at the top can be toggled on and off, and provide the participant with a short reminder of the task.

required for participation, however the website did allow users to leave their email address at their own discretion. The two test clusters were presented always in the same order — the cluster *abbuż* was always the fourth cluster to be presented to a participant, and the sixth cluster was always the test cluster *kbirt*. Apart from the clusters evaluated in a single session, no other tracking mechanisms were used. So if the same person returned to the evaluation page the next day, the system treated this as a new participant.



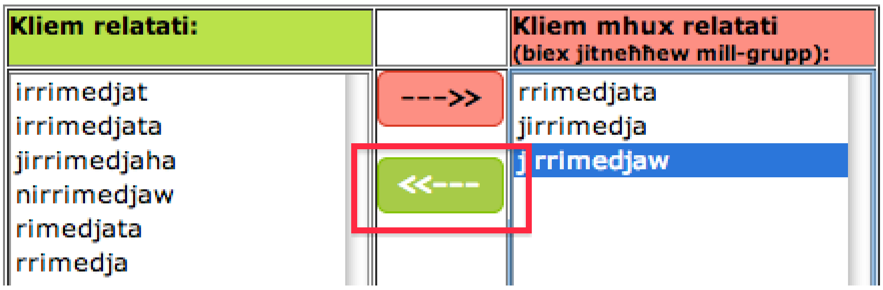
Kliem relatati:

irrimedjat
irrimedjata
jirrimedja
jirrimedjaha
jirrimedjaw
nirrimedjaw

Kliem mhux relatati
(biex jitneħħew mill-grupp):

rrimedjata

Fig. 3.5 Removal of words



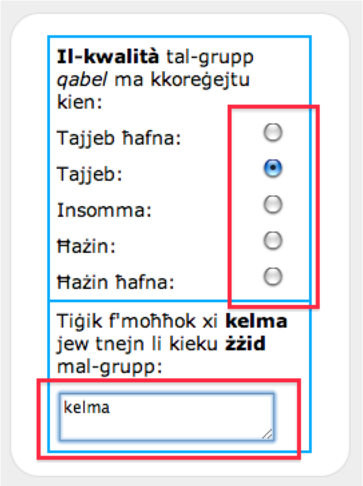
Kliem relatati:

irrimedjat
irrimedjata
jirrimedjaha
nirrimedjaw
rimedjata
rrimedja

Kliem mhux relatati
(biex jitneħħew mill-grupp):

rrimedjata
jirrimedja
jirrimedjaw

Fig. 3.6 Reinserting words in a cluster



Il-kwalità tal-grupp *qabel* ma kkoreġejtu kien:

Tajjeb ħafna: ○

Tajjeb: ●

Insomma: ○

Ħażin: ○

Ħażin ħafna: ○

Tiġik f'moħħok xi **kelma** jew tnejn li kieku **żżid** mal-grupp:

kelma

Fig. 3.7 Rating the quality of words

3.5 Analysis of results

In this section we analyse the responses received from the two evaluations carried out. The expert evaluation received a total of 153 responses over a set of 101 clusters. The three experts were allocated a fixed amount of clusters to evaluate, with 26 clusters common to all three experts. The non-expert evaluation received a total of 2117 valid responses from 106 participants covering the 300 clusters and the additional two test clusters. However, from the 106 participants, responses from six participants were discarded because they did not carry out the task appropriately[6], leaving a total of 100 participants covering 1848 responses over the 300 clusters and 135 responses over the two test clusters, summarised in table 3.10.

3.5.1 Removal of words

One of the main objectives of the evaluation was to have participants remove unrelated words from the clusters. From a quantitative perspective, analysing how many words were actually removed from the clusters provides an insight into how well the clustering technique preforms. Table 3.11 below divides the clusters into bins reflecting the amount

[6] As an example, one participant removed all words from every single cluster evaluated, and therefore such results cannot be included in the evaluation. It is only in clear cases where the majority of clusters were not properly evaluated that such responses were discarded. In cases of genuine errors, these were left as part of the responses, which will be analysed through inter-annotator agreement between participants.

Table 3.10 Summary of the responses received from the evaluation

	Expert	Non-expert
Participants	3	100
Total Clusters	101	300
Number of Responses	153	1848
Avg. No. of Clusters per participant	51	18
Test Clusters	–	2
No. of Responses on Test Clusters		135
Cluster *abbuż*		72
Cluster *kbirt*		63

of words removed, and displays the percentage of the clusters used in the expert and non-expert evaluations that fall into each bin. For example, experts did not remove any words from 54% of clusters, whilst non-experts did not remove any words from 56% of the clusters.

Table 3.11 Number of words removed – Cluster bins and percentage of clusters in each bin

Words removed	Expert	Non-expert
0	54% (82)	56% (1025)
1	3% (5)	4% (79)
2	18% (28)	15% (274)
3 – 4	10% (15)	10% (189)
5 – 7	5% (7)	7% (131)
8 – 10	5% (8)	2% (44)
11 – 20	3% (4)	4% (66)
21 – 30	3% (4)	1% (18)
31 or more	0% (0)	1% (22)
Total evaluations	153	1848

The same data is shown in table 3.12, this time showing the percentage of words removed from a cluster. This view is important to consider because removing 1 word from a 6-word cluster gives a higher percentage than removing 1 word from a 20-word cluster.

The clearest observations from these results are related to the extremes. From the evaluations carried out there were just over half that had no words removed. This can be seen as a positive indication given that the clusters were created fully automatically without any

Table 3.12 Percentage of words removed – Cluster bins and percentage of clusters in each bin

Words removed	Expert	Non-expert
0%	54% (82)	56% (1025)
1 – 5%	1% (2)	1% (14)
5 – 10%	5% (8)	4% (79)
10 – 15%	3% (4)	3% (60)
15 – 20%	6% (9)	4% (77)
20 – 40%	15% (24)	16% (287)
40 – 60%	5% (7)	7% (137)
60 – 80%	9% (14)	1% (101)
over 80%	2% (3)	4% (68)
Total evaluations	153	1848

predetermined knowledge built into the technique. At the other end of the scale, there is a very small percentage of evaluations where clusters had a rather large number of words removed. In analysing the latter clusters, it was clear that these are the situations where the technique fails. For example, for the cluster with the stem *ittra* 'letter', this contained several unrelated words because *ittra* is either a substring in several other words or has a very high orthographic similarity. Clustered together with *ittra* were *tittraduċi* 'translate', *ittratat* 'treated', *ittardja* 'delayed'. These type of errors could be catered for in the clustering technique by empirically adjusting the weights in the calculation of the semantic cohesiveness of a cluster through the comVal metric described above in eq. (3.7). For such a scenario, a development or training dataset would be required so as to test different parameters and decide according to the results what the best weighted balance would be between the orthographic and semantic similarity of words.

The test clusters (table 3.8) provided an insight into the 'attentiveness' of participants in the non-expert evaluation. From the 135 responses for the test clusters, 17 responses did not remove any words in these clusters — 13 for the *abbuż* cluster, and 4 for the *kbirt* cluster. This equates to 12% of the responses. Both clusters had a word which is completely unrelated to the head word, and should have been removed. The cluster *abbuż* 'abuse' had the word *akkużat* 'accused', and the cluster *kbirt* 'I grew' had the word *kiser* 'he broke'. Both were specifically chosen for their orthographic similarity with the words in the cluster, and similar to the type of errors that the automatic clustering technique could make.

Table 3.13 Removal of words in the test clusters

	abbuż cluster	*kbirt* cluster
Total responses	72	63
No word removal	13 (18%)	4 (6%)
Removing unrelated word only	42 (58%)	22 (35%)
Removing also derivational words	17 (24%)	37 (59%)

Further analysis of the test clusters and the words removed is provided in table 3.13, where an interesting pattern of responses to the different clusters can be noticed. The percentage of responses removing derivationally related words in the *abbuż* cluster was 24%, whilst in the *kbirt* cluster it was 59%. The difference is quite substantial and it might indicate that the semantic concepts in stem-and-affix vs. root-and-pattern morphology when it comes to derivationally related words is more distinct in the latter group. In other words, considering the cluster for *kbirt*, the word *kburi* 'proud' is derived from the same root √KBR. Nearly 60% of the responses removed this word and did not consider it as a related word to *kbirt*. On the other hand, participants might have perceived derivationally related words in the stem-and-affix morphology as semantically closer in meaning to the head word. For the cluster *abbuż*, there was *abbużivament* 'abusively-ADV', which was only removed in 6 of the responses, and **abbużazzjoni* 'abuse-NOUN' which was purposely formed using a common suffix *-azzjoni* to form a noun. However, this word is not attested in the MLRS corpus and a Google search for **abbużazzjoni* does not return any results. Yet only 17 (24%) responses removed this word, with the majority leaving it as part of the cluster.

The difference in the treatment of derivationally related words coming from a root-and-pattern morphology to a stem-and-affix morphology might also be a reflection that the productivity in Maltese morphology is largely in the latter system, with a large number of loan words coming from Anglo-Saxon or Romance origin. It also demonstrates how highly productive the stem-and-affix morphology system in Maltese is, and how easily native speakers accept new word formations which are plausible. The different treatment of derivationally related words between the two morphological systems could be the basis of an interesting study from a psycholinguistic perspective, though one that is beyond the scope of this work.

3.5.2 Quality ratings

The number and percentage of words removed cannot be taken as the sole measure of quality of the clusters produced. Users were asked to specifically rate the quality of a cluster, and although this is a rather subjective opinion, the correlation between this judgement and the number of words removed is calculated using Pearson's correlation coefficient. A *perfect* cluster would have no words removed and be given a high quality rating, whilst a *bad* cluster would probably have several words removed and be given a very low quality rating, thus providing a correlation between the two. Table 3.14 shows the ratings given per evaluation and the Pearson's correlation coefficient between the quality rating and the average percentage of words removed.

Table 3.14 Quality ratings per evaluation, and the correlation between the quality rating and the percentage of words removed from a cluster

Quality ratings	Experts	Non-expert
Very Good	22% (34)	44% (810)
Good	35% (53)	30% (546)
So-so	26% (40)	15% (272)
Bad	13% (20)	7% (128)
Very Bad	4% (6)	5 % (92)
Correlation:	0.7837	0.8217

Similar to the removal of words, the majority of the evaluators provided a high rating to the clusters, with over 70% of the evaluations given a good or very good rating. An interesting observation over these results is that whilst non-expert participants gave a 'very good' quality rating to a large number of clusters, the experts were less inclined to give such a rating, showing a stronger preference for the 'good' rating instead. Experts were expected to rate clusters not only on the words that were present in the cluster, but also on the words that were omitted by the system. An expert's judgement can be considered as more objective especially due to the linguistic understanding of what morphologically related words are. The correlations between the quality rating and the percentage of words removed is high overall, indicating that indeed participants had a tendency to give a better quality rating to a cluster when fewer words were removed.

3.5.3 Inter-Annotator Agreement

Apart from qualitative and quantitative measures, we also considered whether there is overall agreement between participants on the actual words removed. Agreement between participants (often referred to as raters or coders in the literature) is referred to as Inter-annotator agreement (IAA) and was calculated using Krippendorff's Alpha-Reliability Coefficient (Artstein and Poesio, 2008; Krippendorff, 2011). Krippendorff's alpha is a generalisation of several other reliability indices, and was chosen because it can be adapted to multiple raters (unlike, say, Cohen's Kappa). Mathematically, agreement is actually calculated by evaluating the observed disagreement (D_o) in relation to the expected disagreement (D_e) and subtracting this from 1, which would be full agreement. The basic formula is:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{3.8}$$

The agreement over a cluster was calculated as follows. Let C be a cluster, consisting of words $w_1, \ldots, w_n$. For every evaluator, we represented the cluster as a vector v of binary values, so that $v_i = 1$ if word w_i was left in the cluster by the evaluator, and $v_i = 0$ if it was removed. The resulting matrix of vectors, representing the evaluators' decisions for cluster C, is then used to calculate the IAA using an implementation of Krippendorff's alpha[7] in R[8]. The coefficient ranges between 0 and 1, with 1 indicating full agreement.

The IAA results are presented in table 3.15, providing an overview of the agreement for the expert and non-expert groups. The IAA for the expert evaluation is carried out over the 26 clusters evaluated in common by all the three experts[9]. The average agreement achieved overall in the respective evaluations is at 0.908 for the expert group, and 0.598 for the non-expert group. The highest and lowest agreements are also given, together with the percentage of clusters spread into bins according to the range of IAA achieved.

The expert group had a very high average agreement of 0.908, with 84% of the clusters having very high agreement between annotators — this can be expected since both the task and the linguistic knowledge of morphologically related words are well known to this group of participants. On the other hand, the average agreement of 0.598 for the

[7] The implementation used for Krippendorff's alpha is part of the IRR library http://rss.acs.unt.edu/Rdoc/library/irr/html/kripp.alpha.html.

[8] http://www.r-project.org/

[9] Experts also evaluated a further 25 clusters each, so as to gain a broader coverage of evaluated clusters, but these 75 clusters were only checked by one expert, and therefore inter-annotator agreement is not applicable for these clusters.

Table 3.15 Inter-Annotator Agreement

Description	Experts	Non-expert	Test
No. of Clusters	26	300	2
Avg. no. Evaluators	3	6.16	67.5
Avg. Agreement	0.908	0.598	0.4615
Lowest Agreement	-0.0126	-0.166	0.425
Highest Agreement	1.0	1.0	0.498
Bins:			
Negative	4% (1)	23% (68)	0
less than 0.20	0% (0)	7% (21)	0
0.21 - 0.40	0% (0)	7% (20)	0
0.41 - 0.60	7% (2)	5% (16)	100% (2)
0.61 - 0.80	4% (1)	8% (24)	0
0.81 - 1.00	84% (22)	50% (151)	0

non-expert group is much lower and again this is to be expected, given the gaps in linguistic knowledge and expertise of the general Maltese native speaker. The agreement between participants is however adequately high, especially when considering that 50% of the clusters have a very high agreement (between 0.81 and 1).

The test clusters result in lower agreement mainly due to the difference in strategies employed by participants as to whether derivationally related words should be removed or not, discussed previously in §3.5.1. The low agreement here is to be expected since the clusters were purposely designed to find different strategies used by participants.

In further analysis of the IAA, negative agreement is rather unusual — by definition, Krippendorff's alpha returns a value between 0 and 1. However, when we looked into the particular clusters with negative agreement, the negative agreement always occurred when only one participant would have removed a couple of words from the cluster, thus creating a matrix were the value '0' is seen only in one row e.g. two words removed by the same expert e_3, with an example of such a matrix shown in fig. 3.8.

Krippendorff's alpha is calculated on the basis of the expected disagreement. When the majority of the vectors contain only '1's, and a single vector contains a couple of '0's,

$$
\begin{array}{ccccccccccccc}
 & w_1 & \ldots & & & & & & & & & w_n \\
e_1 & 1 & \ldots & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
e_2 & 1 & \ldots & 1 & 1 & 1 & 1 & \mathbf{0} & \mathbf{0} & 1 & 1 & 1 \\
e_3 & 1 & \ldots & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array}
$$

Fig. 3.8 An evaluation matrix showing only two instances where an evaluator discarded a couple of words.

Krippendorff's alpha severely punishes this phenomenon since raters must show some level of covariation in their agreement. This situation resulted in a negative alpha[10].

Through this examination, we also noted quite a few instances where alpha was negative or very low due to outliers. This situation occurs when there is a general strong agreement amongst the majority of participants, with one outlier providing a different response from the other participants. Since on average each cluster in the non-expert evaluation was evaluated by more than six participants, it was possible to recalculate IAA excluding outliers. To do this, the pairwise agreement was calculated to identify which of the evaluators in a cluster were actually outliers. If a third or less of the participants were outliers, then the IAA was re-calculated without the outliers. If more than a third of the evaluators were outliers, then all evaluators were included (as per the original IAA calculation), meaning that there was overall considerable disagreement between participants regarding which words should have been removed. In practice this meant that if six participants evaluated a cluster, the number of outliers must be two or less for the responses of the outliers to be discarded from the IAA calculations. Table 3.16 shows the IAA results calculated after the responses of the outliers were discarded.

The agreement is substantially much higher than the previous IAA, primarily due to cases where one or two participants removed different word/s from the rest of the group. Due to the type of task, and the issues already highlighted with the removal of derivationally related words, it is quite understandable that there were a number of outliers for most of the evaluated clusters. The average IAA for the non-expert evaluation increased considerably from 0.598 to 0.897 — reflecting that there was a large number of clusters where at least four people agreed on the words that should be left in a cluster. The agreement in the

[10] See http://dfreelon.org/2009/12/14/from-the-mailbag-121409/ and Krippendorff (2004) for further information.

Table 3.16 Inter-Annotator Agreement excluding outliers

Description	Experts	Non-expert	Test
No. of Clusters	26	300	2
Avg. no. Evaluators	2.96	5.46	47.5
Avg. Agreement	0.948	0.897	0.7655
Lowest Agreement	0.42	-0.156	0.72
Highest Agreement	1.0	1.0	0.811
Bins:			
Negative	0% (0)	2% (7)	0% (0)
0.00 - 0.20	0% (0)	2% (7)	0% (0)
0.21 - 0.40	0% (0)	2% (6)	0% (0)
0.41 - 0.60	7% (2)	2% (5)	0% (0)
0.61 - 0.80	4% (1)	9% (28)	50% (1)
0.81 - 1.00	89% (23)	82% (247)	50% (1)

test clusters also increased, reflecting that there was some majority agreement between participants about the contents of the two clusters.

3.5.4 Hybrid morphology and the clustering technique

One of the main questions that this research sought to discuss was whether the type of techniques used were suitable for a language that has a hybrid morphological system in which different (concatenative and non-concatenative) processes are at work in tandem. To gain insight into this question, we analysed the clusters evaluated by the expert group, which were purposely selected to be roughly balanced between the two processes. Table 3.17 provides an overview of how the 101 clusters were divided between concatenative (CON) and non-concatenative (NC) and the size bins for these clusters. Through this first analysis it is possible to observe that concatenative clusters tend to be larger in size than non-concatenative clusters. In principle, both processes should have similar-sized clusters since for example the inflective process of verbal morphology would remain the same irrespective of whether the word has a stem or a root. However, due to the stem variation occurring in the root-and-pattern process, the clustering of such inflective words would be more challenging. This could be one of the reasons behind the difference in size of clusters.

Table 3.17 The spread of clusters used in the expert evaluation divided between concatenative and non-concatenative processes and the respective cluster size

Size	NC	CON
< 10	53% (25)	26% (14)
10–19	23% (11)	37% (20)
20–29	13% (6)	15% (8)
30–39	2% (1)	9% (5)
> 40	9% (4)	13% (7)
Total	47	53
Evaluated by all experts	13	13
Evaluated by one expert	34	40

The number and percentage of words removed was also analysed by distinguishing between the two types of processes. Table 3.18 shows the number and percentage of words removed according to the two processes. The clusters belonging to the non-concatenative process had a relatively larger percentage of words removed when compared to those of the concatenative group. Only 45% of non-concatenative clusters had no words removed compared to 61% of the concatenative clusters. However, the gap closes when considering the percentage of clusters which had a third or more of their words removed, with 25% for the non-concatenative and 20% for the concatenative group. However, the concatenative group also had clusters which had more than 80% of their words removed. This might indicate that although in general the clustering technique performs better for the concatenative group, there might be certain aspects or situations that end up forming bad clusters. This was observed for instance with the cluster formed for *ittra* ‘letter’, described in §3.5.1. This cluster contained a number of morphologically unrelated words. However, the evaluation exercise provides a set of clusters which have been corrected and improved, and could be used as a development set to improve the clustering technique.

In terms of quality ratings between the two processes, a similar trend can be observed. Table 3.19 provides the breakdown of the quality ratings for clusters split between the two processes and the correlation of the quality to the percentage of words removed. The non-concatenative clusters generally have lower quality ratings when compared to the concatenative clusters. But both groups have a strong correlation between the percentage of words removed and the quality rating, clearly indicating that the perception of a cluster's quality is related to the percentage of words removed.

Table 3.18 Number of words removed per cluster bin in the expert evaluation split by concatenative and non-concatenative processes

Removal of words					
By Quantity	NC	CON	By Percentage	NC	CON
0	45% (33)	61% (49)	0%	45% (33)	61% (49)
1	0% (0)	6% (5)	1–5%	1% (1)	1% (1)
2	34% (25)	4% (3)	5–10%	7% (5)	4% (3)
3–4	8% (6)	11% (9)	10–20%	5% (4)	11% (9)
5–7	7% (5)	3% (2)	20–30%	17% (12)	4% (3)
8–10	4% (3)	6% (5)	30–40%	8% (6)	4% (3)
11–20	1% (1)	4% (3)	40–60%	7% (5)	3% (2)
21–30	0% (0)	5% (4)	60–80%	10% (7)	9% (7)
31 or more	0% (0)	0% (0)	over 80%	0% (0)	4% (3)

Table 3.19 Quality of clusters split by concatenative and non-concatenative processes and the correlation between the quality and the percentage of words removed

Quality	NC	CON
Very Good	17% (12)	28% (22)
Good	33% (24)	36% (29)
So-so	34% (25)	18% (15)
Bad	12% (9)	14% (11)
Very Bad	4% (3)	4% (3)
Correlation:	0.780	0.785

Clearly, there is a notable difference between the clustering of words from concatenative and non-concatenative morphological processes. Both have their strengths and pitfalls, but neither of the two processes excel or stand out over the other. One of the problems with non-concatenative clusters was that of size. The initial clusters were formed on the basis of the stems, and due to stem variation the non-concatenative clusters were rather small. Although the merging process catered for clusters to be put together and form larger clusters, the process was limited to a maximum of two merging operations. This might not have been sufficient for the small-sized non-concatenative clusters. In fact, only 10% of the clusters contained 30 or more words when compared to 22% of the concatenative clusters. A solution might be to consider the size of the clusters as part of the decision process in merging clusters. If the clusters are of a particular small size, further merging could be allowed. The threshold would have to be determined empirically through experiments, possibly using the dataset obtained through the evaluation.

The problem of size with concatenative clusters was on the other side of the scale. Although the majority of clusters were of reasonable size, it seemed as though the bigger the cluster, the more non-related words were found. In order to explore this problem further, one possibility would be to check whether there is a correlation between the size of a cluster and the percentage of words removed from it. It is possible that the unsupervised technique does not perform well on larger clusters, and if such a correlation exists, it would be possible to set an empirically determined threshold for cluster size.

Given the results achieved, it is realistic to state that the unsupervised clustering technique could be further improved using the evaluated clusters as a development set to better determine the thresholds in the metrics proposed above. This improvement would impact both concatenative and non-concatenative clusters equally. In general, the clustering technique does work slightly better for the concatenative clusters, and this is surely due to the clustering of words on the basis of their stems. This is reflected by the result that 61% of the clusters had no words removed compared to 45% of the non-concatenative clusters. However, a larger number of concatenative clusters had a large percentage of words removed. Indeed, if the quality ratings were considered as an indicator of how the technique performs on the non-concatenative vs the concatenative clusters, the judgement would be so-so to good for the non-concatenative and good for the concatenative clusters. Thus the performance is sufficiently close to render the technique as valid for a language with a hybrid morphological system such as that found in Maltese.

3.6 Conclusion

This chapter described a fully unsupervised technique that segmented words and clustered morphologically related words together. The segmentation process used transitional probabilities to discover affixes automatically and proposed a ranked list of potential affixes. The top ranked affixes were then used to propose word segmentations. Since the segmentation process was intended to aid the clustering technique, all possible valid segmentations were taken into consideration. The clustering of words used the potential segmentations as a point of departure to group words together. Further improvements to the clusters were then carried out through the inclusion of orthographic and semantic similarity of words. A metric was devised to measure how related words in the same cluster were. This was used as the basis for merging together related clusters and reduce the number of clusters by nearly 80%, resulting in 4,524 clusters. Since no resources were available to evaluate the clusters automatically, an evaluation exercise was set-up with two groups of participants — experts and non-experts. The evaluation sought to assess the quality of the clusters from various angles, and allowed participants to remove words from clusters and rate clusters in terms of the perceived quality. Moreover, two test clusters were used to analyse the strategies that participants used when it came to derivationally related words. The responses were analysed for their inter-annotator agreement, which was found to be quite high especially when outliers were discarded. The results of the evaluation also provided a dataset of clusters which could be used in future as a development set to fine-tune the clustering technique. The response by the expert group was further analysed to determine whether the techniques used were feasible to be applied on a hybrid language such as Maltese. The resulting analysis revealed that although the techniques are better suited for clusters of words following a concatenative process, their application on the non-concatenative segment of the language still provided fairly reasonable results, and that clusters in both categories would require further improvement.

Although clustering has a role in the morphological analysis of words, it is limited to simply grouping such words together. The morphological meaning of each word still needs to be determined. The segmentation process can contribute to identifying the morphemes in words. However ultimately, labels are required to associate meaning to the morphemes in words. In the following chapters the research deviates from the clustering task, and focuses solely on the task of labelling, using supervised machine learning techniques to learn classifiers for the different morphological aspects present in Maltese.

Chapter 4

Morphological Labelling using Supervised Techniques

4.1 Introduction

Having looked at the tasks of segmenting and clustering of morphologically related words together, the natural direction is to now turn our attention to morphological labelling of words. The segmentation task provided a list of possible segments that were used as the foundation for the clustering task. The latter then used orthographic and semantic information to improve the clusters. Although a portion of the clusters were evaluated by experts and non-experts, the clusters do not contain information on the actual type of morphological relation that words have. This means that the clusters themselves are not directly linked to the labelling task. The segmentation task provides the basis of splitting words into affixes; however different segmentation approaches will be used before returning to the probabilistic segmentation model.

There are several possible approaches to learning morphological labels — this research looks at morphological labelling as a classification problem. A classifier is a type of algorithm that learns how labelled data (training data) is categorised, and generalises the observations into a model or representation of this data. It then makes predictions on unseen instances based on the initial observations. The predictions are restricted to the set of classes learned during the training phase, which in this case would be the resulting morphological labels. This approach can be quite beneficial to morphology learning since it is not practical to have a list of all words in a language and their morphological labels — and for Maltese this type of data is limited. Moreover, a language evolves and new words come into use. A system that can generalise its current knowledge and apply it to new words is a favourable characteristic for a morphological analyser.

The proposed morphological classification system deals with three primary part-of-speech categories — verbs, nouns and adjectives. The initial focus is on verbal morphology since this is the most complex and challenging in terms of its morphological properties. The classification system consists of a series of classifiers, each representing a single morphological property, which are then executed in a cascading sequence and with each classifier providing more information to the following classifiers. The classifiers in the verb cascade sequence were trained on a dataset put together from Ġabra[1], an online lexicon focussing on inflections of verbs of Arabic origin. The noun and adjective cascades used a dataset taken from the ongoing dictionary project[2]. The individual classifiers were evaluated on the training data using a 10-fold cross validation system during the development cycle. Once the cascade sequence was in place, this was evaluated as a whole on unseen data.

The rest of the chapter is structured as follows. The first part of this chapter describes the work carried out on the verb cascade. §4.2 defines morphological labelling as a classification problem and what type of features will be used from a machine learning perspective. It describes a number of experiments carried out to determine the best data representation and describes some issues encountered in using the Ġabra dataset. §4.3 then provides analyses for different experiments carried out to find the best cascade sequence. A number of different representations were tested, described in §4.3.2, which aimed at analysing whether it was possible to remove ambiguity in the data and therefore achieving better performing classifiers. However, this resulted in classifiers that overfitted the data and performed worse when applied to unseen data. A similar setup was carried out for the noun and adjective cascades, using a dataset extracted from a Maltese-English dictionary. This is described in §4.4, followed by a description of the experiments and the results in §4.5. A final conclusion is presented in §4.6, outlining the results of the research, a way forward for the classification system and future developments for the cascade system.

4.2 Morphological labelling as a classification problem

A morphological classification system takes a segmented word as its input, and classifies it according to some *features* that the system would use to predict which class a word should fall into. The term feature here refers to anything that can provide information

[1] http://mlrs.research.um.edu.mt/resources/gabra/

[2] http://www.dizzjunarjumalti.com

about a word's morphological properties and resulting labels. So for instance, in Maltese, words ending in the suffix *-a* tend to be feminine (e.g. *sabiħa* is the feminine for *sabiħ* 'beautiful'). Thus the suffix *-a* can be seen as a feature which allows the classifier to predict that the word's gender belongs to the feminine class. We refer to features such as affixes as *basic features* because they form an integral part of the word itself. There are also more complex features which can be used by the classification system. For example, once a word such as *sabiħa* has its gender classified as feminine, the classification system can infer that the number of the word is singular. In this way, the classification of a word's gender is used to classify other morphological properties of a word. We refer to these as *second-tier features*. The distinction between basic and second-tier features is that basic features are the elements that can be extracted from a word (e.g. affixes), whilst second-tier features are the morphological properties of the word (gender, number). All are equally seen as features by the classification system, since the second-tier feature Gender can also be used to classify another second-tier feature Number. To differentiate between the reference of a morphological property in its linguistic sense and a second-tier feature, the latter are written with an initial capital letter (Gender), whilst the former are written in the normal way (gender).

In adopting a classification approach and seeing morphological analysis as a sum of multiple morphological features, an analyser can be viewed as a cascade of classifiers, where each classified feature provides information to the remaining classifiers in the cascade. Figure 4.1 depicts an abstract view of this concept. This type of approach would be valid only if there is indeed some form of dependency between second-tier features. If, on the other hand, there is no dependency between second-tier features, then the cascade would perform no better than a group of independent classifiers that use only the basic features as input. Through the implementation of the cascade classifiers, we investigated whether there is dependency in the different morphological properties in Maltese, and to what extend this dependency is worth taking into account in a morphological analysis system.

In summary, the research not only sought to learn a classification system to label words with their morphological properties, but also to analyse the level of dependency between the morphological properties. From a computational perspective, there are several techniques that can be used as classifiers. The experiments carried out explored various techniques, including decision trees, logistic regression and Naïve Bayes. Decision trees and logistic regression gave very similar results in terms of F-measure, with the difference that decision trees ran much faster, whereas logistic regression sometimes suffered from being

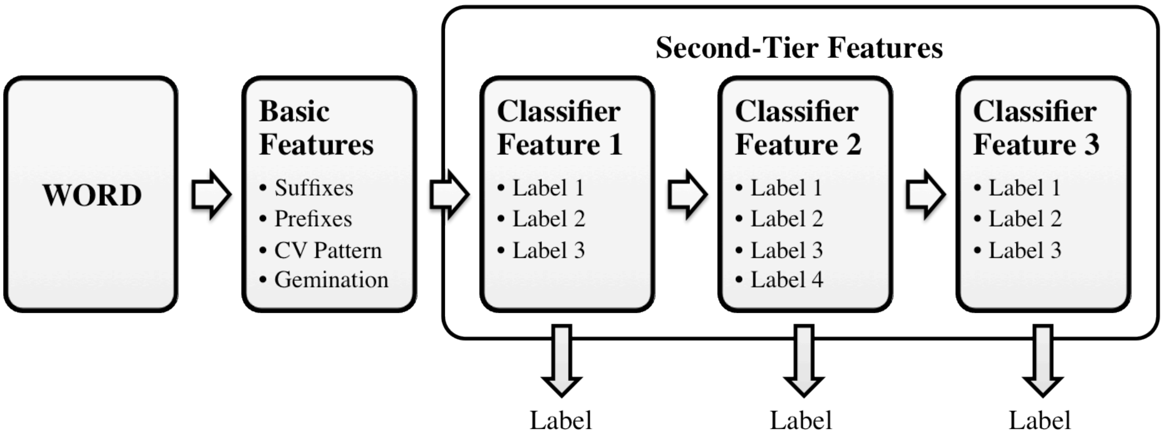


Fig. 4.1 An abstract system showing a cascade of classifiers each outputting a label and feeding the information into the next classifier

stuck in local maxima points. Naïve Bayes produced a slightly lower F-Measure than decision trees but has the added advantage that it associates a probability to the predicted classification. This is relevant in morphological analysis, especially in the cases of ambiguity when a word can have two different meanings. Actually, in inflection this is called a syncretism, i.e., when one form occupies two or more cells in a paradigm. For example, *tikser* √KSR 'break', can refer to both 2SG and 3SGF. A desirable feature of a morphological analyser is that, in such cases, it provides both options as labels to a word, which can then be disambiguated at a later stage when the context of the word is taken into account.

4.2.1 Data sources and preparation

A classifier requires labelled data in order to train a model which can then generalise and predict labels for unseen instances. Labelled data can come in different formats and usually needs to be preprocessed before being fed into a learning algorithm. Table 4.1 shows some random examples from the English dataset taken from the 2010 Morpho Challenge competition[3]. The words in this dataset are segmented, parts are labelled and if more than one analysis is available, they are given separated by commas. However, note that some of the morphemes are left out entirely and replaced by labels. This is standard in the output of a morphological analyser.

[3] http://research.ics.aalto.fi/events/morphochallenge2010/datasets.shtml

Table 4.1 Examples of labelled data from the Morpho Challenge English dataset 2010 competition

Word	Segmentation and labelling
misunderstanding	mis_p understand_V +PCP1, misunderstanding_V
mathematician's	mathematics_N ian_s +GEN
overbalanced	over_p balance_V +PAST
defeats	defeat_V +3SG, defeat_V +PL

Morphologically labelled data for Maltese comes from two separate sources. The first is the lexicon *Ġabra*, which contains roots and lexemes and their inflections. The second is an ongoing project which is creating an online Maltese-English dictionary[4] (referred to from now on as the dictionary project or the dictionary data). The principal distinction between the two datasets is that Ġabra focuses on inflectional words, particularly verbs, whilst the dictionary project focuses on derivationally related words. The dictionary project used a scanned version of the Aquilina dictionary (Aquilina, 1987–1990) and developed a program to extract the dictionary entries from the text version. Through this thesis, the program was extended to extract also derivationally related words and their morphological properties. On the other hand, Ġabra was based on the work of Spagnol (2011), which analysed templatic verbs in Maltese and exhaustively categorised 1,932 roots and over 4,143 verbs. Camilleri (2013) used the roots and their categories and created a grammar to generate all the inflective forms for each root. The resulting wordforms, together with the morphological information, are stored in a database as a lexicon and an online interface is available to query the database. Due to the different content of the two resources, these were used at different phases of the project. The Ġabra collection was ideally placed for the initial experiments in verbal morphology, whilst the dictionary data was essential for experiments on the noun and adjective categories. The focus in this section is on the Ġabra data and how this was used in context of the experiments carried out in verbal morphology.

The Ġabra database was primarily a collection of verbal inflections, with each word having various morphological labels associated with it. Since Ġabra was automatically generated through the description of a grammar, there were cases of over-generation or

[4] http://www.dizzjunarjumalta.com

errors in the word formations.[5] A snapshot of the database was taken in March 2014 and used throughout this thesis. The data consisted of 1,928 roots, 10,471 lexemes and 4,773,039 wordforms.

The morphological properties available in the Ġabra data were structured as follows. The lexemes table contained the lemma, general part-of-speech category (shown in table 4.2), the form, the radicals, and whether the entry is transitive, intransitive and ditranstive. This information was then automatically associated to all the generated wordforms from the lexeme. The wordforms contained further grammatical information: gender, number, person, subject, perfective/imperfective (tense/aspect), imperative (mood), direct object, indirect object and polarity.

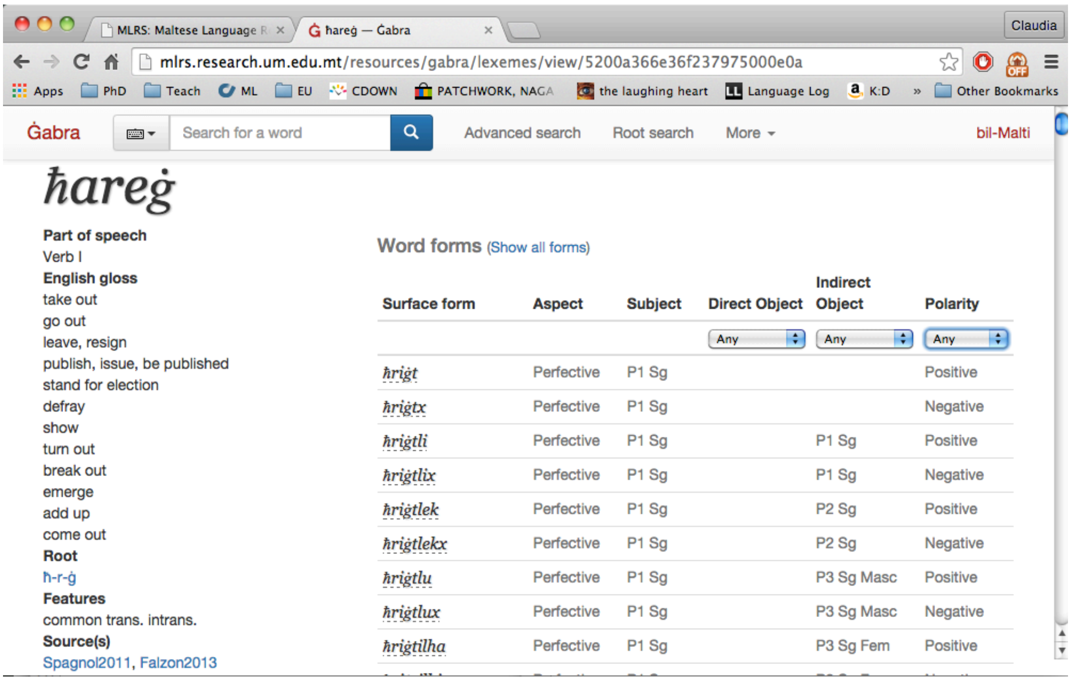
Table 4.2 Distribution of the lexemes over their part-of-speech tags as classified in the Ġabra database

	Lexemes
Total	10,471
Nouns	4,668
Verbs	4,746
Adjectives	251
Determiners	7
Prepositions	19
Adverbs	21
Pronouns	5
Proper nouns	13
No POS tag	724

Although table 4.2 shows a large number of lexemes for nouns, most of the wordforms in the database belonged to the verb category, which accounted for over 99% of the wordforms in the Ġabra database. Therefore these data are being used solely to investigate the verb category, especially since it is the richest category in terms of morphological properties.

Another important characteristic about Ġabra is that the morphological information is associated with a word as a whole rather than with the segmented word. This is shown in fig. 4.2 which displays a screenshot of the online interface. Each word is listed and, to the

[5] The Ġabra collection continued to be updated and improved over time. However, for the purpose of this research, a snapshot was taken in March 2014 and used throughout this thesis. Any reference to the status of Ġabra describes the status of this snapshot, and might no longer be relevant due to the improvements carried out since then.



ħareġ

Part of speech
Verb I
English gloss
take out
go out
leave, resign
publish, issue, be published
stand for election
defray
show
turn out
break out
emerge
add up
come out
Root
ħ-r-ġ
Features
common trans. intrans.
Source(s)
Spagnol2011, Falzon2013

Word forms (Show all forms)

Surface form	Aspect	Subject	Direct Object	Indirect Object	Polarity
			Any	Any	Any
ħriġt	Perfective	P1 Sg			Positive
ħriġtx	Perfective	P1 Sg			Negative
ħriġtli	Perfective	P1 Sg		P1 Sg	Positive
ħriġtlix	Perfective	P1 Sg		P1 Sg	Negative
ħriġtlek	Perfective	P1 Sg		P2 Sg	Positive
ħriġtlekx	Perfective	P1 Sg		P2 Sg	Negative
ħriġtlu	Perfective	P1 Sg		P3 Sg Masc	Positive
ħriġtlux	Perfective	P1 Sg		P3 Sg Masc	Negative
ħriġtilha	Perfective	P1 Sg		P3 Sg Fem	Positive

Fig. 4.2 The Ġabra online interface showing how the morphological features are displayed online

side, a template of the morphological properties provides the information for each word. It could be possible to reverse-engineer the rules used for the generation of the wordforms and try to associate the labels to the separate morphemes, resulting in something similar to the example given previously for English in table 4.1. However this would be a laborious task that would reduplicate the effort carried out on the rule production. Instead, the aim of this research is to inquire to what extent it is possible to learn all the morphological features found in the Ġabra dataset by extracting as many basic features as possible from the word itself. Thus, the remainder of this section will mainly focus on the extraction of basic features from words and the preparation necessary to port the Ġabra data into the necessary representation for the classification system.

4.2.2 Feature specification

The features and their possible values that were used by the classification system required to be specified in advance. These must be fixed, and once a classifier is trained, it only accepts data in the format and content that it recognises. This means that the specification of the basic and second-tier features must be representative of both the training data and possibly unseen future data. If, for instance, a new suffix *-s* were to appear in future morphological formulations, and this was not part of the training data that was used to model the classifiers, the system would not be able to classify that instance.

The starting point was the list of words themselves. The basic features are those characteristics that can be extracted automatically from a word. On the basis of the data available from Ġabra, described in §4.2.1, the features that could be extracted automatically from a word were its **affixes** and **composite suffixes** (the procedure is described further on in §4.2.3), the consonant-vowel pattern of the word (**CV-pattern-word**) and the stem (**CV-pattern-stem**) and whether a word has a **geminate** consonant or not. For instance, gemination of the middle consonant of a triliteral verb produces a derived so-called 2nd form. Thus, from the 1^{st} form of the verb *qasam* 'split/share', the 2^{nd} form is produced through the duplication of the middle consonant, deriving *qassam* 'distribute'. The CV-pattern of a word or stem was produced using a word-to-phoneme transcription method developed by Borg et al. (2011) for a Maltese speech synthesiser. For example, the phonological transcription of *ksirtekx* is [ksɪrtɛkʃ], which then produced the pattern ccvccvcc. Together, these formed the basic features which were extracted for every instance/word in both the training and testing data put together from Ġabra.

Ġabra also contained morphological information for each word, such as person, number, direct object, tense, aspect, mood, etc. These are the second-tier features — the morphological properties of a word that the classification system should be able to provide information for. There were a number of second-tier features in Ġabra and in order to keep the list of features compact, the properties *tense/aspect* and *mood* were joined into one single feature, abbreviated to **T.A.M.**. These three features are mutually exclusive, and in the database by Camilleri (2013), they are listed under a single field. Thus, this decision was taken to retain consistency with Camilleri. The second-tier features found in the database are listed in table 4.3, together with all their possible values. The null value was used when the dataset does not contain particular information for a specific word.[6] The features marked in asterisks (frequency, not duplicate, transitive, intransitive and ditransitive) were discarded for two reasons: (i) gaps in the values of the data, since the 'null' value does not make it fully clear whether it should be considered as false or not; (ii) the

[6] The choice of maintaining null as a feature value was due to the design of the Mongo database from which the data was ported. In Mongo, for instance, a boolean field can be left null. This means that it is neither true, nor false, but rather not known at that stage. When porting the data to MySQL all null values were retained, and these were transferred to the representation of the datasets even if the value was at times redundant.

amount of positive examples was too small to be statistically significant for the feature to be learnt. The radicals feature (referring to the root consonants of a word, such as √ksr for *kiser* 'break') was not used as a second-tier feature because it is specific only to words of Arabic origin. In Maltese, the role of the radicals in the morphological system is a matter of some debate. For example, while evidence has been found that roots are implicated in the mental representation of Maltese verbs (Ussishkin et al., 2015), readers of Maltese texts exhibit far less sensitivity to the order of radicals than their counterparts in Arabic or Hebrew (Perea et al., 2012). Furthermore, template morphology is known to be unproductive in Maltese (Hoberman and Aronoff, 2003). The psycholinguistic evidence corroborates a view sometimes articulated in the theoretical and descriptive literature (Fabri, 2009), that the root may play an important organisational role in the Maltese lexicon, but may be less crucial to morphological processes than it is in other Semitic languages. For example, many verbs in Maltese are now inflected on the basis of processes historically derived from Romance (see Mifsud, 1995b).

Table 4.3 Second-tier features and their relative values in the Ġabra database

Feature	Values
Person	1, 2, 3, null
Gender	M, F, null
Number	Sg, Pl, null
Form	0 – 9
Subject	1Sg, 2Sg, 3SgM, 3SgF, 1Pl, 2Pl, 3Pl, null
T.A.M.	Perf, Impf, Imp, null
Direct object	1Sg, 2Sg, 3SgM, 3SgF, 1Pl, 2Pl, 3Pl, null
Indirect object	1Sg, 2Sg, 3SgM, 3SgF, 1Pl, 2Pl, 3Pl, null
Polarity	Pos, Neg, null
Verbtype	strong, geminate, weak-final, weak-medial, weak-initial
POS*	N, V
Frequency*	Common, null
Radicals*	various values
Not duplicate*	True / False / null
Transitive*	True / False / null
Intransitive*	True / False / null
Ditransitive*	True / False / null

The features verbtype and form were retained even though they are highly specific to the Ġabra data. The form was retained as an experiment to analyse how well a classifier would learn this feature on the basis of the CV-patterns of the word and the stem. Generally, each form represents particular derivative patterns and it would be interesting to see to what extent this feature could be learnt and generalised. The feature verbtype was used as part of the segmentation process, described in §4.2.3 below, and also used for the specification of the basic feature geminate. It was included in the second-tier feature list to analyse if there is dependency on this feature in terms of morphological information, and whether the feature itself could be modelled appropriately or not. The features person, number and gender in verbs are represented as the single feature subject in Ġabra. All features are initially left in the data and some experiments are carried out with these features to find the best representation possible.

A classifier was trained to learn each of the second-tier features that were included in the classification system. The system as a whole used the individual classifiers as a cascade, each producing the appropriate label and passing on the information learnt to the following classifier. The labels were restricted to those identified in table 4.3 as the expected output from the classification system.

4.2.3 Segmentation and composite suffixes

The primary basic feature that could be extracted from a word is its affixes. The segmentation technique used for the clustering of morphologically related words (§3.2.4, page 62) used transitional probabilities to calculate a ranked list of the most likely affixes. It then looked at all possible valid segments using the top 10% ranked prefixes and suffixes, with the clustering technique using all valid segmentations proposed. If the same list of affixes were to be used to segment the words in Ġabra, the first problem encountered would be that the majority of generated wordforms contained in Ġabra use affixes that were not included in the list. Several wordforms in Ġabra are not commonly used in everyday Maltese, and therefore are not attested in the MLRS corpus. This is quite normal since no corpus will cover the full range of a vocabulary and, furthermore, Zipf's law predicts that a significant proportion of words would be hapaxes or unseens. If the same affix list had to be used, a large number of words in Ġabra would remain unsegmented since their affixes, and in particular the suffixes, would not be part of the list. Another possibility would be to use semantic similarity to determine whether a word is related to its potential segmented stem, in which case both words would need to be attested in the MLRS corpus for their similarity to be calculated. The possibility to rerun the technique to discover affixes on the Ġabra wordlist was also considered. However, this would still result in a segmentation process that would propose multiple valid ways of segmenting a word. Table 4.4 specifically shows how this segmentation strategy would result in at least 4 different segmentations for the word *tkissirtx* ‘be broken-Perf.1|2.Sg.Neg’ (form V √ksr), when ideally the system proposes only one definitive segmentation (marked in bold). A single segmentation for every word was a requirement for the labelling task since the data is used to model a classifier.

Table 4.4 Previous segmentation strategy showing the valid segmentation options for the word *tkissirtx*

tkissirtx	tkissirt-x
	tkissir-tx
	t-kissirt-x
	t-kissir-tx

The advantage of the Ġabra database was that the information of the lexeme and the root were available in the database for each wordform. This means that it was possible to easily extract the affixes through a simple rule-based system that detected which part of the word is the stem. This approach was possible for the majority of the wordforms, especially those that have a *strong* root (this was marked by the verbtype in the database). *Weak* roots are those which have the consonants *w* or *j* as part of the root — for example the root √Ħlj produces *ħela* ‘waste-Perf.3.Sg’, *ħlejt* ‘waste-Perf.1|2.Sg’ and *ħlew* ‘waste-Perf.3.Pl’ Strong triliteral roots never have *w* or *j* as part of the root, and all the root consonants are always present in the word formations.

The approach for the segmentation of the Ġabra wordlist was therefore carried out on the basis of the classification of its root. Words with strong roots were segmented through a rule-based approach that looks at where in the word the radical consonants occur. It also takes into account the possibility of gemination, where one of the radical consonants is duplicated in the word formation process (e.g. *kisser* from √ksr ‘break’, duplication occurs for the second consonant producing a form which intensifies the meaning of the word *kiser*). In the case of words produced from roots with weak consonants, these are left unsegmented. This allows a small amount of ‘noise’ in the training data, reflecting the reality that the segmentation of unseen data is not always known.

Maltese morphology, especially verbal morphology, makes high use of concatenative composite suffixes, for example:

ksir	*-ni*	*-hu*	*-lek*
break	-Perf1Pl	-DO-3SgM	-IO-2Sg

‘we broke it for you’

Clearly, composite suffixes contain important morphological information. We posit that segmenting composite suffixes and including them as part of the basic features would provide added value to the classification as a whole. The question is to what extent the classification system would be impacted by having the composite suffix broken down further versus having only the suffix as a whole. From the above example, it is also possible to note the order of suffixes and their positioning, generally as (i) subject, (ii) direct object, (iii) indirect object and finally (iv) polarity (when applicable). The ordering of composite suffixes is important in Maltese morphology. This is similar in concept to that proposed by Plag and Baayen (2009) on English suffix ordering.

Two experiments were carried out to examine different representations of composite suffixes in the dataset, with the aim to measure the added value that composite suffixes provides to the classification system. However, a description of the technical setup of the machine learning system will follow in §4.2.4, before proceeding to the analysis of these experiments in §4.2.5 and further results in §4.3.

4.2.4 Machine learning algorithms and experiment setup

Having identified the data, the features that can be extracted (basic features) and learnt (second-tier features), the focus turns to the different techniques that can be used. A classifier is an algorithm that builds a model of the distribution of features in training data and their joint prediction of the class. A class is the value that a particular feature has. In this case, most features have more than two classes (e.g. Gender can be classified as Feminine, Masculine or Neutral). Once trained, new instances will be classified on the basis of the learnt model. One of the most common classification problems is that of detecting spam email — a classifier is given several examples of what is and what is not spam, builds a model, and then decides whether a new email is spam or not. Some classification systems continue to learn, by allowing users to provide feedback on the predictions that it makes and modifying its model according to the feedback given. The tendency in machine learning is to try different techniques (possibly different parameters) to analyse the output of the results and assess which technique fits the learning problem best. Below follows a very short description of the techniques used in this work. A more complete overview of machine learning techniques and classification systems can be found in various sources, amongst them Hastie et al. (2009).

A **decision tree** specifies the sequence or process of how a decision is taken, using a divide-and-conquer approach. The training process of a decision tree aims at finding the best sequence of how the different attributes are tested and on what basis sub-decisions are taken, finally arriving to a final decision which results in the final classification of an instance. The main advantage of a decision tree is that the output of the model is generally easy to understand (depending on the size) and easy to implement as a sequence of IF-THEN-ELSE statements. One of the pitfalls of a decision tree is that it arrives to a single classification without associating a probability to it. This might hinder the classification of an instance where there is ambiguity. For example, *tikser* is both 2SG and 3SGF, but a decision tree can only pick one label and cannot produce two labels with a likelihood for each label. It also might create an overly complex tree that does not generalise well over the training data, a problem known as overfitting — thus the model learnt is too close to the training data and would have poor predictive performance on unseen data. **Random forest** is an extension of the decision tree algorithm, by which a number of trees are constructed during training, and it outputs the class predicted by the majority of trees. This corrects for the overfitting problem of the decision tree, having a more 'diverse' classification possibility.

Logistic regression is a discriminative probabilistic classification model which aims at finding the probability distribution that best represents the training data, resulting in the classification which has the highest expected value. The main advantage of this technique is that it provides a probability for each of the classes in a classifier. So in practical terms, logistic regression would provide probability values for both labels 2SG and 3SGF for a word like *tikser*. Its main disadvantage is that it is parameterised and must be set according to the type of data it is trained on. This might not always be straightforward, and logistic regression might also take more time to train, so tweaking the parameters could prove to be time consuming.

Naïve Bayes is a probabilistic classifier based on the premise that each feature contributes independently to the classification of an instance. This particular aspect of Naïve Bayes is what makes it a very fast algorithm and it generally performs very well for a number of classification problems. The independence assumption is also a drawback if the likelihood with which a feature corresponds to a class in fact depends on one or more other features. Again here, the classification has a probability associated to it, so it is possible to check for cases of ambiguous classifications when the probability of two proposed classes is close enough.

Support vector machine (SVM) is an algorithm that analyses the training data and recognises patterns of every class in the data, making it a non-probabilistic technique. This means that the prediction will always output a single classification. SVMs can be computationally intensive during training, and are dependent on the choice of the kernel used and its parameters (which is the part of the algorithm that does the pattern analysis). Some kernels can overfit the training data, providing poor prediction performance. Another issue is that the data should normally be linearly separable — meaning it should be possible to draw a hyperplane separating instances of different classes. That is something that can only be discover by testing the data and different kernels. However, SVMs are quite popular in machine learning classification problems since they allows the user to find the appropriate kernel for the particular type of classification problem being learnt.

Finally, the **Majority Class** classifier is the simplest classification technique which takes the majority class as its prediction value. So for example, if there are more instances in the training data for singular than plural, then the prediction of any unseen instances will always be singular. Of course, it is clear that this is an overly simplistic algorithm. However it is used as a baseline for all the above algorithms. In other words, for an algorithm's performance to be considered as acceptable, its minimum requirement is to perform better than the majority class algorithm.

Since the classification system took a cascade approach, a **benchmarking** system was used as a result. Each classifier for the second-tier features could be learnt with only the basic features. But as the cascade progressed, more second-tier features were added, providing more information to the classifiers further down the line. The performance of any second-tier feature classifier should be improved through the information provided by the preceding second-tier feature classifiers. In the case that a classifier did not improve, this meant that the classifier for that second-tier feature did not require input from other second-tier feature classifiers to aid it with its classification. The **floor benchmark** is defined as the performance of a second-tier feature classifier that uses *only* the basic features as input. The **ceiling benchmark** is defined as the performance of a second-tier feature classifier that uses the basic features and *all* other second-tier features as input. The floor benchmark represents a classifier positioned at the beginning of the cascade, and the ceiling benchmark represents the positioning of a classifier at the end of the cascade. These benchmarks were used to investigate and derive the optimal sequence of the second-tier feature classifiers in such a way that the resulting classification cascade maximises the overall improvement of each classifier from the floor benchmark, bringing it as close as possible to the ceiling benchmark.

The above machine learning techniques were all implemented through the WEKA data mining software (Hall et al., 2009), available both through a graphical user interface and as an open-source java library. The training and testing datasets were formatted according to the specification expected by WEKA. One of the main advantages of using WEKA is the standard presentation of results between the different algorithms and the way all algorithms can be evaluated. It provides accuracy for the overall classifier, as well as F-Measure for every class (label) available[7]. It is also possible to use ten-fold cross validation or to test a model on completely unseen data. The primary aim of this investigation was to determine the optimal sequence of cascade classifiers, and then to perform a final evaluation on completely unseen data. The initial analyses of the classifiers' performance described below was measured using a 10-fold cross validation evaluation over the training data itself. The final evaluation used completely unseen data. To ensure clarity, the latter is always labelled as UNSEEN in the remainder of the report to distinguish between the two. Generally, the initial 10-fold cross validation performs better than the unseen because the evaluation is using the same data used to model the classifier.

Initial experiments explored various algorithms described above, with a particular focus on decision trees, logistic regression and Naïve Bayes. The latter performed best in terms of speed, learning a model within a minute or two. Decision trees took slightly longer to compute a model, usually under five minutes, and produced better results in terms of accuracy and F-Measure over Naïve Bayes. On the other hand, logistic regression took the longest time, at times over 24 hours, and did not always converge to a solution. Several experiments were carried out to tweak the parameters, resulting in some improvements in execution time. However, when logistic regression did converge, the results were very close to those obtained using decision trees. Therefore, the analyses below present the results obtained using decision trees. An comparison of the different machine learning techniques is described later at the end of §4.3, where the performance of the different techniques is discussed in more detail.

4.2.5 Data representation

4.2.5.1 Composite suffixes

A number of experiments were carried out to determine how best to represent the data, how to choose the training and test instances from the 4.7 million wordforms, and the representation of the split composite suffixes described in §4.2.3 above (page 101). The representation of the split composite suffixes as part of the basic features had to have a fixed number of placeholders to facilitate the computational processing of this data. Two experiments were carried out in order to decide which type of representation would be

[7] In order to clearly distinguish between the two, unless otherwise specified, accuracy is expressed as a percentage (e.g. 98% or just 98 in the plots of graph figures), whilst F-Measure is expressed as a decimal between 0 and 1 (e.g. 0.98). This ensures consistency throughout the analysis of the results and figures.

best. In each experiment, the last placeholder was used solely for the *-x* suffix which always comes at the end of a word and represents the negation of a word. The first experiment used a total of 5 placeholders, and the split composites were placed in order from right to left, leaving the split composites *close* to each other (due to the last placeholder being used by the negative). A second and more intuitive experiment used 4 placeholders and the split composites were placed from left to right. The difference between the two representations is best illustrated through an example, shown in table 4.5. The suffix *-t-lek-x* usually denotes either -Perf1Sg-2Sg-Neg or -Perf2Sg-2Sg-Neg.

Table 4.5 An example of the different representation for a split composite suffix

-tlekx	S1	S2	S3	S4	S5
Experiment 1			t	lek	x
Experiment 2	t	lek		x	NA

The automatic technique of segmenting composite suffixes was discussed in §3.2.3 and shown to have an accuracy of 52%. Given the importance of composite suffixes in terms of the morphological information they contain, it was important to have a more accurate segmentation of suffixes. A small rule-based system was therefore implemented to segment 128 composite suffixes. A dataset was created by randomly selecting 30,000 words. These instances were processed into two separate training sets with the different split composite suffix representations shown above in table 4.5. Figure 4.3 shows the F-Measure obtained by a decision tree in classifying different second-tier features using the different representations. The majority of the second-tier features are better classified in the second experiment, when 4 placeholders were used and composites were placed from left to right.

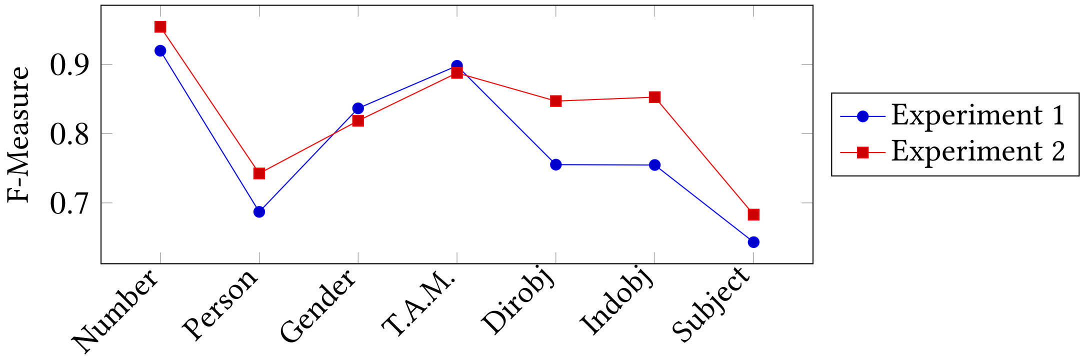


Fig. 4.3 The results of two experiments with different composite suffix representation

The following experiment aimed at assessing the importance of actually having the split composite suffixes as part of the basic features. Two datasets of 30,000 wordforms were used, but with a different distribution in terms of whether the segmentation of the composite suffixes was known or not. The distribution is shown in table 4.6.

Table 4.6 Distribution of wordforms according to the segmentation of the composite suffixes

Dataset name	Composite suffixes distribution
30A-Split	10K words not segmented, suffixes not known 10K words segmented, suffixes known, composite suffix not split 10K words segmented, suffixes known, composite suffix split
30B-Known	30K words segmented, suffixes known, composite suffix split

The rationale behind the two datasets was that 30B-Known represented the ideal scenario where all suffixes could be segmented correctly into their composites. 30A-Split represented a more realistic scenario, where the system does not have enough information to segment all words or split suffixes into their composites. The F-Measure on the two datasets is shown in fig. 4.4. As expected, having more composite suffixes within the data improves the performance of the different second-tier feature classifiers, and practically all classifiers performed better when the dataset consisted only of words where the split of the composite suffixes was known. Given these results, the number of rules to segment suffixes was increased from 128 to 214, ensuring a better coverage of the different morphological aspects contained in the composite suffixes.

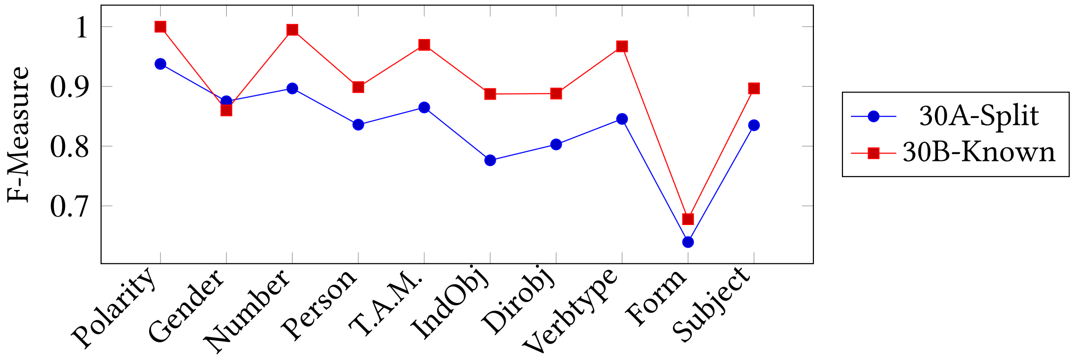


Fig. 4.4 F-Measure for two datasets having different different distribution in suffixes and composite suffixes

4.2.5.2 Feature distribution

The second-tier features obtained different results, and some features were seemingly easier to learn than others. In analysing the datasets and the results, it transpired that particular values of a second-tier feature were under-represented. This resulted in a *skewed* model, favouring the better-represented classes. Generally, since the word formation in Ġabra is automatically generated, the distribution between the features and their values was rather well represented. However, selecting a purely random dataset from 4.7 million words does not necessarily bring forward the same distribution present in Ġabra. Thus to ensure a better representation of the morphological features present in Ġabra, 200 lexemes were randomly selected and the final representative dataset was created by extracting all the wordforms of those lexemes, resulting in a dataset of 173,994 wordforms.

Further analysis of the selected wordforms showed some errors in the automatic generation of some of the words related to the imperative (mood under T.A.M.). Although different productions were included in terms of morphological information, the actual words produced were always the same. This means that the data contained, for example, *oħroġ* for both Imp2Sg and Imp2Sg-1Sg, when the correct word for the latter production should be *oħroġni*. Further examples produced from *ħareġ* 'go out' are shown in table 4.7.

Table 4.7 Some examples of the imperative for √Ħrġ - *ħareġ* 'go out' as found in the Ġabra database

Database word	Subject	DirObj	IndObj	Correct word
oħroġ	P2 Sg	null	null	oħroġ
oħroġ	P2 Sg	P1 Sg	null	oħroġni
oħroġ	P2 Sg	null	P1 Sg	oħroġli
oħorġu	P3 Pl	null	null	oħorġu
oħorġu	P3 Pl	P1 Sg	null	oħorġuni
oħorġu	P3 Pl	null	P1 Sg	oħorġuli

From a machine learning perspective, this type of error impacts the learning of the direct object and indirect object due to the lack of the appropriate pronominal suffixes. The most practical short-term solution for this dataset was to use only those wordforms in the imperative that had both the direct object and indirect object set as null. This meant that from the original 173,994 wordforms, the training dataset was reduced to 151,382 wordforms. The number of wordforms which were segmented was 126,306 (83%) and 75,937 (50%) had their composite suffixes split.

A separate unseen dataset was put together to test the different classifiers learnt. For this purpose, 20,000 wordforms were selected randomly with one restriction — wordforms chosen must not be present in the training dataset. A slight adjustment was done to all wordforms in the imperative, whereby the direct and indirect object fields were set to null, reflecting the imperative in the training dataset.

4.3 Experiments and results

A series of experiments were carried out to decide the order of the second-tier feature classifiers such that the cascade classification system would obtain the best results possible. The system uses the morphological information learnt from previous classifiers to feed into the following classifiers. This section discusses the approach adopted to select the sequence of the classifiers for the cascade and the results obtained on the unseen dataset. Some further experiments are also described in which some second-tier features which are highly co-dependent are merged together as a single feature, investigating whether it is possible to improve the prediction of such features when merged.

The order in which features were classified also reflects the dependencies between the second-tier features. Some of the basic features were also more relevant to classify certain second-tier features than others. The results reported below cover the most interesting aspects and the overall results achieved by the classification system for the verb category.

4.3.1 The optimal cascade sequence

An optimal cascade of classifiers provides a sequence through which the best possible results are obtained by the individual classifiers as well as collectively. Two approaches were used to find the ideal sequence that seeks to maximise the collective performance of the classifiers. The first used information gain to assess which features provided added value to the classification of other features. The second approach used the floor and ceiling benchmarks of every feature to find the best sequence that maximises the overall improvement, bringing the performance of every classifiers as close as possible to the ceiling benchmark.

Information gain is a score that ranks the most predictive features for a particular classifier. This provided insight as to which features give most information in order to best classify a second-tier feature. Table 4.8 shows the information gain for each of the second-tier features (polarity, indirect object, direct object, number, T.A.M., gender, person, verb type, form) when each one is predicted from a combination of the basic features and each of the second-tier features. The last column (labelled Cont.) provides the horizontal sum of the information gained per feature. The two features that are most useful in the overall classification of the second-tier features are the suffix and the cv-pattern of the word, followed by the prefix and the composite suffixes. The columns look at the individual second-tier features and provide an overview of which features best contribute to its classification. For instance, the classifier for the second-tier feature direct object receives most information gain from the suffix, s1, s2, cv-pattern word and indirect object. The last rows show the total information gain per second-tier feature (Total IG), which is then split to show the information gain from basic features (Basic IG) and from second-tier features (2^{nd}T IG). These last two rows provide particularly interesting insights into which of the second-tier features depend mostly on other second-tier features. This becomes more evident when comparing across features. For instance, the feature polarity gains most of its information from the basic features when compared with other features — in fact, it has the lowest sum for the information gained from second-tier features. On the other hand, the second-tier feature which has the highest gain from other second-tier features is gender, gaining most from the features number and person. This type of analysis indicates that a feature like polarity can be classified early in the cascade, whilst a feature like gender should come later. However, the information gain on its own was not sufficient to determine the actual sequence the classifiers should be in. It simply provided a ratio of the amount of information that a classifier gained from the different features.

The second approach was more systematic, through a series of extensive experiments. For each second-tier feature, we built two classifiers: (i) using only the basic features, referred to as the floor benchmark; (ii) using all basic and second-tier features, referred to as the ceiling benchmark. One classifier represented the lowest expected performance and the other represented the highest expected performance[8] of a second-tier feature classifier. The experiments evolved in an iterative approach, at each stage analysing which of the second-tier classifiers was closest in performance to the ceiling benchmark. When a classifier was identified, it was placed into the cascade. The remaining second-tier classifiers were then retrained for further analysis, this time including any second-tier features already placed in the cascade. This process was repeated until all second-tier classifiers were placed in the cascade.

[8] The ceiling classifier might not represent the best performance necessarily. Imagine a scenario where a feature actually interferes with the classification of another feature, thus our approach would result in a worse ceiling classifier. However it was evident from the analysis of the information gain that if there was such a case within the second-tier features, the difference would be rather insignificant.

Table 4.8 Attribute selection using information gain ratio for the second-tier features

Feature	POL	IND	DIR	NUM	TAM	GEN	PER	VER	FORM	Cont.
Prefix	0.00	0.00	0.00	0.10	**0.63**	0.26	0.43	0.42	0.40	2.23
Suffix	**0.84**	**2.46**	**2.04**	**0.80**	0.44	0.49	0.41	0.51	0.11	**8.09**
S1	0.08	0.44	**0.78**	0.46	0.23	0.24	0.14	0.12	0.01	2.50
S2	0.09	**0.90**	**0.56**	0.26	0.12	0.17	0.04	0.11	0.01	2.27
S3	0.02	0.39	0.10	0.23	0.09	0.13	0.04	0.11	0.01	1.12
S4	**0.50**	0.05	0.01	0.16	0.03	0.07	0.00	0.10	0.01	0.94
CVWord	**0.71**	**1.73**	**1.22**	0.40	0.37	0.26	0.24	**0.56**	**0.97**	**6.47**
CVStem	0.00	0.00	0.01	0.20	0.08	0.11	0.07	**0.73**	**1.16**	2.35
Gem1	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.05	0.06
Gem2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.26	0.39
POL	–	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.01	0.03
IND	0.00	–	**0.26**	0.00	0.01	0.00	0.00	0.00	0.00	0.27
DIR	0.00	**0.26**	–	0.01	0.01	0.00	0.00	0.00	0.00	0.28
NUM	0.01	0.00	0.01	–	0.01	**0.29**	0.03	0.00	0.01	0.34
TAM	0.01	0.01	0.01	0.01	–	0.00	0.01	0.00	0.01	0.05
GEN	0.00	0.00	0.00	**0.29**	0.00	–	**0.47**	0.00	0.00	0.77
PER	0.01	0.00	0.00	0.03	0.01	**0.47**	–	0.00	0.01	0.52
VER	0.00	0.00	0.00	0.00	0.00	0.00	0.00	–	0.10	0.10
FORM	0.01	0.00	0.00	0.01	0.01	0.00	0.01	0.10	–	0.13
Total IG	2.28	6.25	5.01	2.96	2.03	2.50	1.89	2.88	3.12	
Basic IG	2.25	5.98	4.73	2.62	1.98	1.73	1.37	2.78	2.99	
2^{nd}T IG	0.03	**0.27**	**0.28**	**0.34**	0.05	**0.77**	**0.52**	0.10	0.13	

Polarity was the first second-tier feature classifier to be placed in the cascade. This was also evident from the information gain analysis, since this feature depended solely on the basic features. Table 4.9 shows the process explained above which was carried out to decide which second-tier feature should be placed in the cascade. The analysis was based on the improvement achieved by the retrained classifiers, which now used the basic features and the output of the Polarity classifier. The classifier which achieves the closest F-Measure to its ceiling is the one for the indirect object feature. The process was again repeated with the classifiers retrained using the basic features and the output of the Polarity and Indirect Object classifiers, with the aim of finding the next feature according to the best possible F-Measure improvement.

Table 4.9 Analysis of F-Measure difference between the ceiling benchmark and the retrained classifier which included the basic features and the polarity second-tier feature.

2^{nd}Tier Features	Ceiling	Basic+Pol	Difference
Num	0.978	0.950	0.028
Gen	0.983	0.879	0.104
Tam	0.993	0.959	0.034
Ver	0.963	0.928	0.035
Ind	0.939	0.922	**0.016**
Dir	0.957	0.933	0.024
Per	0.866	0.747	0.119
Form	0.909	0.866	0.043

All the second-tier features were processed as detailed above with the exception of verbtype and form. These two features are very particular to the Ġabra dataset and specific to words formed from the root (implying Semitic origin). If the classification system were applied to a generic corpus such as the MLRS, there would be a large percentage of words that would not require the output of these two classifiers. With the data available, it was not possible to know whether the presence of these classifiers earlier in the cascade would negatively impact the classification of words of Romance or English origin. In a practical setting, these two classifiers could either be retrained and include a representative number of Romance and English-originating examples so as to offer a better coverage of Maltese, or else be left out entirely from the classification system. Both features do not offer information pertaining to the inflectional morphological properties of a word — which is mainly what the Ġabra data represents — but rather the structural pattern that would be used in the production of derivationally related words. Although the experiments were never designed to focus only on inflection, the limitation came from the type of data available through Ġabra. However, since this particular data has the two features at its disposal, it was interesting to analyse to what extent such features are classifiable. These two features were therefore placed at the end of the cascade so as not to impact the other second-tier classifiers in any way.

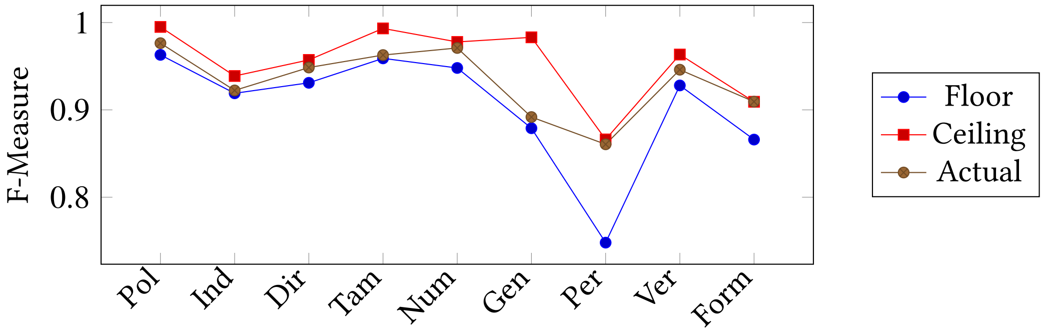


Fig. 4.5 The resulting cascade sequence, comparing the performance of each classifier to its Floor and Ceiling benchmarks.

The above analysis provided the resulting optimal cascade sequence as follows: **Polarity, Indirect Object, Direct Object, T.A.M., Number, Gender, Person, Verbtype and Form**. Figure 4.5 shows the performance of each classifier when it was actually included in the cascade classification system, and comparing the F-Measure with its floor and ceiling benchmarks. Each of the classifiers include the basic features and any preceding second-tier feature classifiers in the cascade. For the most part, the performance of the classifiers is very close to the ceiling, the only exception being the feature gender. The choice between classifying gender or person first is quite a close call, with both features achieving close-to-ceiling performance if the other feature is classified first. Gender wins the 'race' by about 0.005 difference in improvement if it is classified first. Figure 4.6 shows how the two features perform as more second-tier features are added to the cascade. In fact the only 'spike' in the curves is when either feature is included in the cascade. This shows that apart from the dependency on the basic features, Gender and Person are largely dependent on each other as second-tier features.

Additionally, fig. 4.5 above also shows those second-tier features that are highly dependent on information gained from other second-tier features. This was evident from the wider gap in performance between the floor and the ceiling benchmarks, notably for the features T.A.M., Gender, Person, Verbtype and Form. Other features such as Polarity and Indirect Object were more dependent on the basic features alone, demonstrating a closer gap between the floor and the ceiling benchmarks. For the most part this matched the anal-

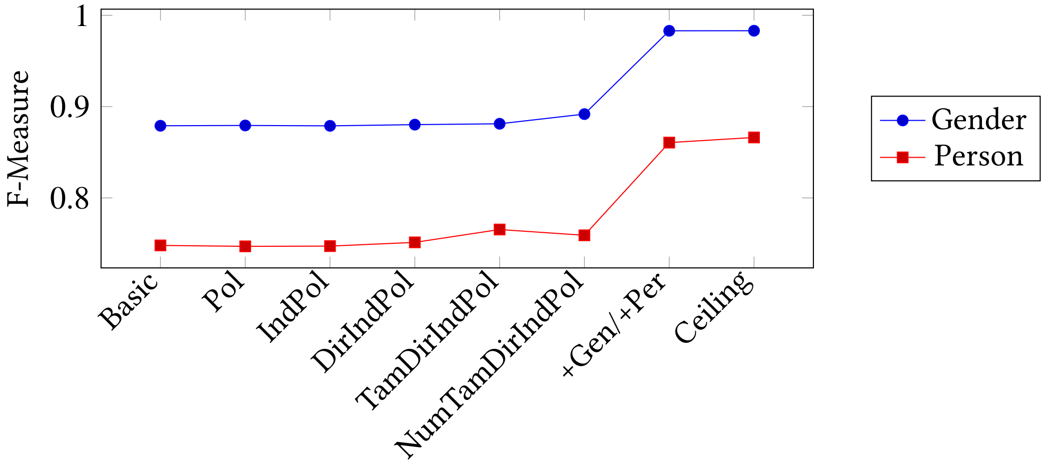


Fig. 4.6 Improvement curve of the Gender and Person features as more features are included in the cascade, with both reaching the ceiling benchmark when either feature is added first

ysis on the information gain ratio between the basic features and second-tier features, with the exception of Verbtype and Form. The information gain ratio for both these two features showed practically a full dependence on basic features, whilst in practice the classifier for either feature improved mostly when they are left to the end of the cascade. Figure 4.7 shows the performance curve for the Form classifier as it is trained using more second-tier features, with the final spike when all other features were included, hence reaching its ceiling benchmark.

The results justify the ‘experimental’ approach taken in determining the sequence of the cascade classifiers, rather than simply relying on the information gain ratio alone. By actually training a number of different classifiers and analysing their performance at different points of the cascade, it was possible to find the sequence which maximises the overall performance of the cascade classification system.

The remaining issue was that of circular dependancy of second-tier features, particularly between person and gender. The following part describes some modifications to the dataset and a series of experiments to analyse whether it would be possible to obtain improvements in the performance of the classifiers by merging together specific second-tier features.

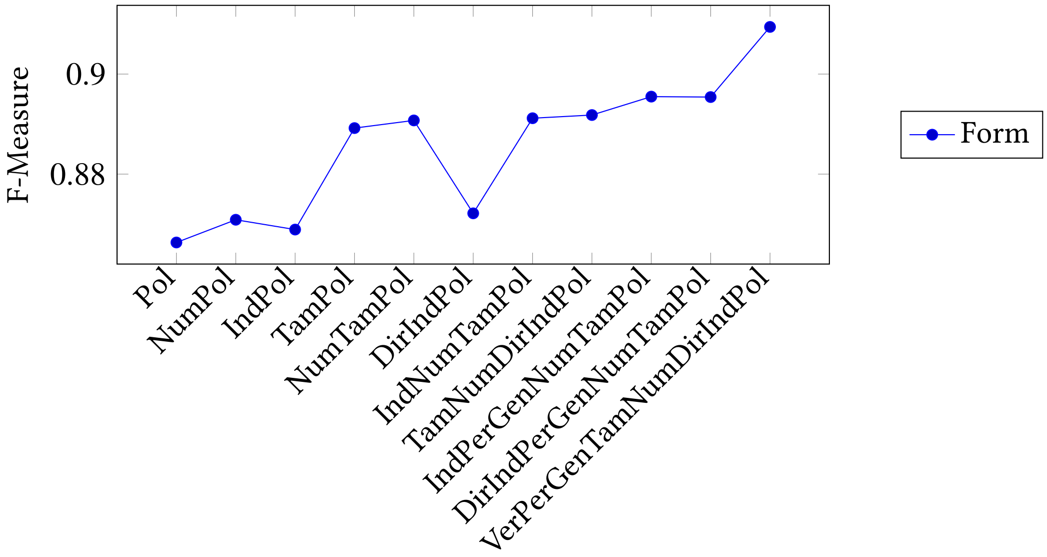


Fig. 4.7 The performance of the Form classifier as different second-tier features were included in the cascade

4.3.2 Merging second-tier features

Some of the second-tier features were clearly dependent on others to produce good results. This was evident from both the information gain ratio in table 4.8 as well as in the results obtained from the various classifiers produced in deciding the cascade sequence. The most obvious circular dependency was between gender and person, with each reaching the ceiling benchmark only if the other was introduced into the cascade (as shown in fig. 4.6 above). The information gain ratio table provided an overview of the main dependency as well — mainly between the direct and indirect object; number and gender; and gender and person. The dependency between direct and indirect object was relatively small, and in fact the gap between the floor and ceiling benchmarks for these two features is of 0.03 (direct object) and 0.02 (indirect object). This means that which one is classified first will not have a significant impact on the performance of the cascade. However, the difference between the floor and ceiling benchmarks was far more substantial for gender (0.10) and person (0.12). The dependency could also point towards those morphological properties which contain ambiguity. For instance *tikser* is ambiguous in both gender and person, being either 2Sg or 3SgF. In an ideal scenario, a morphological analyser would present both analyses and rely on a parser which uses context to determine the correct analysis of the word.

A possible approach to eliminate ambiguity in the dataset was to *flatten* the data when more than one possible analyses is available for a wordform. In practical terms, whenever the dataset contained more than one entry for the same wordform, the different second-tier features were merged into a single value. In this case, the entry for *tikser* would be labelled as 2|3.Sg.Null|F. The limitation of this flattened representation is that it might not be clear that the second person is neutral (null), and the third person is feminine, and that second person feminine is incorrect. To counteract this problem, second-tier features can be merged before the data is flattened. In this case, if the features person and gender are merged, then flattened, the analysis for *tikser* would be 2Null|3Fem.sg. The removal of ambiguity could result in an improved overall performance for the classification system, something which was worth investigating. The following modifications were made to the original dataset and each classifier was retrained with the modified dataset.

Flattened Dataset When more that one entry exists for a particular word, all entries are flattened into one instance. This means that *tikser* would have the following values for person: '2|3', gender: 'null|f', number: 'sg'. We refer to this dataset as **Flat**.

Merging Two Features This process merged two features together before flattening the data. We experimented with merging person and gender (**PerGen**), number and gender (**NumGen**), and number and person (**NumPer**). The analyses then looked at which pairing of features could remove ambiguity and minimise the potentially negative impact of feature dependency.

Merging Three Features Here the features person, gender and number are combined as a whole feature. Since the Ġabra data contained only verbs, this refered to the **Subject** of the word.

The experiments with these five datasets followed the exact same process carried out with the original dataset. For the sake of clarity, the original dataset is hereafter referred to as the **Regular** dataset. For each dataset a number of experiments were carried out to determine the optimal cascade sequence. Since the differences in the resulting sequences were minor and primarily due to the merging of features together, the presentation of results will use the same sequence as the second-tier classifiers with the regular dataset. In fact, the Flat cascade resulted in exactly the same sequence as the Regular cascade. However for the merged datasets, each cascade resulted in the feature-pair being inserted into the cascade at the very end. This hints at a certain level of complexity in learning merged features.

Beginning with the Flat dataset, fig. 4.8 shows the floor and ceiling benchmarks for the second-tier feature classifiers, as well as the performance of the actual modelled cascade itself. Simply by comparing the floor benchmarks of the Regular versus the Flat classifiers, shown in fig. 4.9, it is possible to see that by flattening out the data, the classifiers learnt perform better overall even just by using only the basic features. fig. 4.9 shows the floor benchmarks for the Regular versus the Flat cascade. Notably, the difference in F-Measure for the Person classifier is of 0.128. The disparity between the ceiling benchmarks for the Regular and the Flat datasets is more or less the same as the floor benchmarks, with the only difference for the Person classifier which differs by 0.085. The two ceiling benchmarks are shown in fig. 4.10. The interpretation of these results is that, given the technique and the data used, the classifiers for most of the second-tier features can be improved, but the maximum improvement (ceiling) is somehow *capped* at a point where it is not possible for a machine to automatically learn beyond it, at least with the features used here. This indicated that the improvement of the classifiers through flattening the data was minimal, mainly limited to the Person feature, and to a much lesser extent the Gender feature.

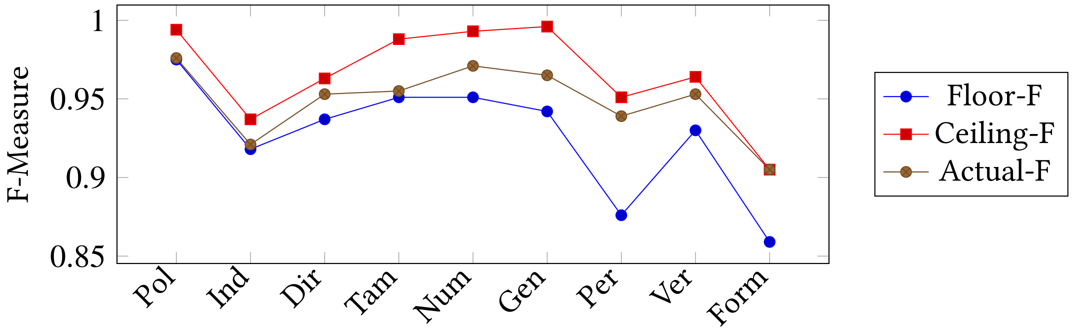


Fig. 4.8 The resulting Flat (F) cascade sequence, comparing the performance of each classifier to its Floor and Ceiling benchmarks

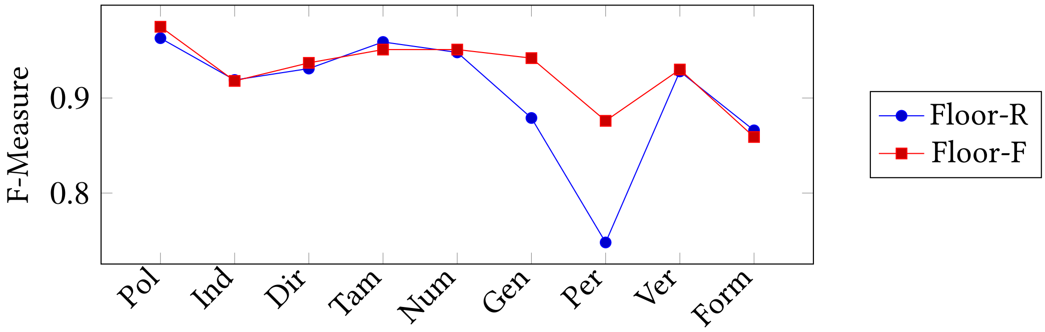


Fig. 4.9 Comparison of the Floor benchmarks for the Regular (R) and Flat (F) datasets

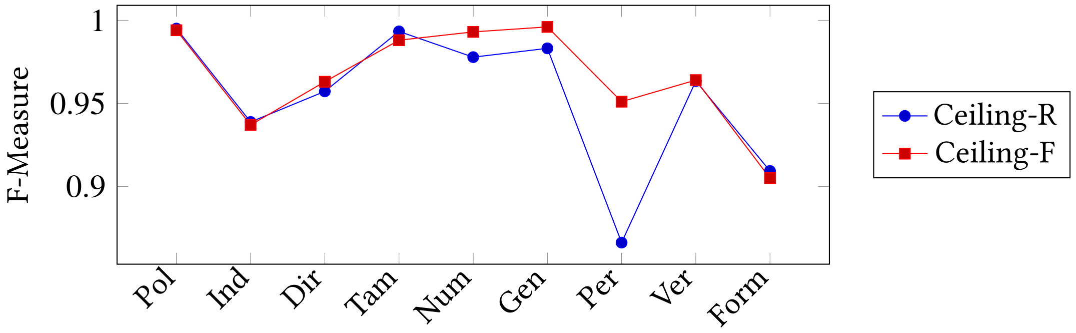


Fig. 4.10 Comparison of the Ceiling benchmarks for the Regular (R) and Flat (F) datasets

Figure 4.11 compares the classifiers learnt for the regular and the flat datasets. All classifiers have very similar performance, apart from Gender and Person classifiers. These classifiers improve by 0.073 and 0.078 respectively after the data is flattened.

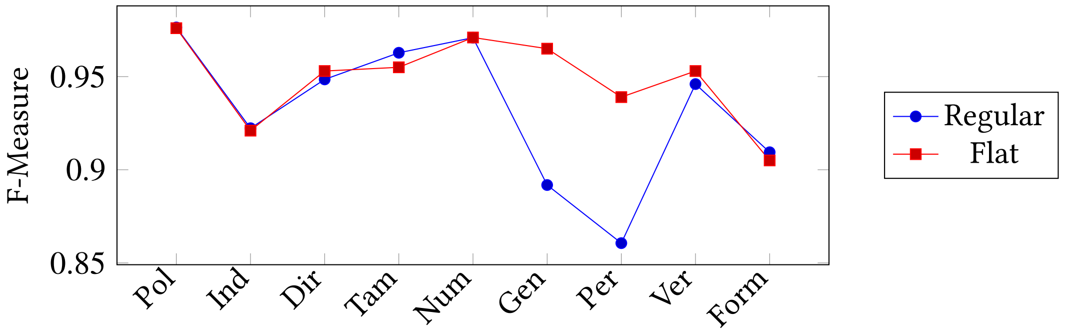


Fig. 4.11 Comparing the classifiers learnt for the regular and flat datasets.

A similar analysis was carried out with feature-pairs being first merged before the data was flattened to remove any ambiguity and co-dependency of features. Figure 4.12 shows all the different classifiers learnt from the various datasets (NumPer, PerGen, NumGen and Subject) as well as the regular and flattened classifiers shown in previous figures. To enable better comparison throughout the different datasets and models, the merged features have been separated in the presentation of the results. However, note that when features are merged, the value given on the individual features is that obtained by the single merged feature (hence, they have the same values). So for instance the subject plot in fig. 4.12 has the same value for Person, Gender and Number. Table 4.10 shows the actual F-Measure values for each of the classifiers, indicating in bold those features which are merged before being flattened.

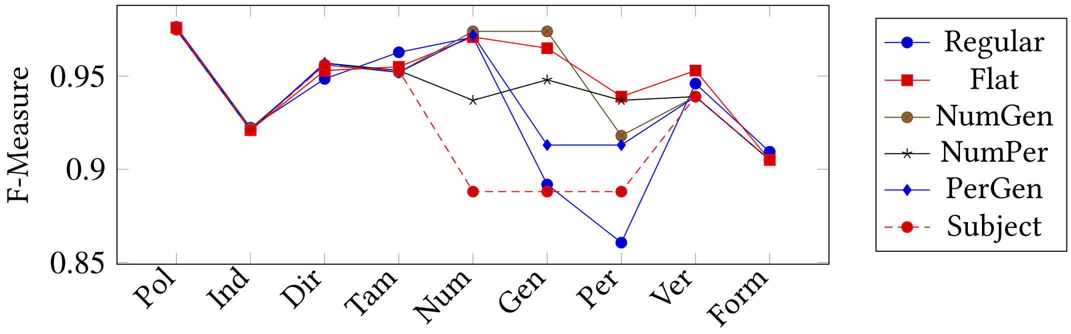


Fig. 4.12 Comparison of the classifiers learnt from the different training datasets.

Table 4.10 F-Measure of the classifiers learnt on the basis of the training datasets; numbers in bold indicate merged features.

Model	Regular	Flat	NumGen	NumPer	PerGen	Sub
Pol	0.976	0.976	0.975	0.975	0.975	0.975
Ind	0.922	0.921	0.922	0.921	0.921	0.921
Dir	0.949	0.953	0.956	0.957	0.957	0.956
Tam	0.963	0.955	0.952	0.953	0.952	0.953
Num	0.971	0.971	**0.974**	**0.937**	0.972	**0.888**
Gen	0.892	0.965	**0.974**	0.948	**0.913**	**0.888**
Per	0.861	0.939	0.918	**0.937**	**0.913**	**0.888**
Ver	0.946	0.953	0.939	0.939	0.939	0.939
Form	0.909	0.905	0.905	0.905	0.905	0.905
Average	0.932	0.949	0.946	0.941	0.939	0.924

The performance of most classifiers remained the same as before when comparing the regular and flattened classifiers. The classifiers that are of particular interest here are those for the three features which have been merged in various configurations before being flattened — Gender, Number and Person. The classifier for the Number feature performed best when it is merged with Gender, achieving a slightly better F-Measure then when it is not merged, the difference being of 0.003. It performed worse when merged with Person, as seen in the NumPer and Subject classifiers, with a disparity of 0.037 and 0.086 respectively. On the other hand, Gender and Person both performed better when merged. The Gender classifier performed best when merged with Number, but the performance was very close to when it was simply flattened without merging with other features. The lowest performance for the Gender classifier was with the Regular and Subject datasets. Similarly the Person classifier performed best when the data was simply flattened, or when it was merged with the feature Number.

The difference in the performance of the classifiers between the flattened dataset and the different merged datasets was generally minimal and too close to argue that merging any of the features provides an added advantage to the classification system.

4.3.3 Classifying unseen data

All the analyses presented so far were based on the evaluation of the classifiers using the training data itself through a ten-fold cross validation system. The following analyses focus on testing the classifiers on completely unseen data. This dataset consisted of 20,000 wordforms randomly selected from the Ġabra database which were not present in the training dataset. It was processed in exactly the same way as the training dataset, creating several datasets for the Regular, Flat, NumGen, NumPer, PerGen and Subject varieties.

The evaluation of the cascade classification system on unseen data could be approached in several ways. One was to look at the performance of each classifier in isolation, using the values of previous second-tier features from the dataset. In practice this meant that to evaluate the Direct Object classifier, the system would use the values for Polarity and Indirect Object as specified in the dataset, thus using the correct values. Another way was to evaluate the cascade executed in full as a whole sequence of classifiers, taking the results outputted by each classifier and feeding the values to the following classifier. This meant that incorrect classifications resulting from previous classifiers are propagated down the cascade. This could influence the performance of the following classifiers negatively. The latter approach mirrors a more realistic scenario as to how such a type of classification system would be used in practice. Thus the following evaluation looks at the unseen data applied to the cascade as a whole, with the resulting *error propagation* down the cascade. Specifically, if the feature Polarity of an instance was incorrectly classified, the incorrect information remained so throughout classification cascade. Due to the implementation and setup of evaluation, the results on the unseen data were calculated using Accuracy (percentage of correct classifications) only. This allows a more transparent view of the nature of error propagation. The figures below provide the accuracy of the learnt classifiers (labelled 'Modelled') as described in the previous section versus the performance of the cascade itself as a whole on the unseen data (labelled 'Unseen').

The cascade classifier for the Regular dataset is shown in fig. 4.13. The Modelled plot shows the accuracy achieved by each classifier when it is learnt and evaluated using 10-fold cross validation. The Unseen plot shows the evaluation of the classifiers on the unseen data with the values predicted by each classifier propagated to the following classifiers. The two plots are compared since they provide an indication of the impact that incorrect classifications have on the cascade. The largest drop in accuracy occured for the features T.A.M., Person, Verbtype and Form. The Verbtype and Form features are very particular, significant only to verbs of Arabic origin. As explained in §4.2.2 (page 100), their inclusion in the cascade was more of an experiment to analyse whether these two features could be learnt and included in the morphological classification system. In particular, Verbtype does not provide morphological information but is rather a classification of the type of radicals that a word has (strong, geminate, weak-final, etc.). The feature Form provides some morphological information related to the pattern for a word's formation. The gap in the results between the Modelled and Unseen evaluations showed that these two particular features were more difficult to learn, achieving an accuracy of 81% and 72% for Verbtype and Form respectively. However, these results are rather adequate when considering the type of basic features available to learn these features — for instance, if the radicals were encoded as part of the basic features, both Verbtype and Form would have achieved better results. However, the radicals are only relevant to the Semitic aspect of the Maltese language, and could have a negative impact on the classification of Romance origin verbs.

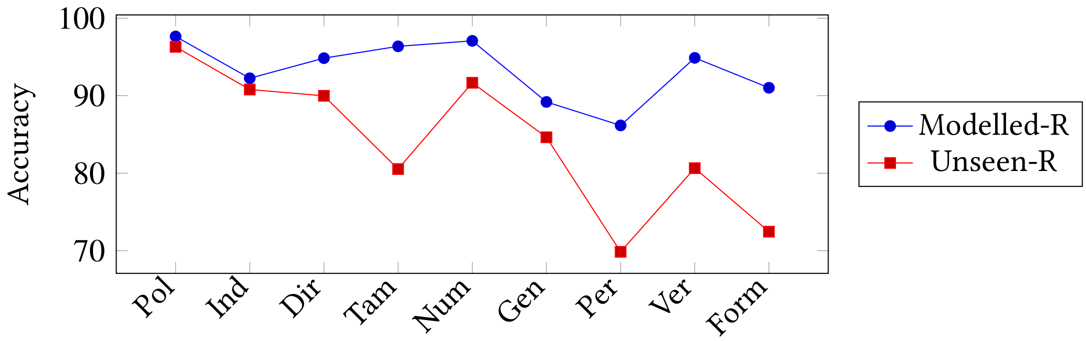


Fig. 4.13 Comparison of the Modelled classifiers against the performance of the same classifiers on Unseen data for the **Regular** dataset

The results of the unseen evaluation reconfirm the previous observations about the difficulties to learn the Person feature. One of the reasons for the drop in accuracy when compared to other classifiers was that this feature is highly dependent on the previous classifier Gender. Thus any errors in the classification of Gender would negatively impact the Person classification. The accuracy for Gender is at 85%, meaning that 15% of the instances were incorrectly classified. The drop in accuracy for the Person feature from the modelled to the unseen is of 17%, which might indicate that it is at least a partial effect of the inaccuracy in the Gender classification.

Finally, the drop in accuracy for the T.A.M. feature probably resulted from the problem described in §4.2.5.2 (page 109). The Ġabra database contained errors in the word formation of the imperative form. As a result, a large number of wordforms were discarded from the training dataset (21,140 words were removed, leaving only 604 words or less than 0.003% of the training data). However, the selection of 20,000 unseen instances for the evaluation resulted in a much higher percentage of the imperative form, with 2,584 words (0.129%). The discrepancy between the training and unseen datasets, even simply in terms of the proportion of imperative instances that were observed when building the models, was probably the primary reason for the drop in performance. The remaining second-tier features perform reasonably well, with less than 6% difference between the Modelled evaluation and the Unseen evaluation of the actual feature as predicted by the cascade.

The performance of the cascade system learnt and evaluated on the Flat data overall produced similar results to the Regular cascade in terms of drops of performance for the features T.A.M., Verbtype and Form. Figure 4.14 shows the performance of the cascade system as modelled on the Flat training dataset, and its evaluation on the unseen dataset. Here, a substantial drop in performance can be observed for the Gender and Person features, with a 17% and 35% drop respectively. This was rather surprising since the idea behind creating a Flat dataset was to remove ambiguity from instances which could be classified as say both second and third person. There can be two reasons for this large drop in performance. First, the evaluation does not take into account partially correct predictions. This means that if an instance was labelled as '2|3' for the Person feature, and the predicted value was '2', this is considered completely incorrect. The second reason might be that by removing the ambiguity completely from the training data, the classifier were less capable of generalising over unseen data, thus ended up with classifiers that overfitted the training data. This is a common problem in machine learning, and the possibility of comparing the performance of the classifiers trained on different representations of the same data allows the analysis to take the overfitting problem into consideration.

When comparing the cascade systems learnt on the Regular and Flat datasets, it is interesting to note that although in the modelled cascades the Flat outperformed the Regular cascade for the Gender and Person features, the evaluation on the unseen data gives the opposite result. Figure 4.15 shows the two cascades. This further strengthens the probability that the cascades modelled on the Flat dataset overfitted the training data.

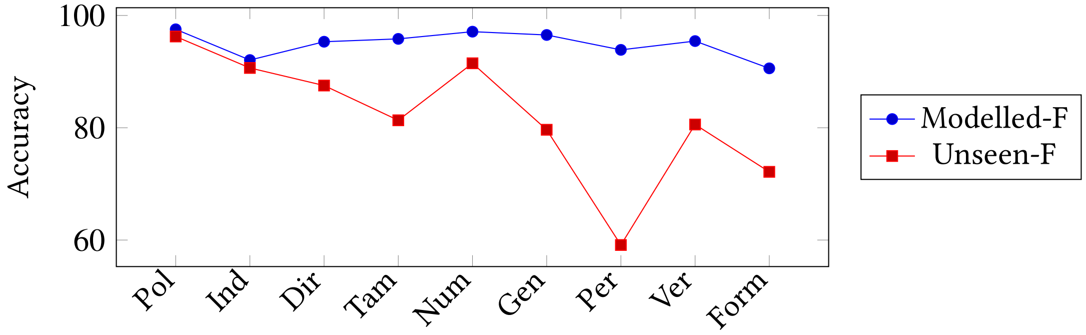


Fig. 4.14 Comparison of the Modelled classifiers against the performance of the same classifiers on Unseen data for the **Flat** dataset

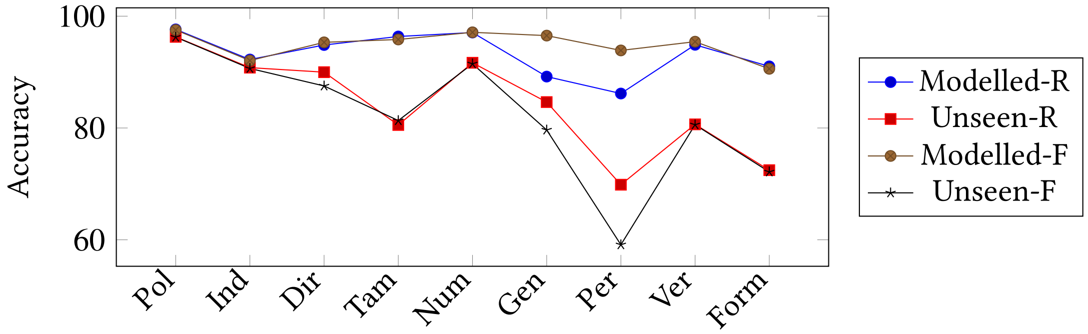


Fig. 4.15 Comparison of the Modelled classifiers against their performance on the Unseen data for the Regular (R) and Flat (F) datasets.

A similar analysis was carried out for the remaining datasets, where either two features were merged (Person-Gender, Number-Person, Number-Gender), or the three features merged into one (Subject). Figures 4.16 to 4.19 compare the Modelled classifiers with their performance on the Unseen data for the PerGen, NumPer, NumGen and Subject cascades respectively. The performance of these cascades was very similar to that of the Flat cascade, with the only difference where the features were merged. Figure 4.20 then compares all the cascades, showing that the Regular cascade classifiers performed best overall on unseen data.

The performance of the different merged features was always slightly worse than when they were left as individual feature classifiers. For instance, in comparing the PerGen classifier against the Flat classifier, the merged feature PerGen achieved a very similar accuracy to the Person classifier in the Flat cascade but a much lower accuracy for Gender, which performs much better in the Flat cascade. This indicated that merging features together

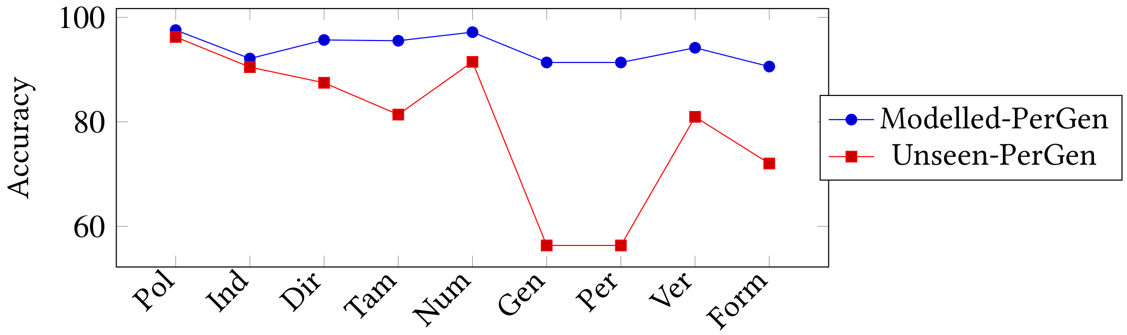


Fig. 4.16 Comparison of the Modelled classifier against the performance of the same classifiers on Unseen data for the **PersonGender** dataset

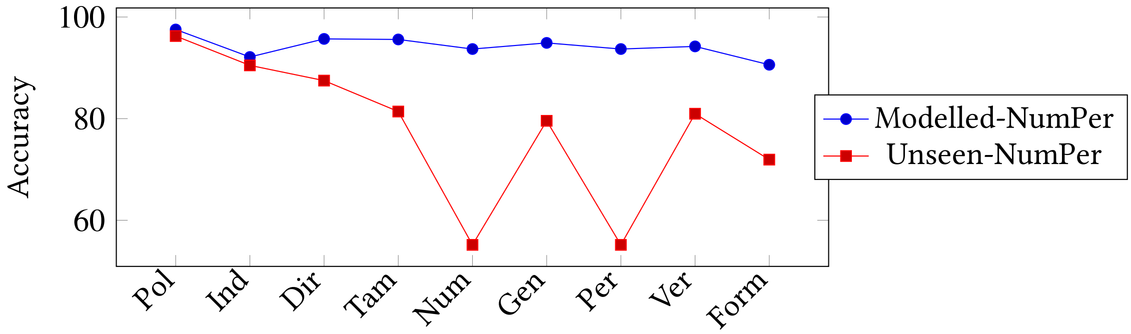


Fig. 4.17 Comparison of the Modelled classifier against the performance of the same classifiers on Unseen data for the **Number-Person** dataset

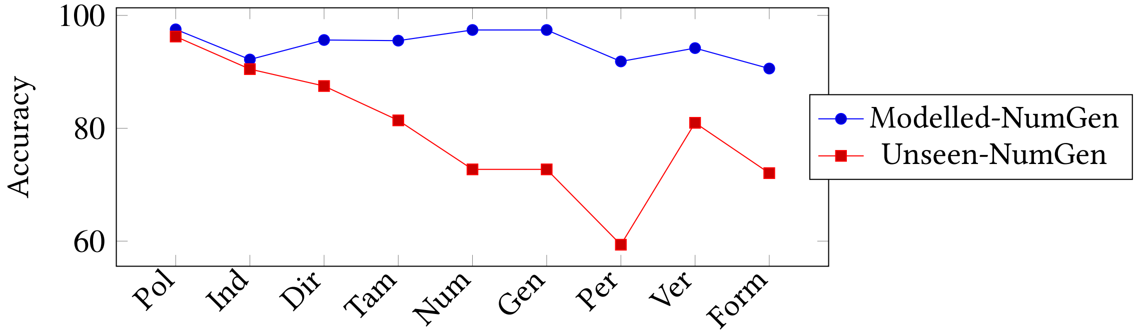


Fig. 4.18 Comparison of the Modelled classifier against the performance of the same classifiers on Unseen data for the **Number-Gender** dataset

adds complexity to the learning of classifiers and negatively impacting the prediction on unseen data.

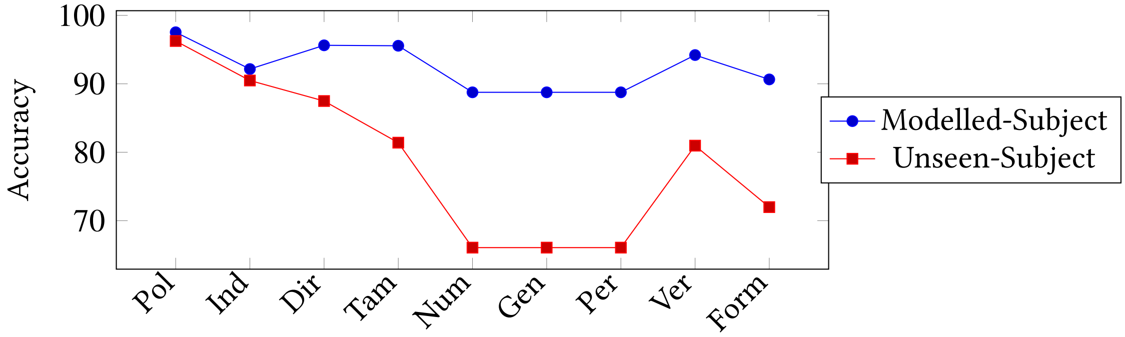


Fig. 4.19 Comparison of the Modelled classifier against the performance of the same classifiers on Unseen data for the **Subject** dataset

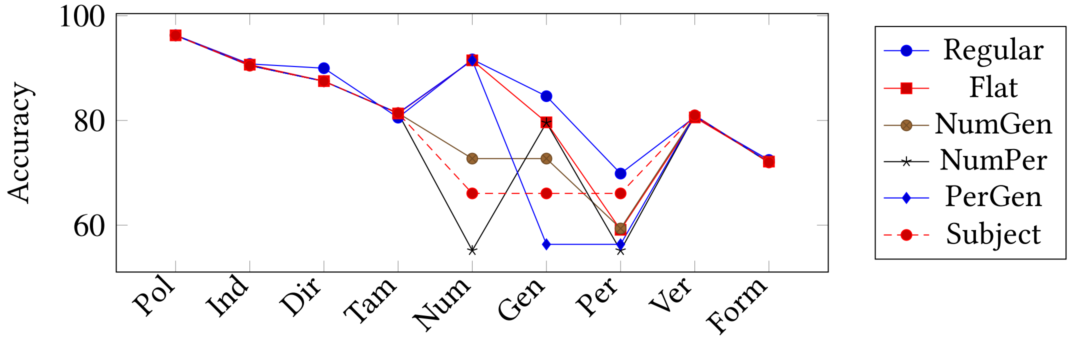


Fig. 4.20 Comparison of all the cascade classifiers on the unseen data

From the above analyses it is clear that the Regular cascade performed best in terms of predicting labels on unseen data. We identified various reasons for this result. In removing the ambiguity in the Flat dataset, the resulting classifier overfitted the training data, limiting the generalisation over unseen data and resulted in a larger number of incorrectly classify instances when compared to the Regular cascade. The evaluation was also limited since it did not consider partially correct predictions, but only those which match the full label. In the following section we analyse in more depth the performance of the classification system as a cascade, focusing solely on the Regular and the Flat cascades since the other cascades generally perform worse than these two.

4.3.4 Performance evaluation of the Cascade Classification System

In order to assess the actual performance of the Regular and Flat cascades, we compare the classification of the unseen data versus a baseline classification. One approach was to use the Majority Class algorithm, which statistically takes the majority class and simply classifies all instances according to this class. This comparison would be rather simplistic, of course, and would not truly reflect the complexity of the data. Whilst certain features had several values/classes, other features had only two, with the data being split in approximately half. For instance the feature Polarity had positive and negative as values, with the data split practically into half and half — with the Majority Class classifier, the accuracy for Polarity was around the 50% mark, with all instances classified according to the label that had slightly more instances in the training dataset.

Another approach was to classify each feature using only the basic features — this eliminates the idea of a cascade classification system, and treats all second-tier features as though they are being classified in parallel. This is equivalent to the Floor benchmark used before in §4.3.1 (page 111), where the Floor and the Ceiling benchmarks for all second-tier features were used to determine the sequence of the cascade. The same classifiers used for the Floor benchmark analysis were used to classify the unseen datasets so as to compare the performance of the Floor benchmark to the results obtained by the full cascade classification system on the same data. Figures 4.21 and 4.22 compare the results of the cascade with those achieved by the Floor Classifiers and the Majority Class classifier on the unseen data for the Regular and the Flat cascades respectively.

As expected, the Majority Class classifier performed poorly for both the Regular and Flat cascades. However the difference in performance between the actual cascade classification system and the Floor benchmark was close to nil. The idea of having a cascade classification system was that each of the second-tier features would provide information

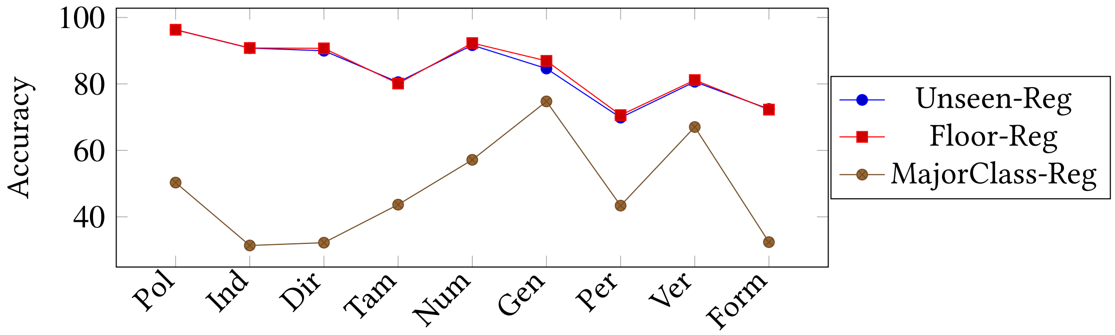


Fig. 4.21 Comparison of the Regular cascade against the Floor and Majority Class classifiers on the unseen data

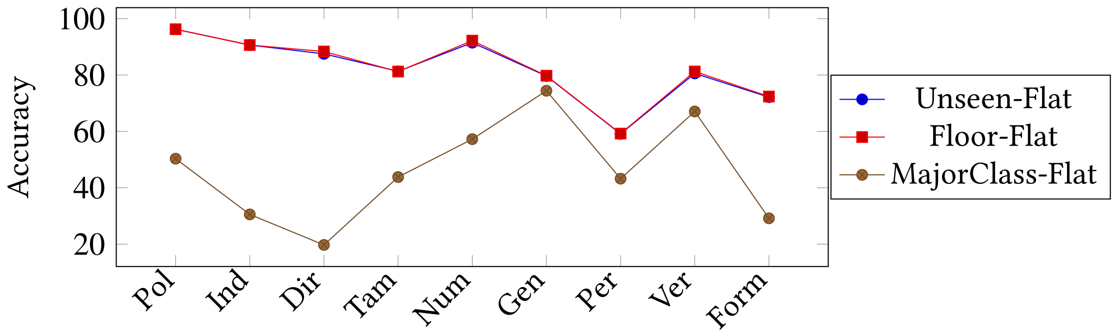


Fig. 4.22 Comparison of the Flat cascade against the Floor and Majority Class classifiers on the unseen data

to the following features in the cascade sequence. Thus hypothetically speaking, the performance of the cascade should be better than that of the Floor classifier. It is important to emphasise here that the Floor classifiers were trained using only the basic set of features, whilst the cascade classifiers were trained using also any preceding second-tier features in the cascade. Although this added information to the classifier, it might also add complexity to the learning task, making it difficult for the classifiers to perform better with the added information. A more detailed look into error propagation was required to understand further this performance. Error propagation occurred when an incorrect prediction was made by a classifier, impacting the performance of any following classifiers in the cascade.

The accuracy on unseen data presented so far looked at each classifier individually, so an instance could have a correct prediction by one classifier, and an incorrect prediction by an other. In the following analysis, we were interested in identifying the following factors. First, the actual accuracy throughout the cascade where an instance is predicted correctly by all classifiers. This means that once an instance received an incorrect prediction, it remained flagged as incorrect irrespective of the classifications received by the following classifiers. This is referred to as ‘Only Correct’. Second, the accuracy of the classifiers, if at every classifier, the incorrect instances were removed from the pool of instances, and only the correct instances were classified further. This is referred to as ‘absolute accuracy’. Figures 4.23 and 4.24 show this analysis for the regular and flat cascades respectively.

The analyses show that by the end of the Regular and the Flat cascades, only 43% and 35% of unseen instances were classified correctly throughout by the respective cascades. However, if Verbtype and Form were not considered, the accuracy goes up to 58% and 48% respectively — a rather positive result especially for the Regular cascade. The largest drop in performance for this cascade was for in the T.A.M. classifier, and this was probably due to the anomaly in the data with respect to the imperative. This indicates that the results might improve if the classifiers were retrained once the data is corrected.

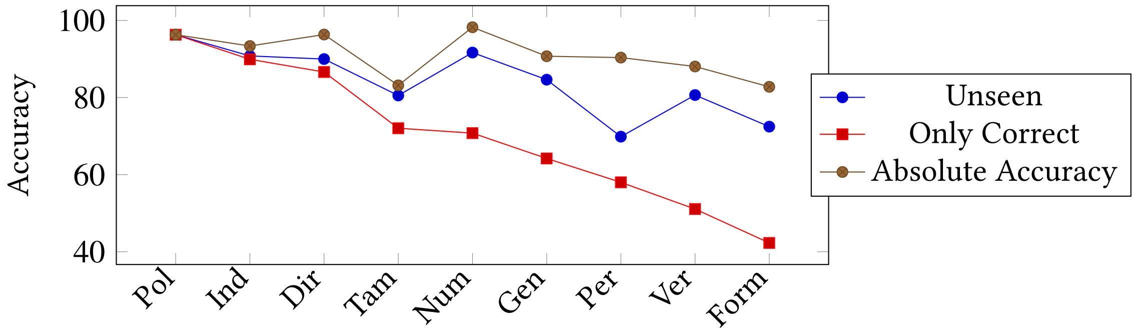


Fig. 4.23 Comparison of the Regular cascade and the percentage of classifications that were correctly predicted throughout the cascade (only correct) and the accuracy considering only correct predictions from the previous classifier (absolute accuracy).

Another factor to consider when analysing the performance of the cascade against the Floor benchmark, which used only the basic set of features, is the actual quality of the basic features which were extracted automatically. Primarily, the identification of the stem and affixes in general had a rather high level of correctness because the radical consonants were used in this process. This, in turn, meant that the identification of the affixes was also correct on the whole. The good quality of the basic set of features might have resulted in them being sufficient for the classification of all the second-tier features. The unseen data that the classification system was tested on also has the same good quality basic features, and a high level of homogeneity since the training and test instances were taken from the same database and preprocessed in the same way. This might not be the case when using

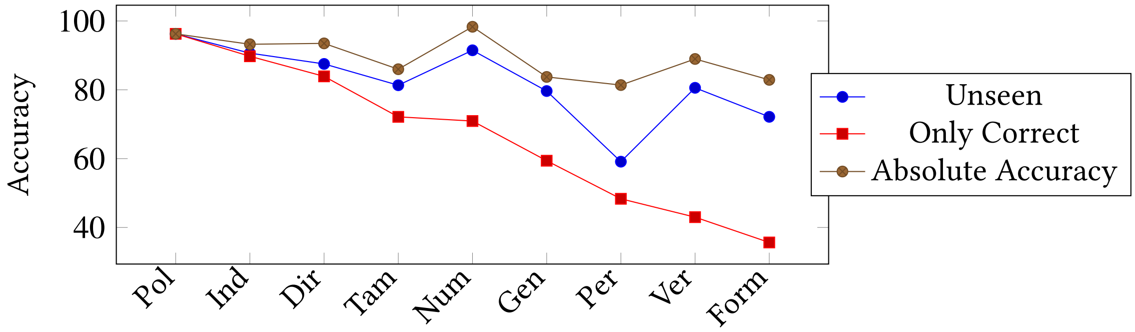


Fig. 4.24 Comparison of the Flat cascade and the percentage of classifications that were correctly predicted throughout the cascade (only correct) and the accuracy considering only correct predictions from the previous classifier (absolute accuracy).

a more realistic data, such as text from the MLRS corpus. A small scale evaluation on text taken from the corpus is described further on in §5.3.

4.3.5 Comparison of different machine learning techniques

The experiments and analyses described above used Decision Trees since this was generally the best performing technique in the experiments carried out initially. The purpose was primarily to explore the best way to carry out labelling, so the specific machine learning technique was a secondary consideration. The following analysis compares different techniques, previously described in §4.2.4 (page 103) using the same cascade sequence and datasets for the Regular and the Flat cascades. These are shown in figs. 4.25 and 4.26 respectively.

Decision Trees remained one of the best performing techniques, slightly surpassed by Random Forests — this is a reasonable result since Random Forests use decision trees as its basis. Random Forests are also known to have the capacity to find a well-fitting model to many datasets, though with the disadvantage that the properties of the optimal model are not completely transparent. This is of course because they rely on multiple Decision Trees. The Majority class classifier, which was already used to compare the results against a baseline, had the lowest performance for all the classifiers. The remaining two techniques, SVMs and Naïve Bayes, perform quire closely to Decision Trees and Random Forests, but on average the performance was slightly lower. One of the main advantages of using Decision Trees as the preferred technique was that the produced tree can be easily transformed into a set of rules reflecting the paths being executed to arrive to a classification. On the other hand, a technique like Naïve Bayes classifier not only provides a prediction, but also associates a probability to all the possible classes. This is particularly useful for wordforms which belong to more than one class, e.g. second and third person. Whereas a decision tree would predict only one class, Naïve Bayes would associate a probability to each class and therefore a final system could take into consideration all classes with a probability above a certain threshold.

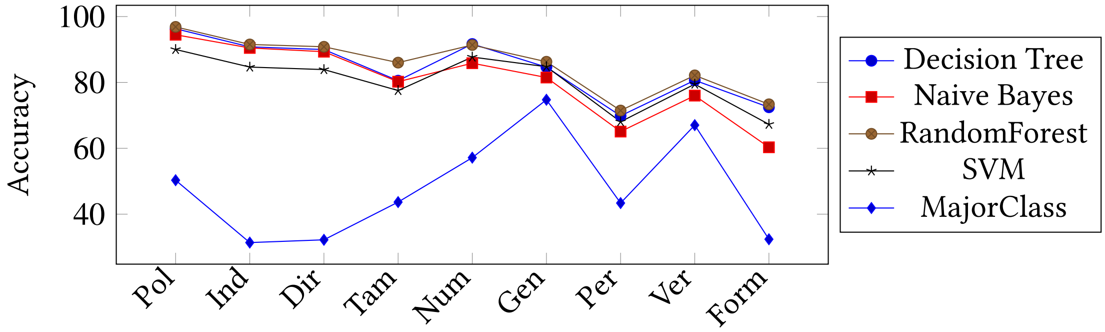


Fig. 4.25 Comparison of Decision Trees, Naïve Bayes, Random Forests, SVMs and Majority Class on the **Regular** cascade.

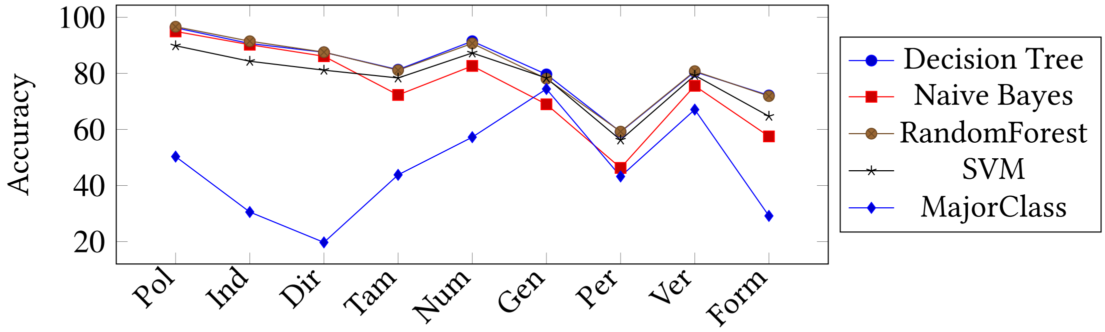


Fig. 4.26 Comparison of Decision Trees, Naïve Bayes, Random Forests, SVMs and Majority Class on the **Flat** dataset.

4.4 Classifying nouns and adjectives

The experiments described so far focused on the classification of verbs and resulted in a cascade system that specifically targets the features of verbal morphology in Maltese. This principle can be extended to other categories, in particular nouns and adjectives. In this section we describe preliminary experiments carried out on the noun and adjective categories, followed by §4.5 which provides an overview and analyses of the results.

The main limitation of the Ġabra database was that it mainly consisted of inflectional verbal wordforms. Therefore, it was necessary to exploit other possible resources to obtain labelled data for other categories. The two other main part-of-speech categories that have non-trivial morphological properties, and to which the principle of a cascade classification system could be applied to, are the noun and adjective categories. The experiments maintained the same principle as before — i.e., seeing morphological labelling as a classification problem, whereby a category has a number of features which can be learnt and modelled, and each classifier provides further information to subsequent classifiers. The same strategy was applied: (i) defining the second-tier features that are 'classifiable'; (ii) finding the optimal sequence of the classifiers; (iii) learning the classifiers on training data; (iv) evaluating the classifiers on unseen data. The experiments carried out in the noun and adjective categories did not require the same level of detail as was done initially for the verbs. For instance, the experiments done to determine the representation and effectiveness of composite suffixes was not repeated since ideally the basic set of features should remain the same for all types of categories. However, a similar analysis was carried out to determine whether the cascade classifiers provided an added benefit to the morphological labelling process.

The dataset used for these two categories was specifically extracted from a digital scan of the Aquilina dictionary (Aquilina, 1987–1990). This meant that, unlike Ġabra, which provided a very well structured and labelled dataset, the data for the nouns and adjectives had to be extracted and preprocessed to generate the necessary wordforms and their respective labels. However, some of the dictionary entries were not adequately extracted or processed, so the data contained noise to some degree. The data was also restricted to the morphological information that is normally found in a dictionary, with some inflectional features specified but with a main focus on derivation. So the data contained several gaps and certain morphological properties were not adequately covered or not present at all. Notwithstanding the difficulties in the extraction process, the dictionary data provided the best resource in terms of having a collection of nouns and adjectives and their respective grammatical information. The exercise was an essential part to move towards a complete morphological analyser, which could always be improved as more data and resources become available for Maltese.

4.4.1 Dictionary data extraction

The first task was to identify and exploit possible data sources that could easily be turned into labelled data. Farrugia (2010) collected a list of nouns and their gender, mainly compiled from the Maltese-English dictionary (Aquilina, 1987–1990). Another compilation from the same dictionary was carried out in Ellul (2015), this time documenting verbal nouns. It is worth noting that these works were primarily focused on specific linguistic features of Maltese, and that the collections were created manually from a hard copy of the dictionary. However, both collections were very narrow in their focus, and the arrangement and cataloguing of the data was particular to the work in question. Another possible resource was the Maltese-English dictionary itself. In parallel with this research, various efforts to digitalise the dictionary were made, and an ongoing project (referred to as the dictionary project) aims at creating an online dictionary using Aquilina as a starting point. Within the dictionary project, a program to automatically extract some aspects of the dictionary was created. This extracted the head word, some basic grammatical information about the head word and etymological information of various dictionary entries. Although the output could provide an initial starting point, all of the derivational information within a dictionary entry was not being extracted, so the data within the dictionary was being under utilised. Figure 4.27 shows an example of a dictionary entry, showing what was being extracted in italics, and what was further required for this work in bold.

In analysing the data sources available, the following were taken into consideration: (i) the amount of work required to transform the data into a useable dataset for the experiments, and (ii) the coverage of the data. The collection from Ellul did require some effort to transform into a dataset, and much less was required for the collection from Farrugia. However both were very limited in the coverage of information — the latter was restricted to gender analysis of nouns, and the former to verbal nouns. The dictionary extraction required more work, but its coverage was much broader and better represented the type of data required for a morphological system. Extending the dictionary extractor would not only result in a dataset for this research, but also feed back into the dictionary project, thus also contributing beyond this research. The dictionary extraction was deemed to be the best overall approach to obtain the required dataset.

> *ABBUŻ [Mamo], n.m. (pl. ~i)* 1. Abuse, unjust or corrupt practice (naqtgħu dawn l- ~ i, let's put an end to these abuses). 2. (leg.) Misuse, abuse; - ta' l-awtorita, abuse of power (use of undue authority); - tal-fiduċja, abuse of confidence, breach of trust; - tal-patria potestas, abuse of paternal rights; - tal-poter, abuse/misuse of power. **-IV, a.m. (f. - iva, pl. - ivi)** Abusive. **~ IVAMENT, adv.** Abusively. [id.] **~ |A, v.i. (imperf. + a, pp. ~ at, vn. ~ ar)** 1. To abuse, to make bad use of (minn) s.th.; to take unfair advantage of (bi) S.o., to presume upon, assume that one can take advantage of s.o. in asking favours (qed t~a mit-tjubija ta' missierek, you are abusing your father's generosity; ma rridx min j-a bija, I do not want anyone to take advantage of me; se n-a bik, imma nixtieqek tgħinni, I am presuming on your good nature to ask for your assistance). 2. To dare, have the necessary courage to (ma n~ax ngħidlu, I dare not tell him). *[Sic./It. abbus-ujo; -ivu; -ivamente; Sic. -ari]*

Fig. 4.27 A dictionary entry — the text in italic shows what was extracted by the initial program and the text in bold shows further extraction required for this research.

The extractor is built using regular expressions and parsed the dictionary text files which were taken from scanned images and converted to text files through OCR software. Each line in the text file represented a dictionary entry, and these were treated as strings, matching parts of the entries according to the constraints of the regular expressions. Extending this program meant designing further expressions and including a reasoning component to the program to transform the wordforms. In fig. 4.27 above, note how certain information, although present within the entry, must be processed further in order to obtain the wordforms. For instance in order to extract and form the word *abbużi*, the program must process the information available in the text *(pl. ~i)*. Once the regular expressions captured this information, the reasoning component processed it and transformed it into the necessary wordform, associating with it the relevant grammatical information. Table 4.11 shows the list of words extracted and transformed from the sample entry above.

The extended extractor was able to capture most of the derived words within an entry. However, it was not a foolproof system and there were some errors in the extraction process or difficulties within the reasoning component. Some morphological aspects were not captured at all. Figure 4.28 shows that the noun *kamra* 'room' can take a pronoun suffix

Table 4.11 A sample list of words and their features extracted through the program. The words in bold were extracted through the work carried out for this thesis.

Stem	Word	Suffix	Cat	Gen	Num	Type
abbuż	*abbuż*		noun	masc	sg	
abbuż	**abbużi**	i	noun		pl	
abbużiv	**abbużiv**		adj	masc	sg	
abbużiv	**abbużiva**	iva	adj	femm	sg	
abbużiv	**abbużivi**	ivi	adj		pl	
abbuż	**abbużivament**	ivament	adv			
abbuż	**abbuża**	a	verb			intrans.
abbuż	**abbużat**	at	verb			pastpart.
abbuż	**abbużar**	ar	noun			verbal

and becomes *kamarti* ‘my room’, *kamartek* ‘your room’ — however due to the formatting of the entry, it was difficult to extract this information automatically. Another aspect relates to OCR errors, where for instance the tilde sign (~) is converted into a dash sign, so it was not always clear when the headword should be inserted or not. Some of these problems were actually present in the dictionary itself as typesetting errors, and where possible the regular expressions took these errors into consideration. It is clear that although this resource was exploited as best as possible, more work would be required to extract further information from the dictionary and to reduce the errors from the extraction and reasoning process.

KAMRA [Sol 284r], n.f. (pl. [Vass] kmamar) 1. Room, chamber; ~ tal-banju, bathroom; il- ~ l-baxxa, (joc.) water closet, loo; ~ tal-bejt, attic, box-room; tal-ġenb/laterali, side-room; ~ ta’ l-imbarazz, lumber room; ~ tal-ħasil, washing-room; ~ tal-logħob, gaming-room; ~ tan-nar, the place, gen. a room, away from the inhabited area where fireworks are made; ~ tal-pranzu/ta’ l-ikel, dining room; ~ tas-sodda, bedroom; il- ~ tiegħi, tiegħek, eċċ./**kamarti, kamartek**, eċċ., my, your, etc. private room.

Fig. 4.28 An example of morphological information that is not captured by the extractor, shown in bold.

The extended extractor produced a wordlist of 63,540 words together with their main category and other morphological features, similar to the sample list shown in table 4.11 above. Table 4.12 shows the number of words extracted for the different part-of-speech categories. The adverb category does not have any particular morphological features such as gender and number and the quantity extracted was rather low when compared to other categories, representing around 1% of all words extracted. For this reason, adverbs were not included in this analysis and in further experiments. The noun category was the most represented one, with 63% of the words extracted in this category, followed by adjectives and verbs. A number of entries did not have a category clearly associated to them or it could not be correctly extracted, thus these words are ignored.

Table 4.12 Number of words extracted from the dictionary per grammatical category

Category	**No. of Words**	**Percentage**
Nouns	39,678	63%
Adverbs	636	1%
Adjectives	10,731	17%
Verbs	8,465	13%
Category not captured	4,030	6%
Total	63,450	

4.4.2 Examining morphological properties

In order to know which morphological properties could be used as second-tier features for the extracted words, an analysis was carried out for each part-of-speech category. Table 4.13 provides an overview of this analysis. For each part-of-speech, the table lists the morphological property together with a list of values attested for that property, and the number of words having those values.

One of the primary observations of the extraction and reasoning process was the difficulty in finding consistency throughout the dictionary entries, which might have resulted in some processing errors. For instance, for the adjective category there is the value ‘inv’ under both gender and other. The reason for this is that the extraction process would have captured the string ‘inv’ at a different/unexpected location within the text. So for 13 entries it was not automatically clear whether the ‘inv’ string related to gender or whether it was a different marker in the text. Similarly the value ‘c’ under other, where it was not always clear whether this was referring to the collective or counted. Again, under nouns, there is countable under number, and pl.counted under other. The two sets are distinct, and were left distinct because of how they were actually marked in the dictionary. These are all considered as minor issues, since generally the number of words falling in these peculiar values were small and mostly insignificant.

The most represented properties and values for the noun category were gender (feminine, masculine and blank — only the singular forms are marked for gender, hence a large number of nouns marked as blank), number (mainly singular and plural) and verbal nouns. For adjectives the main properties were gender (feminine, masculine, invariable and blank), number (singular and plural) and other (mainly 'agent'). For verbs the main information available was related to the type of verb, whether it was transitive, intransitive or past participle. The number of imperfective words was rather small because the transformational rules specified in the dictionary were not always clear and consistent, so this information was ignored.

4.4.3 Extracting basic features

Through the extraction and reasoning components, the majority of words had affixes directly associated with them since the dictionary entries specified how a word is transformed to obtain another word. However, a large number of words did not have such information associated with them (25,809 (40%) words). There were also a number of cases where the identified stem/affix was not correct. This was particularly common for entries where the dictionary would mark the stem boundary as required by the stemming process for the derivational word transforms, but this would result in an error for a particular inflectional transform. An example of such a case is shown in fig. 4.29. The vertical bar indicates where the word should be segmented for the entry *aċċeler|atur*, resulting in the stem *aċċeler-*. However, the transformation for the plural — *(pl. ~i)* — would use the stem and incorrectly produce **aċċeleri* instead of using the whole headword and producing *aċċeleraturi.* For most such entries it was not possible to automatically correct the extraction of this information. Ideally such words would be cross-checked with another resource such as the MLRS corpus to check if the word is attested. Further improvements to the reasoning component could also attempt to deal with such errors by creating a set of processing rules that could post-process such error-prone word transforms in an automatic manner. However, these errors were left as part of the dataset used in the experiments described below.

The affixes were identified through the stem and the transformational processes given in a dictionary entry. However for this dataset, suffixes were not split into further compos-

Table 4.13 Analysis of the different features and their respective values for the words extracted from the dictionary

Feature	**Value**	**Nouns**	**Adj**	**Verbs**
	Totals	39,678	10,731	8,465
Gender	femm	9,414	2,943	3
	masc	10,146	3,494	8
	inv	63	1,074	
	neuter		38	
	blank	20,055	3,182	8,454
Number	sg	19,794	7,623	8
	pl	13,980	3,068	8
	singulative	1,501		
	collective	239	1	
	countable	351		
	blank	3,813	39	8,449
Diminutive	true	622	70	
	false	39,056	10,661	8,465
Other	pl. counted	305		
	“c”	20	23	
	inv	63	13	
	mimat	164		
	exclamation	1		
	agent		1,980	
	compar		38	
	blank	39,125	8,677	8,465
Verbal	true	5,388		
	false	34,290	10,731	8,465
Type	imperf.			25
	trans.			2,716
	intrans.			2,245
	past partic.			3,462
	blank	39,678	10,731	17

ite suffixes. The rationale behind this decision was that, unlike the Ġabra dataset where all the inflective forms were given for a word, the dictionary mainly lists derivationally related entries. Leaving the suffix whole would still provide sufficient information to distinguish the morphological properties of the words in the dataset.

> AĊĊELER|ATUR [ESI], n.m. (pl. ∼i) (mech.) Accelerator. ∼AZZJONI, n.f. (pl. ∼jiet) Acceleration. ∼ |A, v.t. & i. (imperf. +a, pp. ∼at, vn. ∼ar) To accelerate. [It. acceler-atore; -azione; -are]

Fig. 4.29 Dictionary entry for **aċċeleratur**

The extractor also identified the radicals when an entry had the root information available. In Ġabra, the radicals were used both in the stemming of words, and in the geminate identification which was used as part of the basic set of features. Within the dictionary data, the only category of interest which had the first consonant geminated was the verbal noun. Gemination was also found under several verb entries in the dictionary, however this was not used as part of the dataset.

The Consonant-Vowel patterns (CV patterns) were extracted for both the whole words and the stems identified through the extraction process. The CV pattern was obtained by first extracting a word's phonetic transcription using a word-to-phoneme transcription method for Maltese developed by Borg et al. (2011). The phonetic transcription was then changed into a CV pattern reflecting the occurrence of consonants and vowels in a string. This was the same process as described previously in §4.2.2 (page 98).

The datasets were divided into two: 90% of the words were randomly selected and used as training data, with the remaining 10% left as unseen data to be used for the final evaluation of the classification cascade system. Again, the training dataset was used during the development cycle to evaluate the classifiers and determine the sequence of the cascades, using a 10-fold cross validation system. The unseen datasets was only used in the final evaluation, and the results using this dataset were labelled as 'Unseen'. The adjectives dataset consisted of 9,658 instances in the training dataset and 1,073 instances in the unseen dataset. The nouns dataset consisted of 35,710 instances in the training dataset and 3,968 instances in the unseen dataset.

4.5 Experiments and results

Similar to the experiments carried out for the verb category, the first set of experiments focused on determining the optimal cascade sequence for the noun and adjective categories separately. Again, decision trees and information gain analyses were used for the initial experiments, using only the training datasets. Once the cascades were defined, they were tested on the unseen data to analyse the actual performance of the cascade and to see whether error propagation had an impact on these two cascades.

One of the main differences between these experiments and those carried out for the verbs was that Ġabra had information related to several morphological properties within its labels, and this resulted in 9 second-tier features in the verb cascade. With the information gathered from the dictionary, both the nouns and the adjectives had 3 second-tier features. For the noun category, these were Gender, Number and Verbal nouns (verbal for short). For adjectives these were Gender, Number and Other. The feature Other for the adjectives was a categorical feature with miscellaneous values, most of them of a morpho-semantic nature, with the main one, ‘agent’ indicating an agentive noun. Although there were other values in the extracted datasets, their frequencies were too small to be statistically relevant and classifiable. For example, this was the case for the diminutive with only 622 instances out of 39,678 nouns, and 70 instances out of 10,731 adjectives — in both datasets this was less than 1%. It was clear that a Majority Class classifier would classify all instances as ‘not diminutive’, and would achieve a very high accuracy rate but such a result would be very misleading.

4.5.1 The optimal cascade sequence

The initial experiments aimed at finding the best sequence for the second-tier features to be learned, maximising the benefit to subsequent classifiers. Again, we used the Floor and Ceiling benchmarks to determine which second-tier feature should be placed next in the cascade. The Floor classifiers used only the basic set of features as their input, whilst the Ceiling classifiers used both the basic set of features as well as all the other second-tier features. Tables 4.14 and 4.15 provide an overview of the process in determining the sequence of the adjectives and noun cascades respectively. At each stage, the feature with the smallest difference to the ceiling performance was selected to be the next classifier in the cascade. The remaining features are then modelled and tested with the basic features and any additional second-tier features that were in the cascade. This process was repeated according to how many second-tier features there were. In both the nouns and adjectives cascades there were 3 second-tier features so this process was quite brief.

The resulting sequence of the adjectives cascade was Other, Number and Gender. In the case of the noun category, the sequence was Number, Verbal and Gender. The analysis of the nouns cascade showed that both Verbal and Gender are highly dependent on the feature Number, and in both cases the F-Measure improved considerably once the cascade included the feature Number. In the case of the adjectives cascade, the second-tier features seemed to be rather independent of each other, as is evidenced from the Floor benchmark for each feature. All features here were already very close to the Ceiling benchmark, and the improvement obtained through the cascade was rather minimal.

Table 4.14 Adjectives cascade sequence selection

1st Classifier	Ceiling	Basic Features	Difference
Number	0.989	0.945	-0.044
Gender	0.861	0.812	-0.049
Other	0.856	0.851	**-0.005**
2nd Classifier	Ceiling	Basic+Other	
Number	0.989	0.958	**-0.031**
Gender	0.861	0.826	-0.035
Other	0.856		
3rd Classifier	Ceiling	Basic+OthNum	
Number	0.989		
Gender	0.861	**0.861**	**0**
Other	0.856		

Table 4.15 Nouns cascade sequence selection

1^{st} Classifier	Ceiling	Basic Features	Difference
Number	0.917	0.821	**-0.096**
Gender	0.946	0.814	-0.132
Verbal	0.998	0.802	-0.196
2^{nd} Classifier	Ceiling	Basic+Num	
Number	0.917		
Gender	0.946	0.935	-0.011
Verbal	0.998	0.998	**0**
3^{rd} Classifier	Ceiling	Basic+NumVer	
Number	0.917		
Gender	0.946	**0.946**	**0**
Verbal	0.998		

This can also be seen in figs. 4.30 and 4.31 for the adjective and noun cascades respectively. The Floor and Ceiling benchmarks provide a band in which the performance of the actual classifiers should be in. The performance presented in these figures used a 10-fold cross validation on the training data itself.

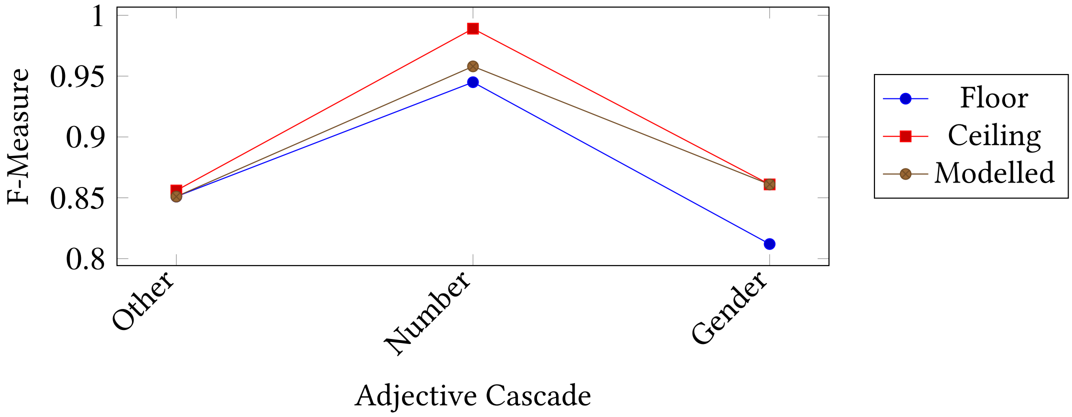


Fig. 4.30 The resulting Modelled adjective cascade sequence, compared the performance of each classifier to its Floor and Ceiling benchmarks

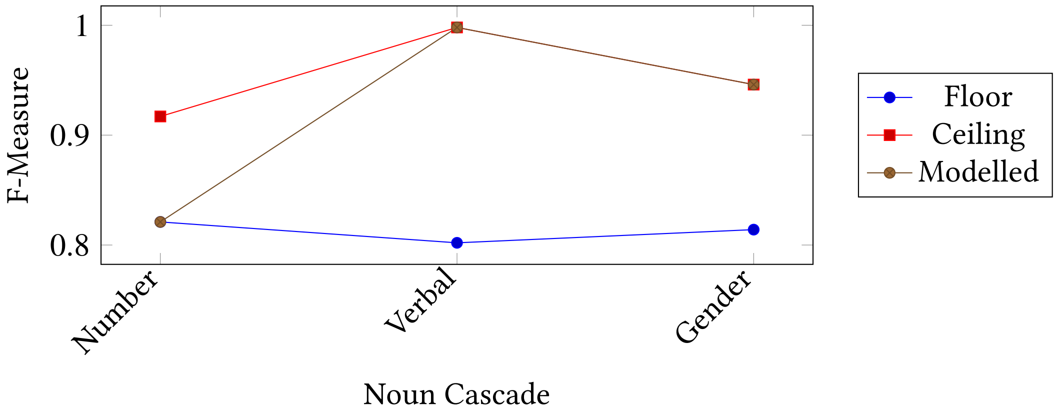


Fig. 4.31 The resulting Modelled noun cascade sequence, compared the performance of each classifier to its Floor and Ceiling benchmarks

4.5.2 Classifying unseen data

The resulting adjective and noun cascades were tested on unseen data, which includes the error propagation along the cascade as incorrectly classified instances are passed on to subsequent classifiers. Figures 4.32 and 4.33 show the performance of the Adjective and Noun cascades on unseen data, in comparison with the performance of the individual classifiers when modelled with the training data.

The overall performance for both cascades was promising. The interesting result was that in both cascades the feature Number performed practically as well as the performance of the actual model. For the noun cascade, this was particularly important since it is the first feature to be classified and the other two features, Verbal and Gender, depend on the Number feature. However, the impact of error propagation is evident on the nouns cascade. The accuracy for the feature Number in the nouns cascade is at 83%, meaning that nearly 17% of the instances were incorrectly classified. The drop in performance of the subsequent features increments, with a 10% difference in the Verbal feature and a 15% drop in the Gender feature. The adjective cascade overall performed better, with the highest drop in accuracy of 7% for the gender feature. This was close to the drop in performance registered in the verbs cascade, where there was a 5% drop in accuracy between the modelled and the unseen results for the Gender feature. All the features in the adjectives cascade also had less dependency on other second-tier features, and therefore the drop in performance was considered as normal when the cascade is applied to unseen data.

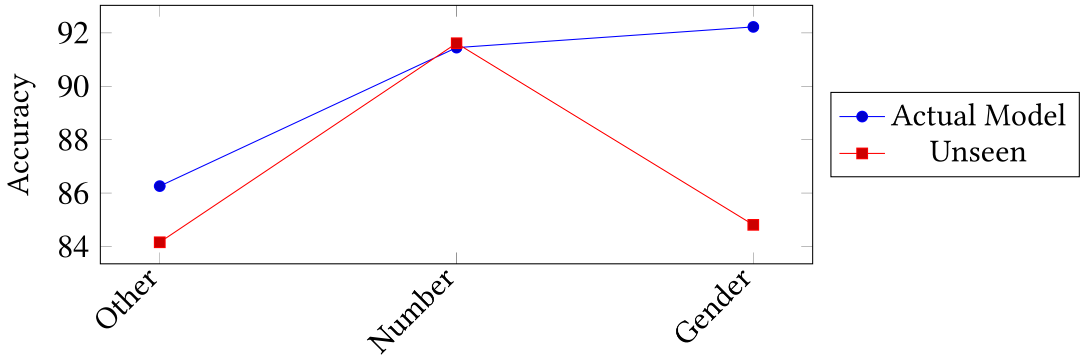


Fig. 4.32 Comparison of the performance of the **Adjectives** cascade between the modelled classifiers and the execution of the cascade on unseen data.

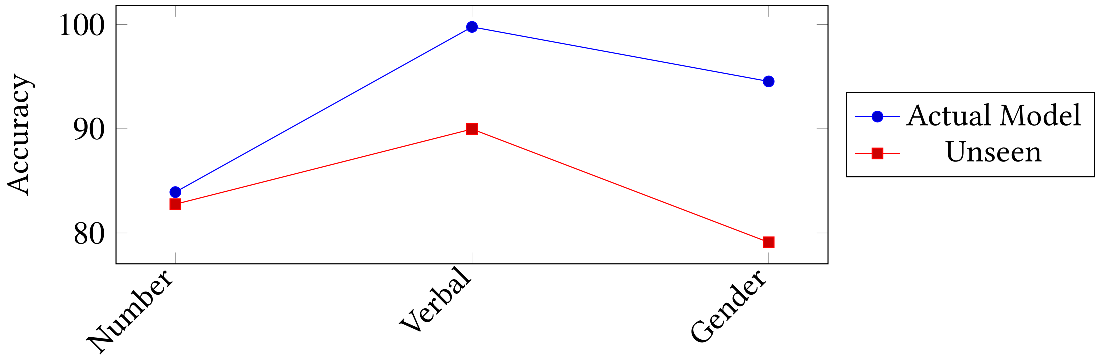


Fig. 4.33 Comparison of the performance of the **Nouns** cascade between the modelled classifiers and the execution of the cascade on unseen data.

4.6 Conclusion

This chapter discussed the view of morphological labelling as a classification problem. The labelling task was approached using supervised techniques and began by first exploring verbal inflections by exploiting the Ġabra database. A classification system was proposed whereby morphological properties were seen as machine learning features. Each feature was modelled by a classifier and placed in a particular sequence such that any dependency between morphological properties would be catered for by the cascade. A number of experiments were carried out to analyse the best data representation and the sequence of the cascade. The experiments also investigated whether merging of features and flattening the data would reduce ambiguity. However it was shown that these different datasets created classifiers which overfit the data, and when the cascades were executed on unseen data the Regular cascade performed best overall, which indicated that the classifiers learnt on the regular dataset allowed for better generalisation of the classification task. The analyses of the Regular cascade also revealed that nearly 60% of the unseen data was classified correctly throughout the cascade when discarding the last two classifiers, Verbtype and Form, which are only relative to verbs of Semitic origin.

The experiments then moved on to create Adjective and Noun cascades, following the same structure used in the Verb cascade. The dataset for these two cascades was obtained through an extractor which was extended purposefully for this research so as to gather more instances for the datasets. The two cascades performed very well on unseen data.

One of the main limitations of the research described above in this chapter was the evaluation of unseen data. The evaluation followed the traditional approach in machine learning, which is to have a certain amount of data that is excluded entirely from the training data and kept as a held out dataset. However, this did not reflect a truly realistic scenario, whereby if such a classification system were to be used on real data, that data would be similar to the MLRS corpus. To address this, a small scale evaluation was carried out using the MLRS corpus and described in §5.3.

The research described in this chapter focused on the individual cascades according to the relative part-of-speech. However in chapter 5, we describe how these three cascades were integrated into a single classification system which also includes the classification of the category.

The individual cascades offer modularity and, in future, as more and improved data become available, the classifiers could be retrained and their performance further improved. Improvement in the dictionary extractor could also result in better quality datasets. The inclusion of more second-tier features in the cascades is also possible as more labelled data becomes available, especially increasing coverage for particular morphological properties in the noun and adjective categories. Experiments can extend the work done for composite suffixes, especially to consider splitting suffixes into composites automatically, and apply composite suffixes to the noun and adjective cascades.

Chapter 5

Towards a complete Morphological Classification System

5.1 Introduction

The previous chapter described the task of morphological labelling as a classification problem. The morphological properties for verbs, nouns and adjectives were identified according to the labelled data available. A feature classifier was modelled for each of the properties where enough data was available. A number of experiments were carried out to determine the ideal sequence of a category cascade, resulting in three separate cascades for each of the parts-of-speech dealt with. An evaluation on unseen data was carried out for each cascade, together with an analysis of the results.

Having viewed each of the part-of-speech categories separately, we now turn our attention towards a more complete view of what a morphological classification system should be like. So far, the individual cascades assume that the part-of-speech category is known. The task of categorising a word's part-of-speech is usually done by a part-of-speech tagger, which looks both at the word itself and its context to determine its category. Although a part-of-speech tagger is available for Maltese[1] (Gatt and Čéplö, 2013), and achieves a very high accuracy, this research looked into the categorisation of words using the same machine learning techniques used so far. But rather than simply looking at the category task in isolation, as a step prior to morphological labelling, it also sought to analyse whether

[1] The POS tagger in question achieves ca. 97% accuracy and was used in the development of the "Korpus Malti v. 3.0 2016", which is now available on MLRS. The tagger is kernel-based, and was developed by training Maltese models on hand-tagged data using the open source SVMTool (for SVMTool see Giménez and Màrquez (2012) http://www.cs.upc.edu/ nlp/SVMTool/SVMTool.v1.4.pdf).

the morphological information of a word could feed back into a word's category classification, thus acting as a reinforcement mechanism to the category classification. Ultimately, the idea is not to propose a revisionist view of POS tagging as a task that should be carried out of context, but to identify the extent to which the basic features and the second-tier morphological features can contribute to the task of category identification, before context is also factored in. In principle, the idea is similar to Habash and Rambow (2005) and Lembersky et al. (2014) who incorporated part-of-speech and morphological labelling into a single system. This idea led to a proposed architecture that integrated category classification within it as a means of determining which cascade output should be chosen as the predicted output. Again, the category classifiers were then tested on unseen data.

The evaluations carried out in this research always focused on a 10-fold cross validation system during the development cycle of a classifier or cascade, and a final evaluation on unseen data. This is always the normal setting in machine learning. However, in moving closer towards a final morphological classification system, it was also important to evaluate the full system on more realistic data. A subset of the MLRS corpus was manually annotated during the development and training of the part-of-speech tagger for Maltese referred to above. This subset was first used to evaluate the category classifiers to analyse the performance of the category classifiers on more realistic data. A further evaluation selected 200 random words from this subset, and these were manually annotated by two linguists with their morphological properties. These *Gold Standard* annotations were then compared to the morphological labels outputted by the final classification system. The evaluation sought to analyse the strengths and weaknesses of the system, and what improvements would be necessary to arrive closer to a morphological classification system that could be used in a practical and realistic setting, such as that of providing morphological labelling to the MLRS corpus.

The rest of this chapter is structured as follows. §5.2 provides an initial overview of an abstract system architecture and describes the experiments carried out for the category classifiers, with the aim to integrate the classifiers into the architecture as a whole. The final Gold Standard evaluation of the full architecture is then described in §5.3, with a final conclusion in §5.5.

5.2 Category classification

In viewing morphological labelling as a classification problem, the various part-of-speech categories could be seen as separate *data streams* through which particular information flows according to its properties. Although category classification is normally done by a part-of-speech tagger, the following experiments looked into category classification with two principal aims. The first was to explore the possibility of creating an integrated architecture that takes a word, extracts its basic features, classifies its category, and processes it accordingly to extract its morphological information. The second was to determine to what extent would it be possible to classify parts-of-speech using the same technique that was used in the cascade classifiers.

The advantage of the cascade system used so far was that it is a modular system, and individual classifiers can be retrained in the future as improved data becomes available without necessarily having to modify the whole cascade. The modularity aspect was extended to the design of the architecture — fig. 5.1 shows an abstract view of the architecture. A word was first segmented and its basic features were extracted. It was first classified for the part-of-speech category using only the basic features. Irrespective of the word's resulting category, a word was passed through the three different cascades, each of which provided the predicted labels according to the different classifiers. A final augmented category classifier took all this additional information, and reclassified the word into its part-of-speech category. The idea behind this final category classifier was that the output of the cascades might help reinforce the category classification.

There are, of course, some drawbacks to the proposed system, especially in terms of the category classification. Part-of-speech taggers generally use context to label a word, calculating, for example, the probability that a determiner is generally followed by a noun. Here the classification was based solely on the word's basic features — its affixes and stem, CV-pattern and geminate features. These are not the usual type of features that a part-of-speech tagger might take into consideration. For instance, in English, words ending in *-ly* are generally classified as adverbs. However, using such features alone is not the norm, and generally the syntactic structure that words appear in provides more insight into their category classification. In developing a category classifier that was based on the basic set of features alone, it was possible to compare its performance to a regular part-of-speech tagger which is available for Maltese. The category classifier which used only the basic features is referred to as the Basic Category classifier. A second category classifier was also developed, augmented with the information outputted by the different cascades. This meant that apart from the basic features, it also used the proposed morphological labels to aid its category classification. This is referred to as the Augmented Category classifier. The Augmented classifier was of particular interest to determine whether the morphological labels could be used to reaffirm a category classification of a word and, if necessary, to

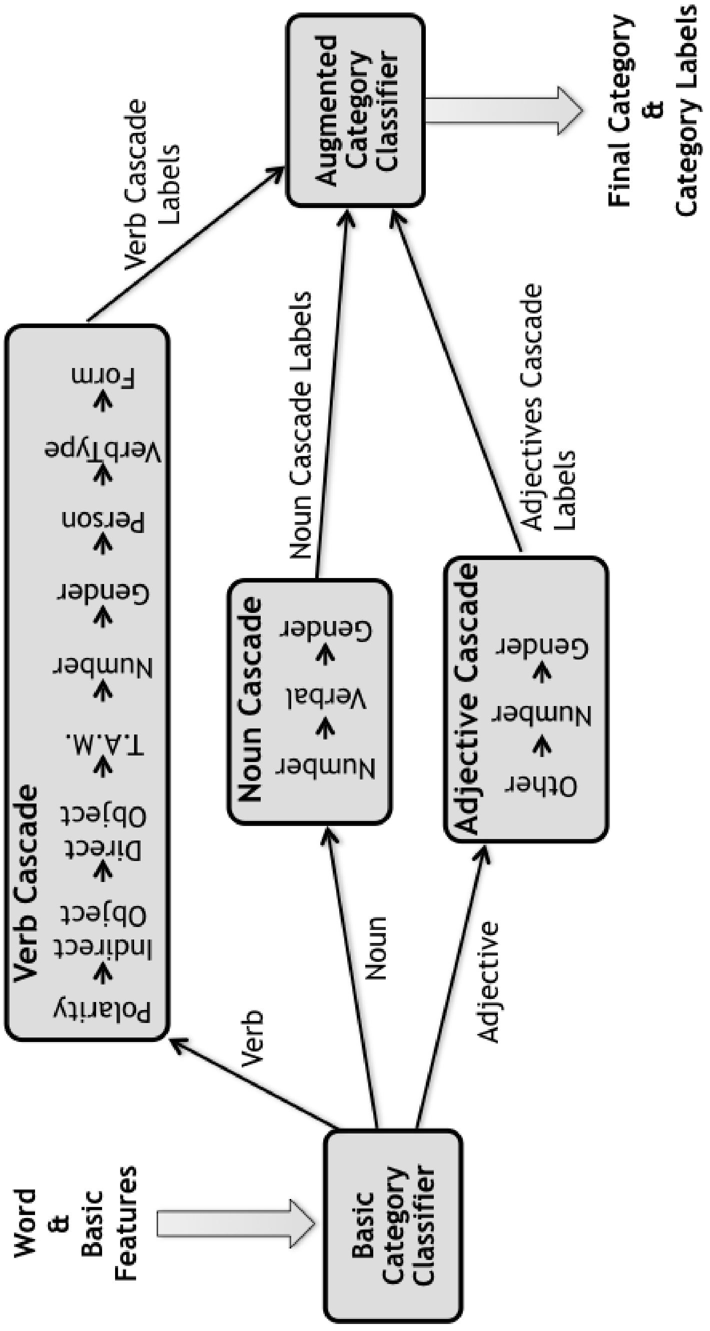


Fig. 5.1 An overview of the architecture for an integrated morphological classification system which includes category classification

modify the hypothesis about the category of a word. In a possible final system, it would be possible for both of these category classifiers at either end of the architecture to be replaced by a part-of-speech tagger that takes context into consideration. However, the experiments were limited to the same type of classifiers used so far on the cascades and which were reliant on the identification of the basic features of a word. The output of a part-of-speech tagger was then used to evaluate the category classifiers within the proposed system.

The proposed approach of combining part-of-speech tagging with the process of morphological analysis is similar in concept to that of Habash and Rambow (2005); Habash et al. (2013); Lembersky et al. (2014) described in §2.4 (page 43), where combining the two tasks provided better results in Semitic languages. The most challenging issue in part-of-speech tagging in Semitic languages is the large number of tags that is necessary for disambiguation. The Maltese tagger currently has 43 tags[2]. It could be that a future development of the Maltese part-of-speech tagger would result in a richer tagset; however it is unlikely to have the same complexity as present in, for instance, Arabic. Although there are these differences between Maltese and other Semitic languages, it is still worth following similar approaches to see whether morphological information could help category classification in Maltese. The proposed architecture is one way whereby morphological information could be integrated with part-of-speech tagging.

A final aim of setting up the proposed architecture, which included the category and cascade classifiers, was to evaluate the system as a whole on real data from the MLRS corpus, reflecting a possible 'working-scenario' for the proposed morphological classification system. For this purpose, a selection of the MLRS corpus which was manually tagged during the training of the part-of-speech tagger was used as the gold standard data for evaluating the system as a whole. This selection consisted of 10,410 word types with the part-of-speech tag being limited to nouns[3], verbs and adjectives. Other tags/words were ignored. This specific selection of the manually tagged corpus is referred to as the *gold standard corpus*.

5.2.1 Experiment setup

The category classification of a word could be approached in two possible ways: either building a 'singular' classifier which outputs the category (noun, adjective, verb), or building three separate 'one-vs-all' classifiers for each category which would output true or

[2] See http://mlrs.research.um.edu.mt/index.php?page=34 for a list of tags. Login credentials might be required but registering for an account to access the MLRS corpus is free.

[3] Proper nouns were excluded since our training data never included such nouns

false (noun vs. all, adjective vs. all, verb vs. all). The two different approaches could possibly yield different results, especially for the adjective category where the data was smaller in quantity. Whilst the initial basic category classifier could take both types of classifiers, the augmented classifier could not be directly modelled as a singular classifier due to different information configurations resulting from the different cascades. For example, the information related to polarity is found only in the verb cascade. The only way to have a singular augmented classifier would be to modify the output of the three cascades in such a way as to have it presented in a streamlined fashion. However this approach might unnecessarily complicate the data representation and make the interpretation of results less transparent. It might also result in excessive null-valued features for several classifiers. Therefore the augmented classifier at the end of the architecture was actually composed of three classifiers, each represented a one-vs-all approach for the respective category.

The proposed architecture introduces a probabilistic aspect in selecting a prediction. This made the Naïve Bayes machine learning technique more suitable for this type of setup. Naïve Bayes provides a probability associated with each label produced and this facilitates the choice between competing sub-instances. It also allows the system to choose the classification with the highest probability from the output of the three one-vs-all classifiers in the augmented category classification. Since the comparison of the different algorithms described in section 4.3.5 (page 130) did not find substantial differences in results between Decision Trees and Naïve Bayes, the shift to using Naïve Bayes for this analysis was not expected to have severe consequences.

5.2.2 Dataset preparation

The primary data sources used in training and testing the cascades so far were Ġabra for the verb cascade and the dictionary for the noun and adjective cascades. However, the dictionary also contained a number of verbs that could also be included in the training data for the category classifiers. This inclusion could provide better category classification results since the verbs in Ġabra focused primarily on inflective wordforms and words of Semitic origin. On the other hand, the dictionary also covers Romance or Anglo-Saxon origin verbs, as well as derivationally related words, thus providing better coverage to deal with the MLRS corpus. Using both sources of verbs, and training the category classifiers on all of the data, however, could result in an unbalanced dataset that would not reflect a realistic ratio of words/part-of-speech found in a corpus. Although the only comparison possible is on word types, the analysis provided in table 5.1 shows the distribution of verbs, nouns and adjectives from the sources Ġabra, the dictionary and the two added together compared to the distribution found in the gold standard corpus from the MLRS corpus. A proper analysis of the categories in the corpus would ideally be done on the basis of the word tokens rather than word types. If both the dictionary and Ġabra were used together, it could result in an unbalanced dataset made up predominantly of verbs.

Table 5.1 An analysis of the distribution between verbs, nouns and adjectives in different data sources, compared to the gold standard taken from the MLRS corpus

	Ġabra	Dictionary	Ġabra+Dict	Gold Std.
Verbs	(100%) 171,278	(15%) 8,465	(78%) 179,743	(50%) 5,237
Nouns	—	(67%) 39,678	(17%) 39,678	(40%) 4,136
Adjectives	—	(18%) 10,731	(5%) 10,731	(10%) 1,037
Total	171,278	58,874	230,152	10,410

In order to consider the different data distributions and the potential impact on the category classifiers, several tests were carried out using the training data but with different proportions of verbs and a different combination of sources (either Ġabra verbs only, or Ġabra and dictionary verbs combined). The results of these experiments indicate that the category classifiers produce the best results when trained on all the available data, i.e., all the verbs in both Ġabra and the dictionary, even though this results in over 78% of the data being verbs. The results below refer to this data configuration where all Ġabra and dictionary data is combined as a single dataset, unless otherwise specified. This dataset consisted of 230,152 instances, 204,288 of which form the training set and 25,864 were held-out as an unseen test set.

5.2.3 Category classification results

The basic category classification was carried out using only the basic features of a word. The different classifiers were trained using the training dataset, then tested on an unseen dataset, with the results shown below in fig. 5.2. The individual classifiers (one-vs-all) for each category performed better overall than a singular classifier which outputs a category classification. The drop in performance was mainly due to the adjective category, presumably because this category had the smallest number of instances in the dataset. The classifier type one-vs-all for the adjectives performed reasonably well, similar to the other two classifiers. Overall, the approach of category classification using one-vs-all classifiers provided better results. In this scenario, an instance is classified by the three different one-vs-all classifiers, and the classification with the highest probability is chosen as the predicted class.

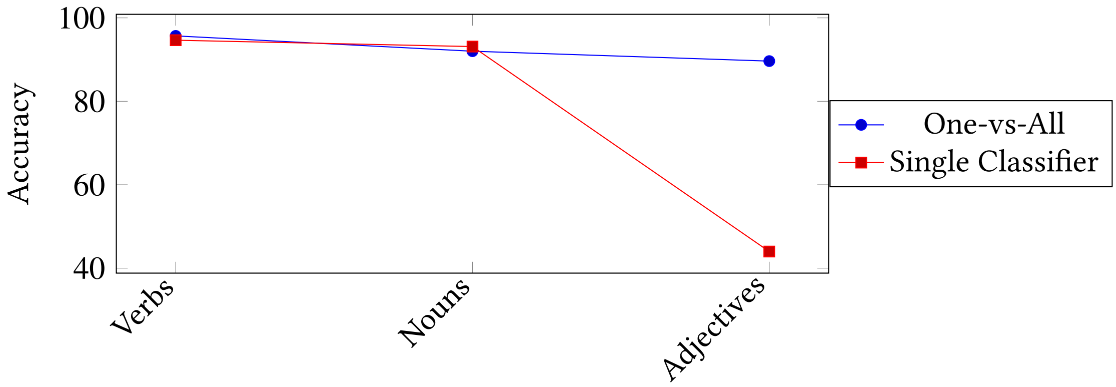


Fig. 5.2 Results for the basic category classification, showing the three individual 'one-vs-all' classifiers in comparison to a singular category classifier, tested on unseen data

The augmented category classification was set up as three individual one-vs-all classifiers, each at the end of the relative cascade, each using the basic features and the output of the relative cascade as input to predict a true or false classification. Each one-vs-all classifier was placed as final classifier for its respective cascade; for example, the verb-vs-all classifier placed at the end of the verb cascade and outputted a binary classification. When a word was processed by the whole architecture, it was classified by all cascades, and the final classification and cascade output was determined according to the highest probability from all the augmented one-vs-all classifiers. Figure 5.3 compares the accuracy of the one-vs-all basic classifiers to the augmented ones. An improvement is registered for the verb and adjective classifications, and practically no difference for the noun classification.

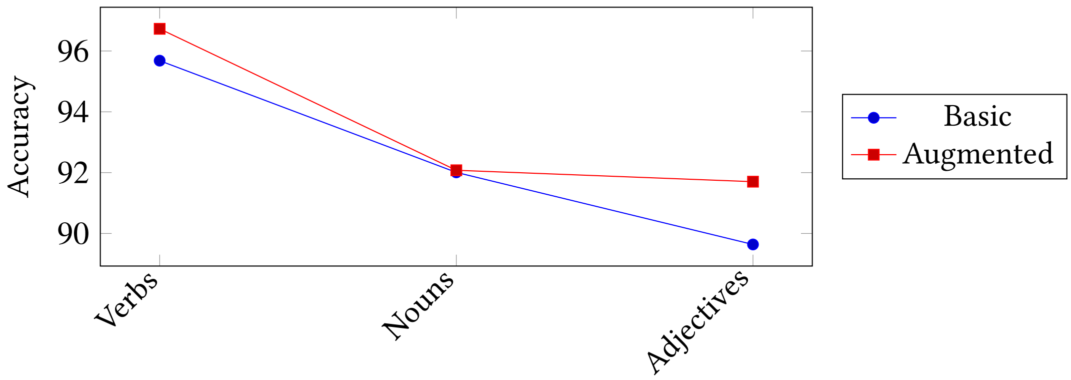


Fig. 5.3 Results of the three individual one-vs-all classifiers as Basic classifiers and as Augmented Classifiers on unseen data

The improvement of the augmented category classifiers over the basic category classifiers shows that, at least for the verb and adjective categories, the category classification is reinforced through a word's morphological labelling. It is possible that as the nouns cascade would be improved through more data and better representation of the category's morphological features, so would the category classification improve. Considering that the category classification is not using context and uses the basic features extracted from a word as its ground data, the overall results are positive. Of course, the analysis must take into consideration that the classifiers are being used in an 'artificial lab' setting — the testing was carried out on data that, although unseen, was similar to the training data since it came from the same source.

5.3 Gold standard evaluation

The classification system so far was evaluated on a component-by-component basis. Each evaluation focused solely on the classification of a particular section of the system, and using similar test data that, although unseen, was extracted from the same source as the training data. Of course, this was done in addition to the evaluation inherent in the cross-validation setup used in the initial experiments in each case. However, this had several implications such as, for instance, the set of basic features extracted from the words shared similar features between the training and the unseen datasets.

Evaluating the system as a whole on realistic data is an important aspect of a final evaluation, as it provides vital insights on the applicability of the current system of classifiers applied to data such as the MLRS corpus. Thus, the remaining part of this chapter focuses on the evaluation of the classification system using a small selection of words from the MLRS corpus that was manually annotated during the development and training of the part-of-speech tagger for Maltese. This subset of the corpus is referred to as the GOLD STANDARD corpus.

The first part of the gold standard evaluation looked again at the category classifiers. However, this time the evaluation used the gold standard corpus. This was processed by the full classification system and the output of the basic and augmented classifiers were compared to the part-of-speech tag in the corpus. The second part of the gold standard evaluation took 200 random words from this selection, and these were annotated with their morphological properties by two linguists. These annotations were then used to compare the output of the classification system.

One of the main challenges in carrying out an evaluation on data such as that found in the MLRS corpus is that of handling 'noisy' data. Most of the data sources used contained a large number of Maltese words with Semitic or Romance origin. Ġabra was a large database of words, a significant subset of which, especially among the verbs, was automatically generated via rule-based methods, whilst the dictionary contains a diverse selection of both Semitic and Romance origin words. However, naturally-occurring Maltese text is likely to be very noisy. For example, the use of English spelling interspersed with Maltese has become more common, as attested by the following examples taken randomly from the MLRS corpus.

(5.1) *Dan it-tip ta' divertiment kien iservi ta'* ***relax*** *għall-familja*
This the-type of entertainment was serve of relax for-the-family
This type of entertainment served as relaxation for the family.

(5.2) *Intervisti* ***live*** *ma' kittieba ewlenin fil-Kavallier ta' San Ġakbu.*
Interviews live with authors primary in-the-Cavaliier of St. James.
Live interviews with prominent authors at St. James Cavallier.

(5.3) *It-tkabbir fis-settur tax-****shipping*** *internazzjonali wassal għal*
The growth in-the-sector of-the-shipping international brought for
domanda akbar.
demand greater.
Growth in the international shipping sector brought greater demand.

Although English words are becoming more common in everyday use in Maltese, the presence of such words in our training datasets is minimal, if at all. Therefore the classifiers have not been exposed to such data, and it is difficult to predict what the performance will be on such words. This evaluation aims at providing a better insight into what would be required in the future to ensure that the morphological classifiers are sufficiently broad to deal with the hybridity of the Maltese language and its evolution.

Finally, the evaluation must also look at the classification system as a whole in terms of its technical architecture and the choices made. The abstract architecture described in fig. 5.1 (page 149) provides a point of departure. This was slightly modified to handle the pre-processing of words to extract their basic features. In the process of evaluating the morphological classification system as a whole, the system was employed in a more realistic scenario of processing words from a corpus that were unknown and all possible features must be extracted.

5.3.1 Extracting basic features

The extraction of basic features was an important and integral part of the whole classification system and the classifiers were dependent upon the identification of basic features. The system was limited to the list of features specified during the training phase of the classifiers — affixes and composite suffixes, CV-patterns of the word as a whole, and the stem, and whether there is gemination in the word or not.

One of the aspects that required consideration was segmentation. This task was particularly important to the system as a whole since the extraction of the basic features relied primarily on the segmentation of a word. Two approaches have been used so far. The first one was used in the clustering experiments and applied transitional probabilities to identify and rank possible affixes. The top ranked affixes were then used to segment a word. The second approach used the radical consonants within the Ġabra database to identify the stem of the word and the resulting affixes. The extraction of dictionary entries also produced a list of affixes since these were marked through vertical bars which separated the stem from the affix. Due to the nature of the different experiments, the affixes produced through Ġabra and the dictionary project were more accurate and offered better coverage than the affixes discovered through transitional probabilities. Another aspect to consider was that all the classifiers were trained using the list of affixes in Ġabra and in the dictionary.

The segmentation process adopted in the final system was similar to that used in the clustering technique (§3.2.4, page 62), which segmented words according to the matching of the affixes. This meant that a word could have more than one possible segmentation. Similar to the clustering approach, the morphological classification system used all suggested segmentations, together with the whole word unsegmented. The representation of a single word resulted in multiple sub-instances of the same word with different segmentations. The sub-instances were seen as 'competing' in the classification process, and the classification with the highest probability would be the one selected as the proposed classification, and indirectly selecting the proposed segmentation. Each of these sub-instances had a different set of basic features according to its identification of the stem and affixes.

The extraction of the CV-pattern for both the stem and the whole word was carried out using the third-party word-to-phoneme transcriber tool developed by Borg et al. (2011). The phonemes were then translated into consonant and vowel patterns. The identification of the geminate feature in the classification experiments was based on the radical consonants as a means to decide whether a word has one of its radicals duplicated. However, the radical consonants for the words in the MLRS corpus are not marked. The segmentation process provides multiple potential segmentations, and different segmentations could have potentially been used to determine whether the geminate feature is present or not, especially for the initial geminate case. However this could not be reliably determined and was therefore left out for this evaluation. Regardless, the impact of the geminate feature on the classification would have been rather minimal.

5.3.2 Choosing the best classification

The representation of a word resulted in a number of different sub-instances due to the use of all possible legal segmentations. This meant that if the segmentation process proposed three possible valid segmentations, plus the word itself unsegmented, the classification system represented the word as four different instances. In order to predict the possible classification of a word, all sub-instances were processed by the classification system and each instance was passed through the whole of the cascade irrespective of its category classification. The prediction of the correct instance was then considered as a function which would choose the best instance according to some performance-based criterion. This allowed flexibility in determining and experimenting with different possible functions to select the best prediction for a given word, and we refer to this as the *prediction function.* The classification system used Naïve Bayes classifiers since they provided both the predicted class and the associated probability for the respective prediction. When an instance was processed by the classification system as a whole, it resulted in a predicted class for each of the second-tier features and the category classifiers, as well as their associated probability value. The prediction function used the classifiers' probability values to determine which would be the best instance to select, and indirectly it would also choose a segmentation to represent the given word. The idea behind this approach was one which allowed flexibility in the choice of the final classification of a word, whereby the results of the classifiers would also be influencing the choice of the segmentation. This was an intuitive choice and posits that a correct segmentation would yield a better classification of the second-tier features and would result in higher probabilities than incorrect segmentations.

In order to determine what the ideal prediction function should be, an experiment was set up using the same datasets used for the training of the augmented category classifiers (described in section 5.2.2 page 151) as development datasets to test different prediction functions. The prediction functions were tested on the basis of the accuracy achieved in the augmented category classification — this meant that a prediction was considered correct if the final category classification matched the category of the word in the dataset.

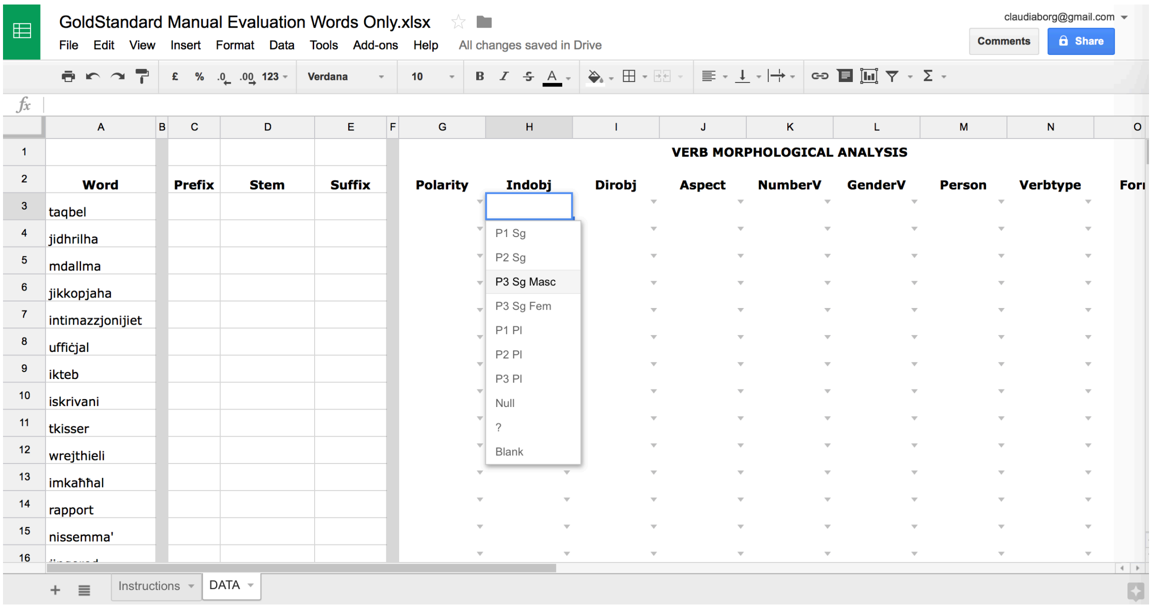
Three main prediction functions were tested. The first function (BASIC) simply chose the instance with the highest probability in the augmented category classification. Although this seemed to rely solely on the category classification, this classifier used all the second tier features for the relative cascade as part of its input. Therefore the whole cascade would be providing information indirectly to the final category classification and to the resulting probability of the classification itself. The second function (SHIFT) chose the instance with the highest increase in probability between the basic category classification and the augmented category classification. The idea behind this function was that the increase in probability demonstrates an improvement obtained in the classification through the cascade, which might indicate that the second-tier features and the final category classifications were more likely to be correct. The last function (InfoGain) chose a prediction based on the probabilities throughout the cascade, using a weighted function to consider the importance that a second-tier feature provided to the cascade according to its information gain ratio. This function was not as intuitive as the first two functions; however it considered the results of all the cascade classifiers directly.

The best prediction function was the BASIC function which picked the instance that had the highest augmented category probability. This function achieved an accuracy of 55% overall. The other two functions performed very poorly in comparison, with the SHIFT function having an accuracy of 28%, and the InfoGain function an accuracy of 20%. The gap in performance between the BASIC and the other two functions might indicate that if the full cascade has to be considered in choosing a prediction, a more complex and comprehensive function would be required to appropriately consider the intricacies that each classifier might be providing to the final prediction of an instance. With the BASIC prediction function in place, the classification system processed a word representing it as different possible instances. The system then used the prediction function to choose the best possible from the sub-instances according to some set parameters. The advantage of such a modular set up is that in future it could be possible to train a prediction function through a more representative development set.

5.3.3 Data preparation for manual annotation

The selection of 200 random words from the gold standard corpus that was annotated by linguists was limited to adjectives, nouns and verbs. Although 200 words was not a large number of words, it was sufficient to provide initial insight into the required improvements to the classification system and into how the system would fare when applied to the data in the MLRS corpus. This could then act as the basis of future work to transform this research into a more practical and final morphological classification system that could be integrated into existing tools for Maltese.

The extracted words were placed in a spreadsheet with instructions on how to carry out the task. The annotators were asked to (i) segment a word into prefixes, stem and suffixes (composite suffixes could also be split further but this task was optional); (ii) give the part-of-speech category of a word; (iii) fill in the morphological information of a word according to the chosen category. Figure 5.4 shows what the annotators were presented with, showing a list of words on the left hand side, with place where to list the prefixes, stem and suffixes, and the values for the morphological properties which had their input restricted according to their respective feature values. The figure shows a drop-down menu displaying the accepted input for the Indirect Object feature. Since the drop-down menu enforces selection once clicked, the option 'blank' was provided in case the annotator clicked on it by mistake. The '?' value was provided for when the annotator wanted to use a tag that was not part of the given options, or if the tag was not known. The value 'null' allowed annotators to state that this feature was not relevant for a particular word. These three options were present in all features, with the remaining values specified according to the feature in question.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1											VERB MORPHOLOGICAL ANALYSIS				
2	Word		Prefix	Stem	Suffix		Polarity	Indobj	Dirobj	Aspect	NumberV	GenderV	Person	Verbtype	For
3	taqbel														
4	jidhrilha														
5	mdallma														
6	jikkopjaha														
7	intimazzjonijiet														
8	uffiċjal														
9	ikteb														
10	iskrivani														
11	tkisser														
12	wrejthieli														
13	imkaħħal														
14	rapport														
15	nissemma'														
16															

Fig. 5.4 The spreadsheet given to annotators showing the selection for the Indirect Object feature.

The instructions also specified that if a word was ambiguous and had more than one possible tag, this should be indicated by duplicating the row for that particular word and the different morphological information would then be entered on the second row (or more rows if and as required).

The manual annotation was carried out by two linguists knowledgeable in tasks such as annotation and correction of Maltese texts. Both annotators filled in the spreadsheet for the whole set of words, providing the possibility to compare responses and agreement. The analysis of the results, discussed further below, compared the tags provided by the annotators to those provided by the classification system for the different morphological properties.

5.4 Analysis of results

This section describes the results of the evaluation carried out using the gold standard corpus (a selection of the MLRS corpus that was manually annotated with part-of-speech tags during the development and training phase of the Maltese part-of-speech tagger). These part-of-speech tags will be referred to as the gold standard tags (abbreviated to GS Tags where necessary). The first part of the results focuses solely on the category classification and compares the output of the classification system with the tags found in the gold standard corpus (§5.4.1). The second part of the results focuses on the evaluation carried out for the cascade classifiers (§5.4.2). The two are analysed independently for a number of reasons. First, the primary interest of this research was in morphological analysis, and thus the actual results of the cascades. For this evaluation it was necessary to rely on human experts to provide tagged data since this was not available. Second, the introduction of category classifiers into the architecture served as a means to exemplify the possible architecture, as a whole, and as an exploration to see how much context mattered in part-of-speech tagging.

5.4.1 Results for the category classifiers

The analysis of the category classifiers looked at their performance in classifying the gold standard corpus. This evaluation took the 10,410 word types and their categories as annotated in the corpus, and processed them through the full architecture setup. First, words were segmented, and those words with multiple segmentations were represented as multiple sub-instances. Each sub-instance was passed through the full architecture, irrespective of its known part-of-speech label. This evaluation was limited to the basic and augmented category classification results since the only tags available for these words were the part-of-speech tags. Table 5.2 shows the average precision, recall, F-measure and accuracy obtained by the basic and augmented category classifiers. Overall, the average performance of the category classification deteriorates when augmented with morphological information. However, fig. 5.5 shows the F-Measure for each category separately, where it is possible to see that the verb category classifier improved when morphological information is included as part of the features.

Table 5.2 Metrics for the category classifiers on the gold standard data

Classifiers:	Basic	Augmented
Precision	0.6398	0.5294
Recall	0.5866	0.5540
F-Measure	0.5706	0.5313
Accuracy	66.91%	64.07%

The verb category has the largest dataset which probably explains why only this category classifier improved through the cascade classification. On the other hand, the other category classifiers were all negatively impacted, with the basic classifiers performing better than the augmented classifiers. The adjective classifier also achieved a rather low F-Measure. The performance of both the adjective and noun classifiers can be attributed to the lack of data, especially by the lack of certain features in the training data, which might have been present in the gold standard. In comparison, the verb dataset included a large number of inflections and was further supplemented through words extracted from the dictionary, making the training data more diverse and with broad coverage.

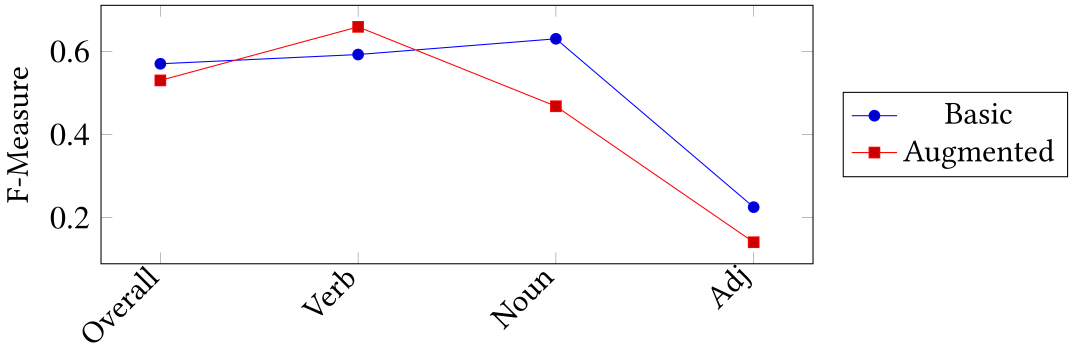


Fig. 5.5 F-Measure for the basic and augmented classifiers on the gold standard data

From the perspective of category classification, it was clear that the proposed architecture was a feasible one. Although the current part-of-speech tagger obtains far better accuracy, with enough training data this type of architecture reinforced the classification of the part-of-speech classification, in particular the verb category. The modularity of the system would allow the classifiers to be replaced by a part-of-speech tagger, thus introducing context into the features used to classify words. Improvements to the dictionary extractor could also yield more data for the noun and adjective categories, and the classifiers retrained with new data so as to improve the accuracy for these two categories.

5.4.2 Results for the three cascades

The analysis for the results of the three cascades focused on the manual annotation of 200 randomly selected words from the gold standard corpus. These words were annotated by two experts, who provided the segmentation, category classification, and morphological information according to the category classification. The words also had the gold standard part-of-speech tag provided through the manual annotation of the corpus, however, this was not provided to the annotators. Since the manual part-of-speech annotation was done in context, it was considered as the correct tag. The two experts had only a wordlist at their disposal, without any context, making the annotation task possibly more challenging or ambiguous. The experts were asked to provide multiple classifications in case of ambiguity. However, both experts provided a single annotation for each word. There was some difference when comparing the part-of-speech tags with those provided by the two experts. Table 5.3 shows the number of instances per category according to the gold standard part-of-speech tag, and the tags given by the two experts and the classification system. Table 5.4 then shows the Precision, Recall, F-Measure and Accuracy obtained by the classification system, where the results are compared to the labels by expert 1, expert 2, their average, and the gold standard part-of-speech tag.

Table 5.3 Comparison of the category classification: frequency of category tags as per the GS tag, the tags by the two experts, and the augmented classification given by the system

Category	GS Tag	Exp1	Exp2	System
VERBS	94	83	86	146
NOUNS	78	81	78	49
ADJ	28	33	33	5
Not Known	–	3	3	–

The results of the system on the subset of 200 words were also compared to the original results described above in table 5.2. The overall accuracy for the augmented category clas-

Table 5.4 Performance of the augmented category classifier versus the tags by the two experts and the GS tag

	Expert 1	Expert 2	Average	GS tag
Precision	0.5102	0.4988	0.5045	0.5411
Recall	0.5200	0.5150	0.5175	0.5500
F-Measure	0.4618	0.4575	0.4597	0.5045
Accuracy	65.75%	63.40%	64.57%	65.25%

sification remained the same at around 65%. Although the main interest in this evaluation was the classification of the morphological features, analysing the category classification results demonstrated that the annotation task might still be prone to errors. The following analyses looked at the individual categories and compared the results of the second-tier classification to the tags provided by the experts. The data was divided into categories according to the gold standard tags, especially since there were some discrepancies between the category classification by the experts. The data was collated into a single spreadsheet and manually analysed to extract the Precision, Recall, F-Measure and Accuracy for the different cascades and classifiers. Since F-Measure and Accuracy are sometimes plotted together in the same graphs, both are presented as decimals ranging from 0.0 to 1.0 so as to enable better visual comparison of the two metrics.

5.4.2.1 The verb cascade

The verb cascade consisted of nine classifiers for the second-tier features that contain morphological information. From the set of 200 randomly selected words, 94 were verbs. The following analysis looked at the labels predicted by the classification system for these 94 words and compared them to the labels provided by the two experts. The analysis here ignored what category the classification system predicted for these instances, and simply focused on the correctness of the cascade of second-tier feature classifiers. The rationale behind this approach in analysing the results is that it would be possible for the system to obtain the part-of-speech category from the tagger and as a result pass it through the correct cascade immediately. Therefore the analysis looks at all the 94 instances, and compares the output of the verb cascade for the sub-instance that obtained the highest augmented category probability. From the 94 verbs, the classification system predicted 79 of them to be verbs. The system still provided the output of the verb cascade on the remaining 15 instances. Thus the labels for the second-tier features were analysed as though the part-of-speech category was known in advance and a word was immediately passed through the correct cascade.

In analysing the results, a discrepancy in the data was noted. Specifically, the Gender classifier, as originally trained, included the values “invariable” and “neuter”. However, these values turned out to be extremely rare in the training data. By contrast, the annotators did use these values occasionally in their annotation exercise, giving rise to the possibility that the classifier’s performance might be affected by a bias introduced during training. In order to provide a more balanced analysis, the results of this classifier are presented twice — once with the values of invariable and neuter taken into consideration, and an alternative analysis considering the invariable and neuter tag as equivalent to blank tags. This meant that if an expert classified an instance as neuter, and the system classified it as blank, in the alternative analysis this was considered as correct. However, if an expert classified an instance as neuter, and the system classified it as masculine, then this was considered incorrect in the alternative analysis. The alternative analysis is marked as Gender-Alt and simply represents the same classifier as Gender, but providing a fairer analysis of results according to the data that was present during the training of the classifiers.

The average Precision, Recall, F-Measure and Accuracy for the verb cascade is shown in fig. 5.6. This was the average obtained from analysing the classifications against both experts. Figures 5.7 and 5.8 show the results for expert 1 and 2 respectively. Looking in detail into these results, overall Precision tends to be slightly higher than Recall, especially for the classifications compared to those of Expert 2. This indicates that when instances are classified as a particular class, the classification tends to be correct, resulting in a higher quality of the classification. However, there are several ‘misses’ in the classifications, so not all instances of a particular class are being captured accordingly. The higher Precision is also reflected by a higher level of Accuracy, which looks only at the percentage of correctly classified instances.

The figures also show the difference in the analysis of the Gender classifier taking neutral and invariable into consideration, and the alternative analysis which considered these two values as blank. As predicted, Gender-Alt provides a better performance, more in line with the remainder of the cascade. The first three features in the cascade — Polarity, Indirect Object and Direct Object — perform best for all the metrics considered. It was always evident, even when carrying out the initial experiments to determine the sequence of the

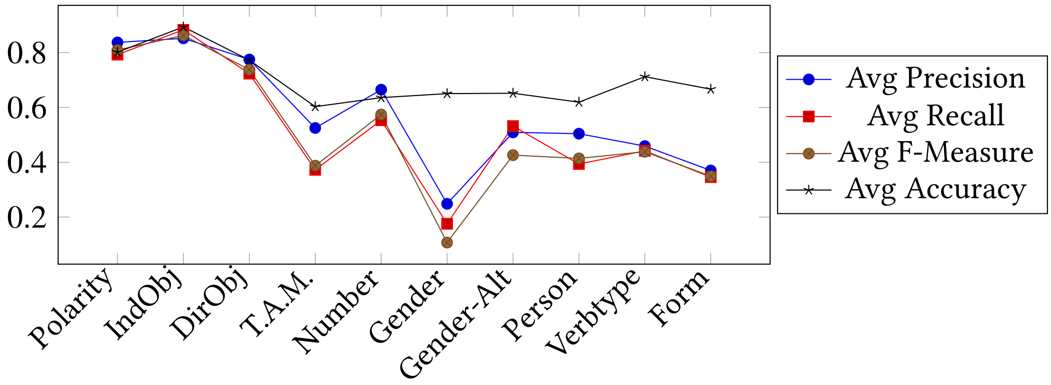


Fig. 5.6 The average Precision, Recall, F-Measure and Accuracy of the second-tier classifiers for the verb cascade evaluated according to the labels provided by the two experts.

cascade, that these three second-tier features were easier to classify and were not reliant on other second-tier features. The drop in performance for the T.A.M. classifier (tense, aspect, mood) can be expected since this was also observed in previous analysis, especially because of the problems with the representation of the imperative in the training data. Similarly, the classifiers for Person, Verbtype and Form did not have a very high performance when they were tested, so the level in performance on the gold standard data is very adequate when taking into consideration the type of data that the classification system is being tested with.

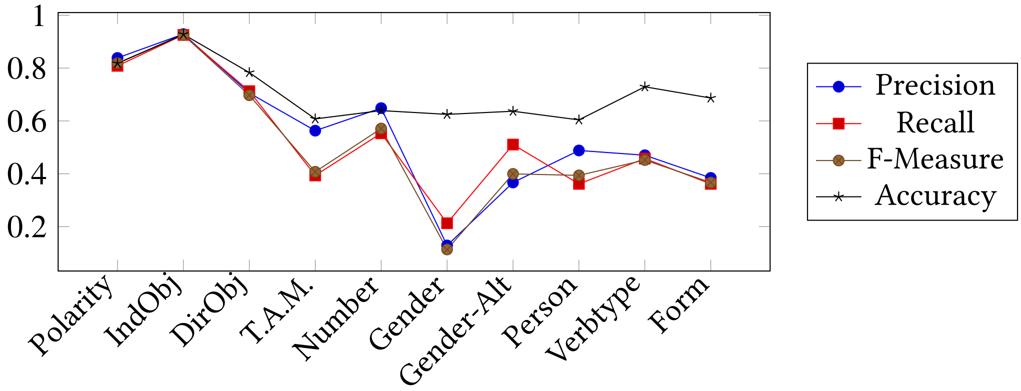


Fig. 5.7 The average Precision, Recall, F-Measure and Accuracy of the second-tier classifiers for the verb cascade evaluated according to the labels provided by Expert 1.

The results of the classification cascade on the gold standard data were also compared to the previous accuracy analysis carried out on unseen data for the verb cascade (§4.3.3,

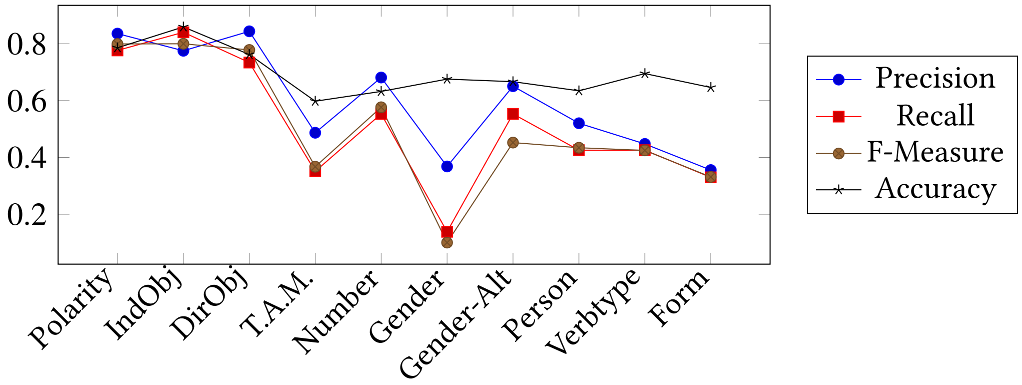


Fig. 5.8 The Precision, Recall, F-Measure and Accuracy of the second-tier classifiers for the verb cascade evaluated according to the labels provided by Expert 2.

page 121). Figure 5.9 shows the average accuracy the system obtained on the gold standard data versus the previous accuracy obtained when testing the classifiers on held out data. This analysis shows the impact on the performance of the classifiers from different data. The unseen dataset, although different from the training data, was still of the same 'shape' — the segmentation of words were known, the distribution of instances in the different classes was more or less similar to that found in the training data — such aspects aid the classifiers to achieve better results. The higher previous accuracy indicates that it could be possible to achieve better results in most classifiers. On the one hand, it could indicate that the training data for the classifiers did not cover the necessary breadth for the verbs found in the MLRS corpus. This could be particularly true since the majority of verbs in the training set were from Ġabra, which had a very high percentage of verbs that followed a root-and-pattern morphology. Whilst the verbs found in the MLRS corpus might be much more diverse and mixed, representing the different facets of the hybrid morphology system in Maltese. On the other hand, it could also indicate that the segmentation and the process to select and predict a sub-instance need to be improved further for the classifiers to obtain better results.

5.4.2.2 The noun cascade

The noun cascade consisted of three classifiers for the second-tier features, and from the set of 200 words, 78 were categorised as nouns. The analysis below looked at the labels predicted by the classification system for these 78 words, and compared them to the labels provided by the two experts, irrespective of the category proposed by the classification

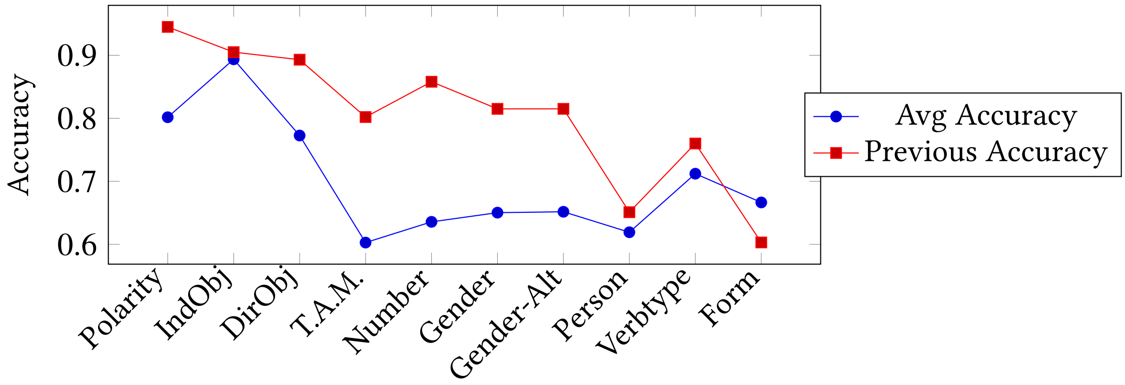


Fig. 5.9 The average Accuracy of the second-tier classifiers for the verb cascade compared to the previous accuracy of the classifiers on unseen data from the regular dataset.

system. Figure 5.10 shows the average Precision, Recall, F-Measure and Accuracy for the predicted classifications when compared to the labels provided by the experts.

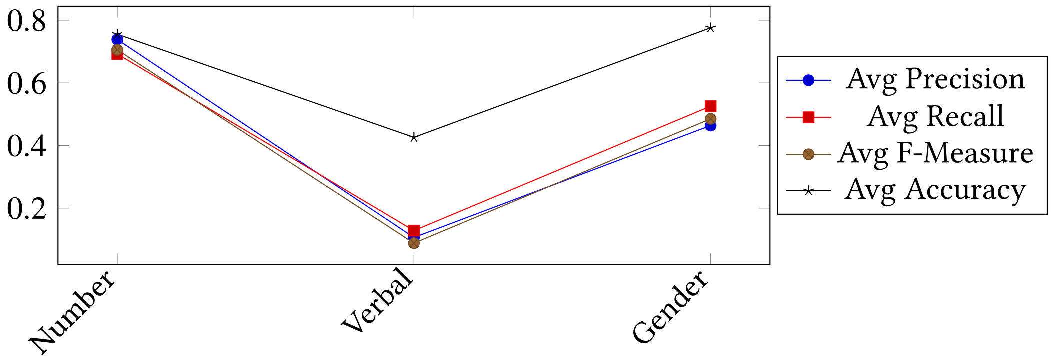


Fig. 5.10 The average Precision, Recall, F-Measure and Accuracy of the second-tier classifiers for the noun cascade evaluated according to the labels provided by the two experts.

Figure 5.10 shows the average Precision, Recall, F-Measure and Accuracy for the noun cascade, comparing the predicted labels by the classification system to the labels provided by the two experts. The performance of the number classifier was quite high. However, the performance of the verbal classifier for the noun cascade was particularly low. This was because the classification misses most of the verbal predictions. The difficulty in classifying this feature becomes more apparent when analysing the level of disagreement between the two experts. From the 78 nouns, 15 were marked as verbal by both experts and 43 were marked as not verbal, with a remaining 20 where the experts disagree in terms of classification. This constitutes 25% of the words where the experts did not agree whether a word should be classified as verbal or not.

Figure 5.11 compares the average accuracy of classifiers on the gold standard data to the previous results obtained by the cascade on the unseen data. The only discrepancy between the two evaluations was for the Verbal classifier. The Number and Gender classifiers both achieve very similar accuracy on the MLRS data and the unseen data. This was a positive result, indicating that the Number and Gender classifiers were trained with enough data to handle a more generic source like the MLRS corpus.

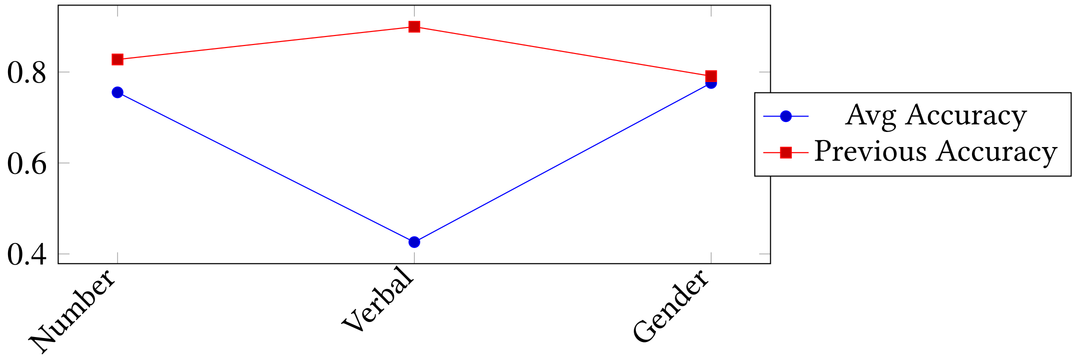


Fig. 5.11 The average Accuracy of the second-tier classifiers for the noun cascade compared to the previous accuracy of the classifiers on unseen dataset.

5.4.2.3 The adjective cascade

The adjective cascade analysis was based on a small number of words and from the 200 words, 28 were adjectives. Similarly to the previous cascades, this analysis compared the predictions of the classification system for these 28 words to the labels provided by the two experts. Figure 5.12 shows the average Precision, Recall, F-Measure and Accuracy for the predicted classifications when compared to the labels of the two experts.

The high accuracy of the Other classifier (which mainly represents the feature Agent) was primarily due to a large number of instances being marked as blank by both the Experts and the system. In fact, the system proposed only two instances predicted as agent and both experts did not provide any labels to any of the words, thus resulting in such a high accuracy. The Number and Gender classifiers perform adequately given the data sparsity for this category. Figure 5.13 compares the average accuracy of the cascade applied with the gold standard data to the accuracy of the cascade classifiers applied to unseen data that was used to evaluate the classifiers. The resulting observations for the Number and Gender classifiers were similar to previous analysis above.

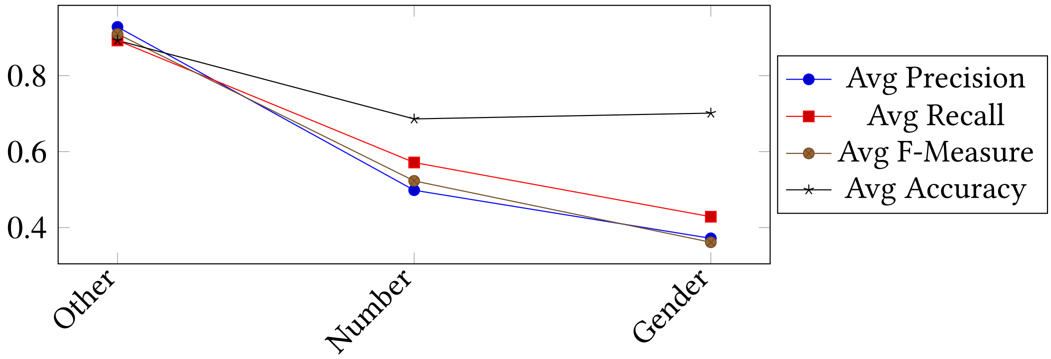


Fig. 5.12 The average Precision, Recall, F-Measure and Accuracy of the second-tier classifiers for the adjective cascade evaluated according to the labels provided by the two experts.

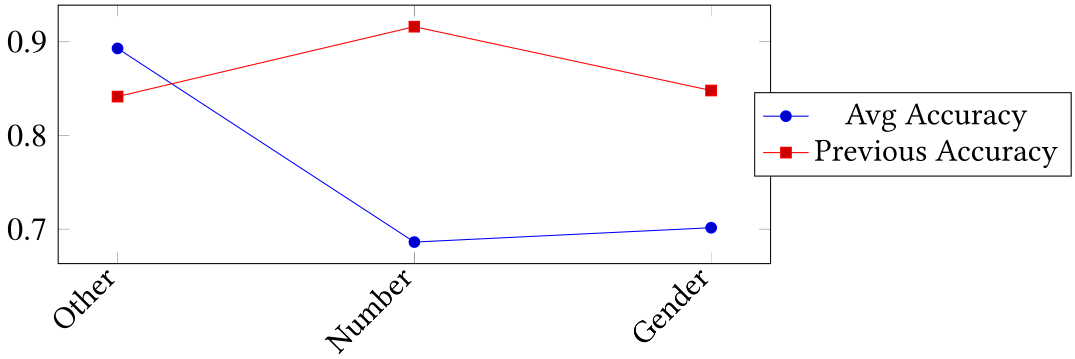


Fig. 5.13 The average Accuracy of the second-tier classifiers for the adjective cascade compared to the previous accuracy of the classifiers on unseen dataset.

5.5 Conclusion

This chapter shifted the focus from the individual cascade classifiers described in chapter 4 to a broader view by proposing how these cascades could be integrated into an architecture as a full morphological classification system. As a result, the first step was to introduce category classification to determine which of the cascades should be used to classify a word. However, the design of the architecture and the experiments went a step further, and questioned whether the output from the cascades could also be used to reinforce the classification of a word's category. This resulted in a system where category classification was carried out both in the beginning, before the cascades, as well as at the very end, after all the cascades were executed. The results were promising, showing that for the category which had most representative data, the verb category, the morphological information did improve the accuracy in classifying words of this category.

Another important factor described in this chapter was the evaluation of the system on the basis of gold standard data, both where the category classification is concerned (available from the manual POS-tagged MLRS data) and where the segmentation and labelling is concerned (done for the purposes of this study). The purpose behind this evaluation was to get an indication of what the performance of the different classifiers would be if the system were to be applied to a corpus such as the MLRS. The evaluation used a portion of the corpus which was manually tagged with part-of-speech during the development and training of the Maltese part-of-speech tagger. These tags were considered to be the gold standard category classifications, and these were compared to the category classification outputted by the system. The system achieved an overall accuracy of 55%; however this was at 75% for the verbs category. The results clearly indicated that more data would required in the noun and adjective categories to improve accuracy further.

A final evaluation took a subset of 200 randomly selected words to evaluate the output of the actual cascades. Two experts annotated these words with morphological information and provided a segmentation and category analysis. This was the first set of words to be manually annotated with morphological information in Maltese. Although the set of words is limited in size, it was sufficient to highlight the various aspects and pitfalls of the system. First, the segmentation technique and the prediction function must be improved. The segmentation currently provides a number of possible segmentations for a word, resulting in a word being represented by a number of different instances. The prediction function then selects the best performing instance. Although both offer adequate solutions with the current data available, these techniques need to be refined to increase the accuracy of the system as a whole. Second, the data sparseness in the adjectives category and, to a lesser extent, the nouns and verbs category impacts the accuracy levels and more data is required to improve the classification results.

Notwithstanding the above, the general results are positive, since it was expected that the classification system would not have the same level of accuracy on the MLRS data as when it was tested on unseen data from the same source as the training data. The MLRS corpus contains a different coverage of the Maltese language than the data used for the training of the classifiers. However, the results demonstrate the viability of the approach taken and show what is required to obtain further improvements to bring the morphological classification system to a higher degree of accuracy.

Chapter 6

Conclusion

6.1 Introduction

This thesis presented the first comprehensive and systematic treatment of Maltese morphology using machine learning techniques. Previous approaches were either rule-based, or restricted in their scope, focusing only on one sub-system of Maltese morphology. The research looked at three different aspects of computational morphology — segmentation, relations and labelling. It also sought to look into whether the techniques could be equally applied to the different morphological processes in Maltese. As noted at the outset, in chapter 1, one of the challenges of dealing with Maltese morphology in a unified manner is the hybrid nature of the morphological system, which has elements of both templatic systems typical of Semitic languages, and stem-based systems typical of Indo-European ones. Although the focus of this work was to deal with the hybridity aspect, the techniques used were language-independent. In concluding this work, we will first give a summary of the main contributions and techniques used, before turning to the limitations and future work.

6.2 Summary and main conclusions

The segmentation techniques used both a rule-based and an unsupervised approach. The unsupervised approach, which was exploited in the clustering and segmentation of morphologically related words, implemented a technique which used transitional probabilities, based on Dasgupta and Ng (2007); Keshava and Pitler (2006), and produced a list of ranked affixes. The affixes were then used to propose various possible word segmentations. The rule-based approaches were based on heuristics pertaining to the particular data that was segmented. The lexicon Ġabra was segmented using the radical consonants found within its database, whilst the words extracted from the dictionary were segmented according to the positioning of the vertical bar in a head word. Although these techniques gave better and more definite results in segmenting a word, they were restricted to the specific wordlists in the respective datasets. However, the final approach to segmentation in the proposed classification system combined different aspects of the two systems. The process proposed potential word segmentations using the same strategy as the unsupervised approach, but utilised the list of affixes obtained from Ġabra and the dictionary, since these were more likely to be accurate. One of the main difficulties with the segmentation task was that since there is no lemmatizer or stemmer for Maltese, the resulting segmentations could not be evaluated directly. This meant that the choices made were based on the general observations over the resulting data. Having an unsupervised approach to segmentation which proposed multiple possible segmentations to the following tasks was ideal in this scenario, and the word segments were indirectly evaluated through the results of the clustering and labelling tasks.

The clustering task also used an unsupervised approach, with the aim of grouping morphologically related words together. The segmentations obtained through the unsupervised approach were used, and initial clusters built on the basis of common stems. To improve the initial clusters, semantic and orthographic similarity were introduced through a combined metric that measured the proximity of words in a cluster, an idea similar to that of Baroni et al. (2002); Schone and Jurafsky (2000, 2001). This resulted in an 80% reduction in the number of clusters. A random sample of clusters were evaluated through crowd-sourcing by native Maltese speakers as the non-expert group, and an additional selection of clusters were evaluated by three linguists as the expert group. The non-expert evaluation sought to gather a general sense of the quality of the clusters and how successful the technique was in putting together only related words. The expert evaluation also looked at this aspect, but went a step further to analyse how the techniques used fared on the concatenative versus the non-concatenative aspects of Maltese morphology. The results demonstrated the difficulty of finding a one-size-fits-all solution. The overall quality of the evaluated cluster was quite adequate, and more than half of the clusters had no words removed and a very small percentage of clusters had several words removed. The quality ratings also reflected this perception, with over 70% of the clusters rated good or very good in the non-expert group. The inter-annotator agreement was rather high, which meant that there was a general agreement on which words should be removed from a cluster. This is an important aspect since in future the evaluated clusters could be used for developmental purposes to improve the clustering algorithm. The comparison between the concatenative and non-concatenative clusters also revealed that the techniques generally fared better on the former type of morphological process, but could still be applied to the latter with a little less success.

The labelling task took a supervised approach, and viewed morphological labelling as a classification task. Each morphological property was seen as a machine learning feature that could be classified according to a set of possible values. The first experiments exploited the data from the lexicon Ġabra, and was mainly focused on inflective verbs of Semitic origin. This was later broadened to nouns and adjectives after the inclusion of the data from the dictionary project. For each of the categories, a number of experiments were carried out to determine the optimal sequence of the feature classifiers. Each cascade was then evaluated using unseen data which was put aside from the same source as the training data. One of the most positive results was that the verb cascade classified 60% of the unseen instances correctly throughout until the Person classifier (i.e. leaving out the Verbtype and Form classifiers from the cascade).

Once the cascades were in place, it was possible to define a full morphological classification system, which introduced category classification as a means of determining which cascade output should be chosen as the predicted output. For the cascades to be integrated into the proposed system, category classifiers were developed, with a basic category classifier at the beginning of the cascades, and an augmented category classifier at the end of each cascade. Although category classification was already available for Maltese through the part-of-speech tagger, the purpose of these classifiers was to examine to what extent morphological information helps to determine the category (which also sheds light on the potential contribution of the morphological labelling task to POS tagging in context).

A final evaluation was carried out using gold standard data. The purpose of the evaluation was to have an indication of how the system would perform on the data such as the MLRS corpus. In the case of the category classifiers, the gold standard evaluation used a portion of the MLRS corpus that was manually tagged during the development and training of the part-of-speech tagger. Two hundred words were then annotated by experts for their morphological properties and these were used to evaluate the actual cascades. As expected, the performance of the cascade classifiers overall was lower than their performance on the unseen data, bar a couple of exceptions, due to the nature of the words found in the MLRS corpus. However, such an evaluation is always important since one of the end goals of a morphological analyser would be to process the corpus.

All the techniques used were language-independent, and therefore, could be applied to other languages as well. However, the design of a cascade classification system is better suited for morphologically rich languages. The segmentation and clustering techniques require only a corpus to obtain a frequency list, making them ideal for bootstrapping morphological resources for low-resourced languages. The labelling techniques require actual labelled data to train and test the classifiers built.

6.2.1 Principal challenges and themes

The hybridity of the Maltese language and its morphology remained an important theme throughout the research. In segmentation, one of the main advantages of the morphological system in Maltese is that the inflectional processes are mainly produced through concatenative affixation, even when stem variation occurs. The difficulty was more in the identification of the actual stem when it came to the segmentation process. Since there was no segmenter tool for Maltese against which the segmentation accuracy can be measured, it was difficult to truly measure its success as a technique on the concatenative versus non-concatenative words in an appropriate way. Since the unsupervised clustering task relied heavily on the preceding segmentation task, the lower quality in the non-concatenative clusters could have been partially due to the segmentation. However, it was difficult to measure this concretely with the data and tools available at the time.

The classification cascades did not distinguish either between the concatenative and non-concatenative systems in the language. However, in the case of the verb dataset, this primarily came from the root-and-pattern based processes (non-concatenative), namely the Ġabra database. The noun and adjective cascades had a more varied data representation, since the dictionary data contained words from both morphological systems. The performance of the classification system generally remained quite close to the model when evaluated on the MLRS corpus. However, the analysis did not go as far as to check the percentage of the balance between the two processes within the data used.

6.3 Future work

The research presented in the preceding chapters opens up several avenues for future work. The importance of the segmentation task was highlighted both through the clustering and the labelling tasks, as a foundational and essential task whose accuracy will impact any following tasks. The various segmentation techniques attempted in this research were limited to the specific task served or the nature of the data being segmented. Further development could look at log-linear models to segment words, such as those proposed by Narasimhan et al. (2015); Poon et al. (2009), to find the best possible segmentation. Such a system could now be augmented with information on the possible affixes which have been extracted from Ġabra and the dictionary data, thus using a semi-supervised approach with a mix of rule-based (affix information clearly encoded with the system) and unsupervised (deciding whether a potential stem is correct on the basis of a pre-trained model) approaches. This type of approach would be ideal to arrive at a useable segmenter for Maltese, since it is difficult to achieve a highly accurate segmenter using only unsupervised techniques. Another view of the segmentation task would be to attempt the techniques on the phonetic transcription rather than the orthographic representation of words.

The clustering techniques could be fine-tuned, especially in the definition of the metric which was used to determine how good a cluster is. The metric, based on orthographic and semantic similarity, was defined without the use of a development set to test the metric and the different weights used. The refinement of this metric could yield a better clustering technique. The clustering technique could also be applied to the Ġabra and the dictionary data, where the idea of paradigms and word families are quite well defined within the data itself. The evaluation of the clusters, in particular through the test clusters, revealed that native Maltese speakers treat derivationally related words in a different way. As described in §3.5.1 (page 81), participants removed certain derivationally related words from the test clusters, but not others. This seems to indicate that there are different conceptual connections in the mental lexicon for derivationally related words. Through Ġabra and the dictionary data, it would be possible to put together clusters that could explore this further, and analyse if there are particular derivational morphological processes that draw the conceptual line between the original word and the derived word, thus no longer being considered related at all. Such a study would take a psycholinguistic perspective to Maltese morphology.

Similar to the development of the part-of-speech tagger, a final morphological analyser could be implemented in a more incremental approach. Creating a cycle between improving the data, retraining the classifiers, evaluating and finding systematic errors where the classifiers do not perform well, and again improving the data for these errors. In the short term, we plan to retrain the classifiers on the data that has been manually checked and corrected from the dictionary project. This should yield improvements in the classifiers, especially for the noun and adjective cascades. These two cascades also require further data to gain a broader representation of their morphological properties. Future research could easily use the same experimental setup used in this research to determine how new features should be classified in the cascade. Experiments in the verb cascade should analyse whether introducing Romance-origin verbs and their inflections would be detrimental to the performance of the classifiers, and if so, why? The verb data can be supplemented by extracting the Romance verbs from the dictionary data, and extending the grammar by Camilleri (2013) to generate all the inflective forms as was done with the root-based system. This would increase the coverage for the verbal morphology further, allowing the classifiers to factor in conjugations where stem variation occurs and where it does not.

Once the data is more balanced between the two morphological systems, a further experiment that this work did not directly look into is whether words can be automatically classified according to their origin or morphological process that they follow. If this were possible, apart from the initial category classifier, a future morphological classification system would also classify a word's processing path — stem-and-affix or root-and-pattern. This would be very beneficial should it turn out to be the case that the classifiers' performance degrades if trained on words with mixed morphology systems.

6.4 Final conclusion

This research set out to explore the morphological analysis for Maltese from a computational perspective. Since previous approaches were either rule-based, or limited to particular sub-systems of Maltese morphology, the possible directions available to this research were quite broad. We chose to attempt solutions for the three main aspects in computational morphology: segmentation, relations and labelling. The end result is a morphological classification system for verbs, nouns and adjectives. Although it might not yet have achieved a sufficiently high accuracy, it certainly provides the foundations for a more complete morphological analyser, with broader coverage. Furthermore, the research also focussed on using language-independent techniques, opening up the possibility of further exploration on datasets from other languages. This could result in further refinement of the techniques themselves and identifying the characteristics of different morphological systems that determine how well certain techniques perform. The scope of the research was not merely a technological one, to create a morphological analyser, but rather to investigate the hybridity of the morphological system in Maltese and how this impacts the results of different techniques. The insights yielded by the techniques used here were intended both as a case study of how computational tools can contribute to morphological research and as an exploration into the practical consequences of harnessing machine learning techniques towards the development of fundamental NLP tools for Maltese.

References

Malin Ahlberg, Markus Forsberg, and Mans Hulden. Semi-supervised learning of morphological paradigms and lexicons. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden 26–30 April 2014*, pages 569–578, 2014.

Alias-i. Lingpipe 4.1.0, 2008. URL http://alias-i.com/lingpipe. Online; last accessed 5-July-2012.

Saba Amsalu and Dafydd Gibbon. A Complete FS Model for Amharic Morphographemics. In *Finite-State Methods and Natural Language Processing*, volume 4002 of *Lecture Notes in Computer Science*, pages 283–284. Springer Berlin Heidelberg, 2006.

Evan L. Antworth. Glossing Text with the PC-KIMMO Morphological Parser. *Computers and the Humanities*, 26(5/6):389–398, 1992.

Joseph Aquilina. *Maltese-English Dictionary.* 2 Volumes. Malta: Midsea Books, 1987–1990.

Ron Artstein and Massimo Poesio. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555—596, 2008.

Duncan Paul Attard. A lexicon server toolkit for Maltese. Bachelor's Thesis, University of Malta, 2005.

R. H. Baayen. Storage and computation in the mental lexicon. In G. Jarema and G. Libben, editors, *The mental lexicon: Core perspectives*, pages 81–104. Elsevier, 2007.

Marco Baroni, Johannes Matiasek, and Harald Trost. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning - Volume 6*, MPL '02, pages 48–57. Association for Computational Linguistics, 2002.

Kenneth R. Beesley. Arabic finite-state morphological analysis and generation. In *Proceedings of the 16th conference on Computational linguistics*, pages 89–94. Association for Computational Linguistics, 1996.

Kenneth R. Beesley and Lauri Karttunen. Two-level rule compiler, 2003.

Jean Berko. The child's learning of English morphology. In *Word*, volume 14, pages 150–177, 1958.

Albert Borg and Marie Azzopardi-Alexander. *Maltese: Lingua Descriptive Grammar.* Routledge, London and New York, 1997.

Mark Borg, Keith Bugeja, Colin Vella, Gordon Mangion, and Carmel Gafa. Preparation of a free-running text corpus for Maltese concatenative speech synthesis. Abstract in the 3rd International Conference on Maltese Linguistics, 2011.

Joseph M. Brincat. *Maltese and other Languages.* Midsea Books, Malta, 2011.

Tomáš Brychcín and Miloslav Konopík. HPS: High precision stemmer. *Information Processing & Management*, 51(1):68 – 91, 2015.

Tim Buckwalter. *Buckwalter Arabic Morphological Analyzer Version 2.0*, 2004. URL http://catalog.ldc.upenn.edu/LDC2004L02.

Joan Bybee. Regular morphology and the lexicon. *Language and Cognitive Processes*, 10: 425–455, 1995.

John J. Camilleri. A computational grammar and lexicon for Maltese. Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, September 2013.

Maris Camilleri. Clitics in Maltese. Bachelor's Thesis, University of Malta, 2009.

Burcu Can and Suresh Manandhar. Probabilistic hierarchical clustering of morphological paradigms. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 654–663. Association for Computational Linguistics, 2012.

Erwin Chan. Learning probabilistic paradigms for morphology in a latent class model. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Computational Phonology and Morphology*, SIGPHON '06, pages 69–78, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

Noam Chomsky. *Aspects of the Theory of Syntax.* MIT Press, 1965.

Noam Chomsky and Morris Halle. *The Sound Pattern of English.* New York: Harper and Row, 1968.

Morten H. Christiansen and Nick Chater. Connectionist natural language processing: The state of the art. *Cognitive Science*, 23(4):417–437, 1999.

Alexander Clark. Memory-based learning of morphology with stochastic transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 513–520, 2002.

Alexander Clark. Supervised and Unsupervised Learning of Arabic Morphology. In Abdelhadi Soudi, Antal van den Bosch, Günter Neumann, and Nancy Ide, editors, *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 181–200. Springer Netherlands, 2007.

Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning - Volume 6*, MPL '02, pages 21–30. Association for Computational Linguistics, 2002.

Mathias Creutz and Krista Lagus. Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology*, pages 43–51. Association for Computational Linguistics, 2004.

Mathias Creutz and Krista Lagus. Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0. Publications in Computer and Information Science, Report A81, Helsinki University of Technology, March 2005.

Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):1–34, 2007.

Maria Cutajar. Computational Linguistics for the Maltese Language. Bachelor's Thesis, University of Malta, 1990.

Angelo Dalli. Computational lexicon for Maltese. Master's thesis, University of Malta, 2002.

Sajib Dasgupta and Vincent Ng. High-performance, language-independent morphological segmentation. In *NAACL HLT 2007: Proceedings of the Main Conference*, pages 155–163, 2007.

Anne N. de Roeck and Waleed Al-Fares. A morphologically sensitive clustering algorithm for identifying Arabic roots. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 199–206. Association for Computational Linguistics, 2000.

Greg Durrett and John DeNero. Supervised learning of complete morphological paradigms. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pages 1185–1195, 2013. URL http://aclweb.org/anthology//N/N13/N13-1138.pdf.

Leanne Ellul. In-Nomi Verbali fil-Malti. Master's thesis, University of Malta, 2015.

Jeffrey L. Elman. Connectionist models of cognitive development: where next? *Trends in Cognitive Sciences*, 9(3):111–117, 2005.

Ray Fabri. Stem allomorphy in the Maltese verb. In *Ilsienna - Our Language*, volume 1, pages 1–20, Germany, 2009. Brockmeyer Verlag.

Ray Fabri. Maltese. In Christian Delcourt and Piet van Sterkenburg, editors, *The Languages of the New EU Member States*, volume 88, pages 791–816. Revue Belge de Philologie et d'Histoire, 2010.

Ray Fabri, Michael Gasser, Nizar Habash, George Kiraz, and Shuly Wintner. Linguistic introduction: The orthography, morphology and syntax of semitic languages. In *Natural Language Processing of Semitic Languages*, Theory and Applications of Natural Language Processing, pages 3–41. Springer Berlin Heidelberg, 2014.

Alex Farrugia. A computational analysis of the Maltese broken plural. Bachelor's Thesis, University of Malta, 2008.

George Farrugia. *Il-Ġens grammatikali fil-Malti.* PhD thesis, University of Malta, 2010.

David Galea. Morphological analysis of Maltese verbs. Bachelor's Thesis, University of Malta, 1996.

Albert Gatt and Slavomír Čéplö. Digital Corpora and Other Electronic Resources for Maltese. In *Proceedings of the Corpus Linguistic Conference*, Lancaster, UK, 2013.

John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27:153–198, June 2001.

John Goldsmith. Segmentation and morphology. In *Handbook of Computational Linguistics and Natural Language Processing*, 2010.

Sharon Goldwater and Mark Johnson. Priors in bayesian learning of phonological rules. In *Proceedings of the Seventh Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON '04)*, Barcelona, Spain, 2004.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. *Standard Arabic Morphological Analyzer Version (SAMA) 3.1*. Linguistic Data Consortium LDC2009E73, 2004. URL http://catalog.ldc.upenn.edu/LDC2004L02.

Bro. Henry Grech. *Grammatika Maltija*. De La Salle Brothers Publications, 6th edition, 1980.

Nizar Habash. Arabic morphological representations for machine translation. In *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, pages 263–285. Springer Netherlands, 2007.

Nizar Habash and Owen Rambow. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 573–580, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

Nizar Habash, Owen Rambow, and George Kiraz. MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 17–24. The Association for Computer Linguistics, 2005.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. A morphological analyzer for egyptian arabic. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 1–9. Association for Computational Linguistics, 2012.

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. Morphological analysis and disambiguation for dialectal arabic. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 426–432, 2013.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

Harald Hammarström and Lars Borin. Unsupervised learning of morphology. *Computational Linguistics*, 37:309–350, 2011.

Martin Haspelmath and Andrea D. Sims. *Understanding Morphology.* Understanding Language Series. Hodder Education, 2010.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., New York, NY, USA, second edition, 2009.

Robert D. Hoberman and Mark Aronoff. The verbal morphology of Maltese: From Semitic to Romance. In Joseph Shimron, editor, *Language processing and acquisition in languages of Semitic, root-based morphology*, pages 61–78. Benjamins, Amsterdam, 2003.

Yu Hu, Irina Matveeva, John Goldsmith, and Colin Sprague. Using morphology and syntax together in unsupervised learning. In *Proceedings of the Second Workshop on Psychocomputational Models of Human Language Acquisition*, pages 20–27. Association for Computational Linguistics, 2005.

Alon Itai and Shuly Wintner. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, 2008.

John Thayer Jensen. *Morphology: Word Structure in Generative Grammar.* J. Benjamins Publishing Company, 1990.

C. Douglas Johnson. *Formal Aspects of Phonological Description.* Mouton. The Hague, 1972.

Mark Johnson and Sharon Goldwater. Improving nonparameteric bayesian inference: Experiments on unsupervised word segmentation with adaptor grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL ’09, pages 317–325, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

David Jurgens and Keith Stevens. The S-Space Package: An Open Source Package for Word Space Models. In *System Papers of the Association of Computational Linguistics*, 2010.

Ronald M. Kaplan and Martin Kay. Phonological rules and finite-state transducers. In *Linguistic Society of America Meeting Handbook, Fifty-Sixth Annual Meeting*. New York, 1981.

Lauri Karttunen and Kenneth R. Beesley. Twenty-five years of finite-state morphology. In *Inquiries into Words, Constraints and Contexts (Festschrift in the Honour of Kimmo Koskenniemi and his 60th Birthday)*, pages 71–83. Gummerus Printing, Saarijärvi, Finland, 2005. URL http://www.helsinki.fi/esslli/evening/20years/twol-history.html.

Samarth Keshava and Emily Pitler. A simpler, intuitive approach to morpheme induction. In *PASCAL Challenge Workshop on Unsupervised Segmentation of Words into Morphemes*, pages 31–35, 2006.

George Anton Kiraz. Multitiered nonlinear morphology using multitape finite automata: A case study on syriac and arabic. *Computational Linguistics*, 26(1):77–103, 2000.

Chunyu Kit. How does lexical acquisition begin? A cognitive perspective. *Cognitive Science*, 1(1):1–50, 2003.

Kevin Knight. Connectionist ideas and algorithms. *Commun. ACM*, 33:58–74, 1990.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, SIGMORPHON '10, pages 78–86, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Kimmo Koskenniemi. *Two-level Morphology: A general computational model for word-form recognition and production.* PhD thesis, University of Helsinki, 1983.

Klaus Krippendorff. Reliability in Content Analysis: Some Common Misconceptions and Recommendations. *Human Communication Research*, 30(3):411–433, 2004.

Klaus Krippendorff. Computing Krippendorff's Alpha-Reliability, 2011. URL http://repository.upenn.edu/asc_papers/43/.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. Morpho Challenge competition 2005–2010: evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, SIGMORPHON '10, pages 87–95. Association for Computational Linguistics, 2010.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 1–9. ACL, 2011.

Gennadi Lembersky, Danny Shacham, and Shuly Wintner. Morphological disambiguation of Hebrew: A case study in classifier combination. *Natural Language Engineering*, 20(1): 69 – 97, 2014.

Ping Li, Igor Farkas, and Brian MacWhinney. Early lexical development in a self-organizing neural network. *Neural Networks*, 17:1345–1362, October 2004.

Charles X. Ling. Learning the past tense of english verbs: the symbolic pattern associator vs. connectionist models. *Journal of Artificial Intelligence Research*, 1:209–229, February 1994.

Charles X. Ling and Marin Marinov. Answering the connectionist challenge: A symbolic model of learning the past tenses of english verbs. *Cognition*, 49(3):235–290, 1993. URL papers/ling93.pdf.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, 2004.

Erwin Marsi, Antal van den Bosch, and Abdelhadi Soudi. Memory-based morphological analysis generation and part-of-speech tagging of Arabic. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, MI, 2006.

Manwel Mifsud. *Loan Verbs in Maltese: A Descriptive and Comparative Study*. New York: Brill, 1995a.

Manwel Mifsud. The productivity of Arabic in Maltese. In *Proceedings of the 2nd International Conference of AIDA*. Cambridge, 1995b.

Guido Minnen, John Carroll, and Darren Pearce. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223, 2001. ISSN 1351-3249. URL papers/minnen01.pdf.

Karthik Narasimhan, Regina Barzilay, and Tommi S. Jaakkola. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics (TACL)*, 3:157–167, 2015.

E.A. Nida. *Morphology: The Descriptive Analysis of Words*. University of Michigan publications: Linguistics. University of Michigan Press, 1949.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T. Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC2014)*, pages 1094–1101, 2014.

Manuel Perea, Albert Gatt, Carmen Moret-Tatay, and Ray Fabri. Are all Semitic languages immune to letter transpositions? The case of Maltese. *Psychonomic Bulletin and Review*, 19(5):942–947, 2012.

Steven Pinker and Alan Prince. On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28:73–193, 1988.

Ingo Plag and Harald Baayen. Suffix ordering and morphological processing. *Language*, 85:106–149, 2009.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 209–217, 2009.

Sandeep Prasada and Stephen Pinker. Generalizations of regular and irregular morphology. *Language and Cognitive Processes*, 8:1–56, 1993. URL papers/prasada_pinker1993.pdf.

Aarne Ranta. *Grammatical framework: programming with multilingual grammars*. CSLI studies in computational linguistics. CSLI Publications, Center for the Study of Language and Information, Stanford (Calif.), 2011.

Brian Roark and Richard Sproat. *Computational Approach to Morphology and Syntax*. Oxford University Press, 2007.

Mike Rosner and Jan Joachimsen. *Il-Lingwa Maltija Fl-Era Diġitali – The Maltese Language in the Digital Age*. META-NET White Paper Series. Georg Rehm and Hans Uszkoreit (Series Editors). Springer, 2012. Available online at http://www.meta-net.eu/whitepapers.

David E. Rumelhart and James L. McClelland. On learning the past tenses of english verbs. In *Parallel distributed processing: explorations in the microstructure of cognition*, pages 216–271. MIT Press, Cambridge, MA, USA, 1986.

Tamara Schembri. The Broken Plural in Maltese: An Analysis. Bachelor's Thesis, University of Malta, 2006.

Patrick Schone and Daniel Jurafsky. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7*, ConLL '00, pages 67–72. Association for Computational Linguistics, 2000.

Patrick Schone and Daniel Jurafsky. Knowledge-free induction of inflectional morphologies. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–9. Association for Computational Linguistics, 2001.

Kairit Sirts and Sharon Goldwater. Minimally-supervised morphological segmentation using adaptor grammars. *Transactions of the Association for Computational Linguistics*, 1: 255–266, 2013.

Benjamin Snyder and Regina Barzilay. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Michael Spagnol. *A tale of two morphologies. Verb structure and argument alternations in Maltese.* PhD thesis, University of Konstanz, 2011.

Andrew Spencer and Arnold M. Zwicky, editors. *The Handbook of Morphology.* Wiley-Blackwell, Malden, 2001.

Sebastian Spiegler and Christian Monson. EMMA: A novel evaluation metric for morphological analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, 2010.

David Stallard, Jacob Devlin, Michael Kayser, Yoong Keok Lee, and Regina Barzilay. Unsupervised Morphology Rivals Supervised Morphology for Arabic MT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 322–327, 2012.

Edmund F. Sutcliffe. *A grammar of the Maltese language, with chrestomathy and vocabulary.* Oxford University Press: Humphrey Milford, London, 1936.

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, 2010.

Adam Ussishkin, Colin Reimer Dawson, Andrew Wedel, and Kevin Schluter. Auditory masked priming in Maltese spoken word recognition. *Language, Cognition and Neuroscience*, 30(9):1096–1115, 2015.

Antal Van den Bosch and Walter Daelemans. Memory-based morphological analysis. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 285–292, 1999.

Antal Van den Bosch, Erwin Marsi, and Abdelhadi Soudi. Memory-based morphological analysis generation and part-of-speech tagging of Arabic. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Semitic '05, pages 1–8. Association for Computational Linguistics, 2007.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Aalto University publication series SCIENCE + TECHNOLOGY 25/2013, Aalto University, Helsinki, 2013.

Gert Westermann and Denis Mareschal. *The Cambridge Encyclopedia of Child Development*, chapter Connectionist modeling, pages 305–308. Cambridge University Press, 2005.

Shuly Wintner. Computational models of language acquisition. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 6008 of *LNCS*, pages 86–99. Springer Berlin Heidelberg, 2010.

Shuly Wintner. *Natural Language Processing of Semitic Languages*, chapter Morphological Processing of Semitic Languages. Berlin and Heidelberg: Springer, 2014.

David Yarowsky and Richard Wicentowski. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 207–216, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

Shlomo Yona and Shuly Wintner. A finite-state morphological grammar of Hebrew. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Semitic '05, pages 9–16. Association for Computational Linguistics, 2005.