Automatic standardisation of texts containing spelling variation

How much training data do you need?

Alistair Baron and Paul Rayson
Computing Department
Lancaster University
{*a.baron, paul*}*@comp.lancs.ac.uk*

Abstract

Large quantities of spelling variation in corpora, such as that found in Early Modern English, can cause significant problems for corpus linguistic tools and methods. Having texts with standardised spelling is key to making such tools and methods accurate and meaningful in their analysis. Gaining access to such versions of texts can be problematic however, and manual standardisation of the texts is often too time-consuming to be feasible. Our solution is a piece of software named VARD 2 which can be used to manually and automatically standardise spelling variation in individual texts, or corpora of any size. This paper evaluates VARD 2's performance on a corpus of Early Modern English letters and a corpus of children's written English. The software's ability to learn from manual standardisation is put under particular scrutiny as we examine what effect different levels of training have on its performance.

1 Introduction

Spelling variation within corpora has a considerable effect on corpus linguistic techniques; this has been shown particularly for Early Modern English, the latest period of the English language to contain large amounts of spelling variation. The accuracy of key word analysis (Baron et al., 2009b), part-of-speech annotation (Rayson et al., 2007) and semantic analysis (Archer et al., 2003) have all been shown to be adversely affected by spelling variation.

Some researchers have avoided the issue in Early Modern English texts by using modernised versions; for example, Culpeper (2002) opted to use a modern edition of Shakespeare's Romeo and Juliet in his study to avoid spelling variation affecting his statistical results. However, modern editions are not always readily available for historical texts and the value of modernised versions has been questioned, especially for Shakespeare's work (see Grazia and Stallybrass, 1993). Another potential solution is to manually standardise the spelling variation within these texts, this may be possible for small amounts of data, but for large corpora such as the Lampeter corpus (c. 1.1 million words) (Schmied, 1994) this soon becomes restrictively time-consuming. For the very large historical corpora now being made available through digitisation initiatives, such as Early English Books Online, a fully manual approach is clearly unworkable.

The solution we offer is a piece of software named VARD 2. The tool can be used to manually and automatically standardise spelling variation in individual texts, or corpora of any size. An important feature of VARD 2 is that through manual standardisation of corpus samples by the user, the tool 'learns' how best to standardise the spelling variation in the particular corpus being processed. This results in VARD 2 being better equipped to automatically standardise the remainder of the corpus. After automatic processing, corpus linguistic techniques can be used with greater accuracy on the standardised version of the corpus, and this avoids the need for difficult and time-consuming manual standardisation of the full text.

This paper evaluates the performance of VARD 2's automatic standardisation in terms of precision and recall. The tool's learning capability is analysed in detail with performance of automatic standardisation measured after increasing levels of training. We wish to discover the optimum amount of training a user should complete, that is the point at which further training produces no substantial improvement in recall or precision. As VARD 2 was designed to deal with Early Modern English spelling variation, the main evaluation will focus on the standardisation of an Early Modern English corpus, in this case the Innsbruck Letters Corpus, part of the Innsbruck Computer-Archive of Machine-Readable English Texts (ICAMET) corpus (Markus, 1999). The corpus is particularly useful for this evaluation as it has been standardised and manually checked with parallel lines of original and standardised text.

Whilst VARD 2 was built initially to deal with spelling variation in Early Modern English, it is not restricted to this particular variety of English, or indeed the English language itself. There are many language varieties containing spelling variation and VARD 2 has the potential to deal any variety of spelling variation, in any language, although customisation may be necessary and training will be required for the tool to 'learn' how to best automatically standardise a particular corpus. The final evaluation in this paper investigates VARD 2's capability in dealing with one of these language varieties with the attempted automatic standardisation of a corpus of children's written English.

The remainder of this paper will first provide a background to the causes of spelling variation, particularly in Early Modern English, and the problems which it creates in corpus linguistics (section 2). VARD 2 will then be introduced and described in detail in section 3. The main evaluation and discussion follows in section 4, where VARD 2's performance is tested on an Early Modern English corpus. Various improvements have been made to VARD 2 which will be incorporated into the next release, these developments are also described and the effects evaluated. Section 5 evaluates VARD 2's capability when automatically standardising a child language corpus. Finally our findings will be summarised and we will discuss plans for future developments and research in section 6.

2 Spelling Variation

Spelling variation is a feature in many corpora, particularly in historical corpora such as from the Early Modern English period but also in modern language varieties such as web-based texts. Whilst this orthographical variation is often of linguistic importance, abnormal spellings can have a detrimental effect on the accuracy of automatic corpus linguistic techniques.

Early Modern English (c. 1450-1700) is of particular linguistic research interest as it is the earliest period of the English language from which a reasonably large corpus can be constructed. This was largely due to a sharp increase in book production through the introduction of the printing press by William Caxton in 1476 and an increasingly literate public (Görlach, 1991: 6). Many historical English corpora have been created containing texts from the Early Modern English period; these include the Helsinki, ARCHER and ZEN corpora (described in Kytö et al., 1994), the Corpus of Early English Correspondence (CEEC) (Nevalainen, 1997), the Corpus of English Dialogues (CED) (Culpeper and Kytö, 1997), the Early Modern English Medical Texts (EMEMT) corpus (Taavitsainen and Pahta, 1997) and the Innsbruck Computer-Archive of Machine-Readable English Texts (ICAMET) corpus (Markus, 1999). Additionally, increasing amounts of textual data from the period are being digitised through initiatives including the Open Content Alliance[1], Google Books[2] and Early English Books Online[3].

The English language was under significant change throughout the Early Modern English period, one reason being that Latin and French were rapidly being replaced by English as the preferred choice for print and speech for many institutions and individuals (see Singh, 2005: 140-147), this especially due to King Henry V's commitment to the vernacular in his official correspondence in 1417 (Richardson, 1980: 727). The language "lacked obligatory rules of spelling, pronunciation, morphology and syntax" (Görlach, 1991: 36) and authors, scribes, editors and printing houses had their individual spelling preferences. Generally, there was no notion of the importance for a single spelling for each word; letters would be added or removed, for example, to ease line justification (Vallins and Scragg, 1965: 71), it was even common to find a word spelt numerous ways in the same text or even on the same page. Some common spelling variant examples are shown in Table 1. The amount of spelling variation declined over time, as described by Görlach (1991: 8-9), Lass (1999: 56) and Rissanen (1999: 187) and by the 18th century printers were using a single spelling for most words, and the modern spelling system slowly became fixed (Vallins and Scragg, 1965: 71). We have recently quantified this trend, showing a gradual decline in the levels of spelling variation until a levelling off by 1700 (Baron et al., 2009b).

Variant	**Modern Equivalent**	**Notes**
"goodnesse"	"goodness"	'e' often added to end of words.
"brush'd"	"brushed"	Apostrophes often used instead of 'e'.
"encrease"	"increase"	Vowels commonly interchanged.
"spels"	"spells"	Consonants often doubled or singles.
"deliuering"	"delivering"	Common for 'u' and 'v' to be interchangeable.
"conuay'd"	"conveyed"	Many combinations of the above.

Table 1: Examples of spelling variants commonly found in Early Modern English Texts

Various pieces of software have been developed (e.g. WMatrix (Rayson, 2009) and Wordsmith Tools (Scott, 2004)) to help perform common corpus linguistic techniques, such as word frequency profiles, key word analysis, concordancing, collocations and annotation. However, these tools and the methods they use are designed to deal with modern English, problems occur when large levels of spelling variation exist - as in Early Modern English. This is mostly due to different spellings of the same word creating inaccuracies in word counts or the reliance on a modern lexicon of words. We have previously highlighted the problem when analysing Early Modern English in the cases of key word analysis (Baron et al., 2009b), part-of-speech annotation (Rayson et al., 2007) and semantic analysis (Archer et al., 2003).

Of course, even in present-day English spelling variation exists, although such problems are much less frequent, fairly well defined and easier to deal with than historical spelling variation. Sebba (2007) discusses society's view and the social reasoning and implications of the varying orthographies found in modern language, whilst Vallins and Scragg (1965: 150-183) dedicates a chapter to the subject of continued spelling variation, pointing out common discrepancies between printers, authors and even dictionaries, for example:

- *-ise* and *-ize* being interchangeable, e.g. *criticise* / *criticize*.

- Mute *e* before suffix being optional, e.g. *judgement* / *judgment*.
- *ct* and *x* being interchangeable, e.g. *inflection* / *inflexion*.
- The joining of words with or without hyphens, e.g. *ice-cream* / *ice cream*.

Spelling variation is more prevalent in other varieties of English, such as child and non-native language, SMS messaging and many web-based communications such as weblogs, emails, chatrooms and bulletin boards. The characteristics and levels of spelling variation changes from corpus to corpus but the spelling variation in these language varieties is likely to present a similar barrier to corpus linguistics techniques as that found to be the case for Early Modern English.

3 VARD 2

In order to address the problem that spelling variation presents to automatic corpus linguistic techniques we have developed a piece of software which acts as a pre-processor for corpus linguistic tools. The initial target of the software was to process historical texts, particularly from the Early Modern English period, augmenting texts with modern equivalents for any variants found within. After further development the software can now be customised and trained to deal with a wide range of spelling variation, not necessarily limited to the English language.

The original tool, named VARD (VARiant Detector) (see Rayson et al., 2005) relied solely on a large (c. 45,000 entries) manually created list of variants mapped to their modern equivalents, this was used to search for and replace any spelling variants found within a text. This technique successfully deals with a substantial amount of spelling variation in Early Modern English but is limited in its performance as it is impossible to include all viable spelling variants in a pre-defined list due to the nature of Early Modern English producing an endless variety of potential forms. Additionally, as the variant list was created to deal with only Early Modern English it is of little use when dealing with the spelling variation found in other varieties of English and in other languages. The tool also permitted little user control over whether a variant was replaced; if a word in the text was listed as a variant it would always have the modern equivalent inserted alongside the word in the text. Whilst this may be desirable in most cases, the user may wish, for example, to have some variants remain in their original form in certain contexts.

VARD 2[4] (Rayson et al., 2008; Baron and Rayson, 2008) was developed to allow a more flexible approach to dealing with spelling variation. As well as utilising the manually created list of variants and modern equivalents used in the original VARD tool, the latest released version, VARD 2.2, employs techniques from modern spell checking software to search for potential variants and find candidate equivalents for variants found. Given a text to process by the user, the tool begins by comparing each word found in the text to a modern lexicon derived from the British National Corpus (Leech et al., 2001) and the Spell Checking Oriented Word List (SCOWL)[5], if a word is not found in the modern lexicon it is marked as a potential variant. For each detected variant a list of candidate equivalents is produced, this list is ranked by a confidence score given to each candidate based on the blend of methods used to find that candidate. Four methods are used to search and rank candidate replacements: the manually created list of variant and modern equivalents, a phonetic matching algorithm, a set of letter replacement rules and an edit distance algorithm.

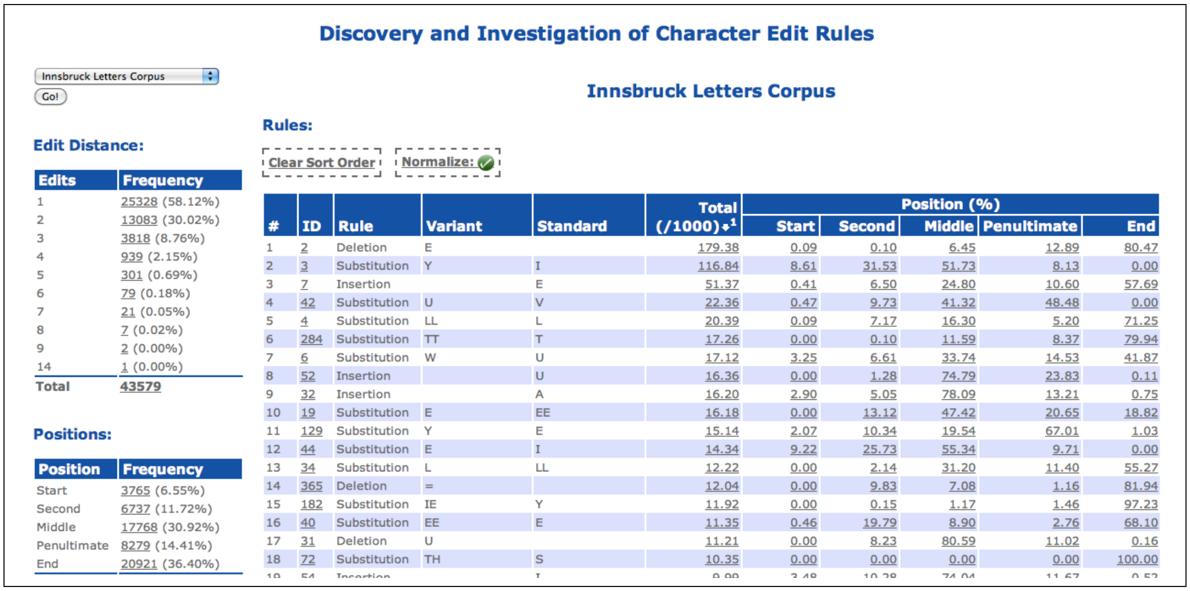
Phonetic matching techniques have been used for decades to identify strings that have similar sounds when spoken, and are commonly used when searching for a name in a database (Zobel and Dart, 1996; Pfeifer et al., 1996). The most familiar phonetic matching algorithm is Soundex, patented by Robert C. Russell and Margaret O'Dell in 1918. The algorithm used in VARD 2.2 is adapted from Soundex and is used to assign each word in the modern lexicon with a phonetic code based on the letters constituting that word. For a given variant, a phonetic code is produced and any words in the modern lexicon with an identical phonetic code are offered as potential equivalents. Soundex (and other phonetic matching algorithms) are typically of high recall but low precision (Hodge and Austin, 2001), this generally applies to the algorithm used in VARD 2.2 resulting in the algorithm often finding the correct equivalent but also finding many false positives which need to be filtered out by other means.

In many sources of spelling variation there exists common letter differences, or groups of letters which are interchangeable. In typing, for instance, the proximity of letters on the keyboard leads to common mistakes, e.g. 'and' mistyped 'anf' and 'are' mistyped 'arte' (Yannakoudakis and Fawthrop, 1983; Mitton, 1996: 77-92). Based on knowledge of the sources of spelling variation, a series of letter replacement rules can be coded which in turn can be used to transform a given spelling variant into alternative forms. VARD 2.2 employs a user-defined list of letter replacement rules to compute these alternative forms, any forms which match words in the modern lexicon are offered as candidate equivalents. The tool comes with a manually derived list of rules for Early Modern English, examples of which include:

- Replace final *ck* with *c*
- Replace *u* with *v*
- Replace *v* with *u*
- Replace final *'d* with *ed*
- Remove final *e*

These rules are presented in more detail by Archer et al. (2006). We have developed a complementary tool to VARD 2 named DICER (Discovery and Investigation of Character Edit Rules). DICER can process a list of variant and equivalent mappings to compute a set of letter replacement rules for each mapping which can transform the variant form into its equivalent. The details of these letter replacement rules are then collated into a database which can be viewed through a set of web pages[6], a sample of the main table produced is shown in Figure 1. The analysis produced by DICER can be used to build a new list of letter replacement rules for VARD 2 or to supplement an existing list. We have shown elsewhere (Baron et al., 2009a) that producing a rule set based on manually standardised samples of a corpus with DICER and introducing this to VARD 2 results in a significant increase in performance when automatically standardising the remainder of the corpus.

The phonetic matching and letter replacement techniques described above are used in conjunction with the manually derived list of variants and modern equivalents to produce a list of candidate replacements. One further technique, an edit distance algorithm, is used on these candidates to add supplementary evidence when calculating a confidence score to rank upon. In VARD 2.2, Levenshtein Distance (Levenshtein 1966, cited by Kukich, 1992) is used to calculate the number of letter



Discovery and Investigation of Character Edit Rules

Innsbruck Letters Corpus

Edit Distance:

Edits	Frequency
1	25328 (58.12%)
2	13083 (30.02%)
3	3818 (8.76%)
4	939 (2.15%)
5	301 (0.69%)
6	79 (0.18%)
7	21 (0.05%)
8	7 (0.02%)
9	2 (0.00%)
14	1 (0.00%)
Total	43579

Positions:

Position	Frequency
Start	3765 (6.55%)
Second	6737 (11.72%)
Middle	17768 (30.92%)
Penultimate	8279 (14.41%)
End	20921 (36.40%)

Rules:

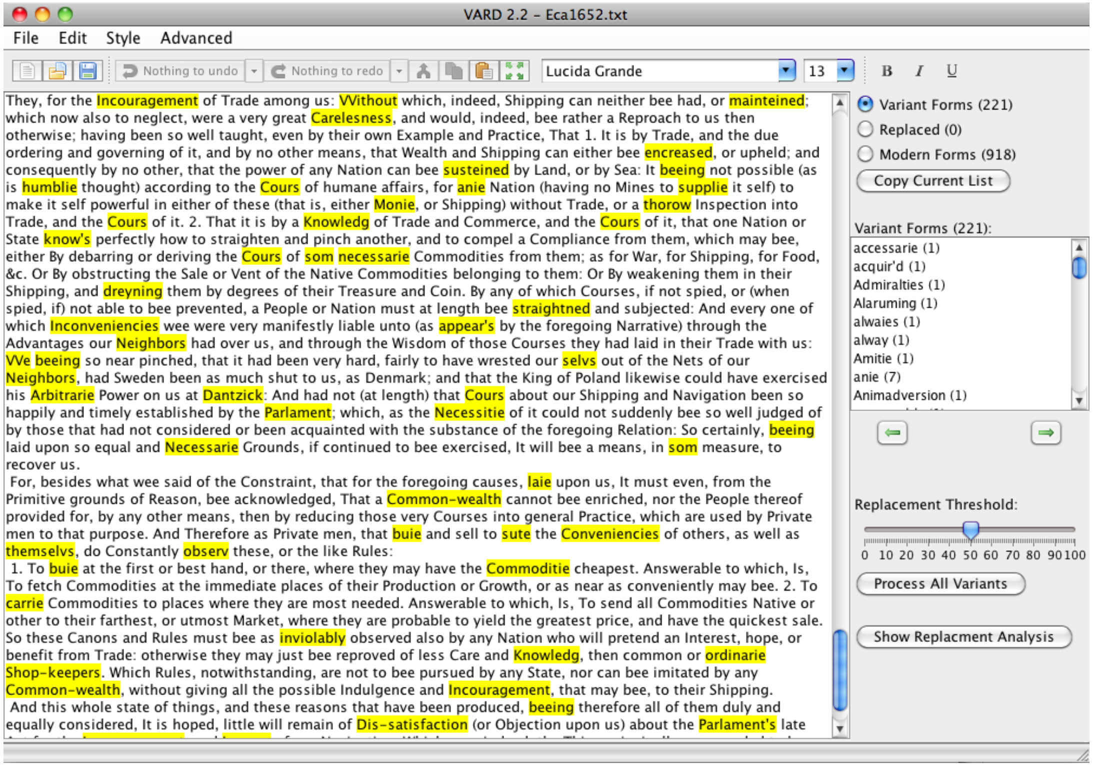
					Total	Position (%)				
#	ID	Rule	Variant	Standard	(/1000)↓1	Start	Second	Middle	Penultimate	End
1	2	Deletion	E		179.38	0.09	0.10	6.45	12.89	80.47
2	3	Substitution	Y	I	116.84	8.61	31.53	51.73	8.13	0.00
3	7	Insertion		E	51.37	0.41	6.50	24.80	10.60	57.69
4	42	Substitution	U	V	22.36	0.47	9.73	41.32	48.48	0.00
5	4	Substitution	LL	L	20.39	0.09	7.17	16.30	5.20	71.25
6	284	Substitution	TT	T	17.26	0.00	0.10	11.59	8.37	79.94
7	6	Substitution	W	U	17.12	3.25	6.61	33.74	14.53	41.87
8	52	Insertion		U	16.36	0.00	1.28	74.79	23.83	0.11
9	32	Insertion		A	16.20	2.90	5.05	78.09	13.21	0.75
10	19	Substitution	E	EE	16.18	0.00	13.12	47.42	20.65	18.82
11	129	Substitution	Y	E	15.14	2.07	10.34	19.54	67.01	1.03
12	44	Substitution	E	I	14.34	9.22	25.73	55.34	9.71	0.00
13	34	Substitution	L	LL	12.22	0.00	2.14	31.20	11.40	55.27
14	365	Deletion	=		12.04	0.00	9.83	7.08	1.16	81.94
15	182	Substitution	IE	Y	11.92	0.00	0.15	1.17	1.46	97.23
16	40	Substitution	EE	E	11.35	0.46	19.79	8.90	2.76	68.10
17	31	Deletion	U		11.21	0.00	8.23	80.59	11.02	0.16
18	72	Substitution	TH	S	10.35	0.00	0.00	0.00	0.00	100.00
19	54	Insertion		I	9.99	3.48	10.28	74.04	11.67	0.52

Figure 1: Screenshot of DICER analysis main table.

edits (insertions, deletions or substitutions) required to transform the variant to its equivalent. Due to Levenshtein Distance being computationally expensive, it is not feasible to calculate the score for a given variant against each word in the modern lexicon, this is why only the words in the subset found through other methods have scores calculated against them.

The confidence score for each candidate replacement is calculated by adding a dynamically weighted score for each method which successfully returned that candidate, these weighted scores always sum to 100%. The edit distance score multiplied by 2% is then subtracted giving a final confidence score as a percentage[7]. The candidates are then ranked upon this score, if two scores are equal the word with the superior frequency (in the BNC or according to the SCOWL word-list) is ranked higher. The weights attached to each method will update each time a method is successful over another method in finding the user-chosen equivalent. This results in the tool 'learning' which methods are more appropriate for the text(s) being processed and thus give higher confidence scores to those candidates found with methods that have been more successful in the past. This capability makes the tool much more flexible when dealing with different varieties of text; training the tool for a particular corpus by processing sample texts first will allow the tool to better find and rank candidate equivalents for variants found in the remainder of the corpus.

VARD 2 can be used in two ways: to manually standardise texts or to automatically standardise a set of texts or corpora. In order to manually standardise texts an interactive mode is available, the main user interface of which is shown in Figure 2. Here a user can view the entire text and have variants highlighted and listed alphabetically, the user can also view variants which have already been replaced and words which were found in the modern lexicon. The user can right-click on a variant in the text or list and be presented with the ranked list of candidate equivalents, this is shown in Figure 3. As can be seen, the tool displays details of how it arrived at a candidate score by indicating which methods (and at what weights) were used to find and score the candidate. The user can choose one of the candidates presented, replacing a particular instance or replacing all occurrences, or if the correct equivalent is not available the user can manually input their own equivalent. Once a candidate has been chosen the variant is replaced in the text by the equivalent, however the original form is always retained for future output or if the user decides to revert a replacement operation.



They, for the Incouragement of Trade among us: VVithout which, indeed, Shipping can neither bee had, or mainteined; which now also to neglect, were a very great Carelesness, and would, indeed, bee rather a Reproach to us then otherwise; having been so well taught, even by their own Example and Practice, That 1. It is by Trade, and the due ordering and governing of it, and by no other means, that Wealth and Shipping can either bee encreased, or upheld; and consequently by no other, that the power of any Nation can bee susteined by Land, or by Sea: It beeing not possible (as is humblie thought) according to the Cours of humane affairs, for anie Nation (having no Mines to supplie it self) to make it self powerful in either of these (that is, either Monie, or Shipping) without Trade, or a thorow Inspection into Trade, and the Cours of it. 2. That it is by a Knowledg of Trade and Commerce, and the Cours of it, that one Nation or State know's perfectly how to straighten and pinch another, and to compel a Compliance from them, which may bee, either By debarring or deriving the Cours of som necessarie Commodities from them; as for War, for Shipping, for Food, &c. Or By obstructing the Sale or Vent of the Native Commodities belonging to them: Or By weakening them in their Shipping, and dreyning them by degrees of their Treasure and Coin. By any of which Courses, if not spied, or (when spied, if) not able to bee prevented, a People or Nation must at length bee straightned and subjected: And every one of which Inconveniencies wee were very manifestly liable unto (as appear's by the foregoing Narrative) through the Advantages our Neighbors had over us, and through the Wisdom of those Courses they had laid in their Trade with us: VVe beeing so near pinched, that it had been very hard, fairly to have wrested our selvs out of the Nets of our Neighbors, had Sweden been as much shut to us, as Denmark; and that the King of Poland likewise could have exercised his Arbitrarie Power on us at Dantzick: And had not (at length) that Cours about our Shipping and Navigation been so happily and timely established by the Parlament; which, as the Necessitie of it could not suddenly bee so well judged of by those that had not considered or been acquainted with the substance of the foregoing Relation: So certainly, beeing laid upon so equal and Necessarie Grounds, if continued to bee exercised, It will bee a means, in som measure, to recover us.

For, besides what wee said of the Constraint, that for the foregoing causes, laie upon us, It must even, from the Primitive grounds of Reason, bee acknowledged, That a Common-wealth cannot bee enriched, nor the People thereof provided for, by any other means, then by reducing those very Courses into general Practice, which are used by Private men to that purpose. And Therefore as Private men, that buie and sell to sute the Conveniencies of others, as well as themselvs, do Constantly observ these, or the like Rules:

1. To buie at the first or best hand, or there, where they may have the Commoditie cheapest. Answerable to which, Is, To fetch Commodities at the immediate places of their Production or Growth, or as near as conveniently may bee. 2. To carrie Commodities to places where they are most needed. Answerable to which, Is, To send all Commodities Native or other to their farthest, or utmost Market, where they are probable to yield the greatest price, and have the quickest sale. So these Canons and Rules must bee as inviolably observed also by any Nation who will pretend an Interest, hope, or benefit from Trade: otherwise they may just bee reproved of less Care and Knowledg, then common or ordinarie Shop-keepers. Which Rules, notwithstanding, are not to bee pursued by any State, nor can bee imitated by any Common-wealth, without giving all the possible Indulgence and Incouragement, that may bee, to their Shipping.

And this whole state of things, and these reasons that have been produced, beeing therefore all of them duly and equally considered, It is hoped, little will remain of Dis-satisfaction (or Objection upon us) about the Parlament's late

● Variant Forms (221)
○ Replaced (0)
○ Modern Forms (918)

Variant Forms (221):

accessarie (1)
acquir'd (1)
Admiralties (1)
Alaruming (1)
alwaies (1)
alway (1)
Amitie (1)
anie (7)
Animadversion (1)

Replacement Threshold: 50

Figure 2: Screenshot of VARD 2.2 interactive interface.

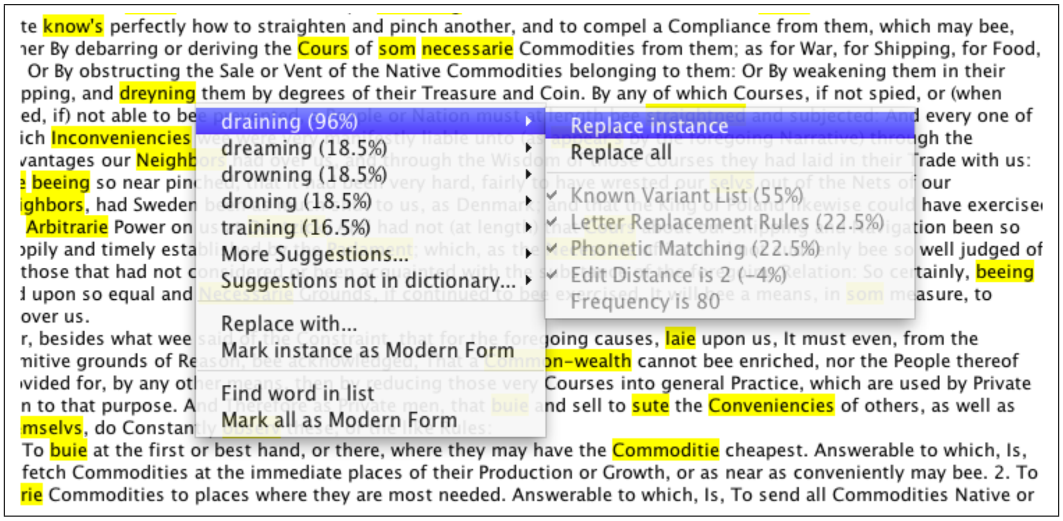
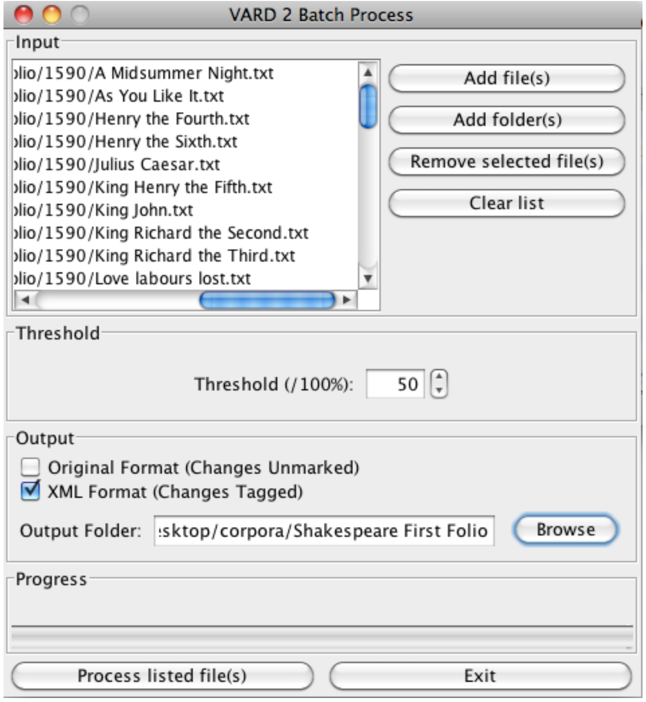


Figure 3: Screenshot of VARD 2.2 variant popup menu.

Other options available in the interactive version include the ability to mark words as variants if they have been mistakenly marked as modern forms (for example, real word errors such as ‘bee’ for ‘be’ can be problematic when contextual information is required to determine if the word is a variant.). The user can mark words as modern forms if they have been mistakenly marked as variants (for example, infrequent proper names are unlikely to be present in the modern lexicon and will be marked as variants), and there is the ability to join words separated by white-space (e.g. line breaks), undo and redo any edit made, search through occurrences of a particular word, add/edit/remove letter replacement rules and many more.

As well as manually dealing with the spelling variation, the user can choose to allow the tool to automatically standardise all variant forms with their highest ranked candidate. In order to retain some control over this procedure a confidence threshold can be provided, this is the minimum score that the top candidate must have for a replacement to be made. This function is available in the interactive mode of the tool to automatically standardise a single text, but also a batch processing mode exists which allows the user to choose a set of texts to be processed in turn. A graphical user interface to the batch processing is provided, shown in Figure 4, or a command-line version is also available.



›lio/1590/A Midsummer Night.txt
›lio/1590/As You Like It.txt
›lio/1590/Henry the Fourth.txt
›lio/1590/Henry the Sixth.txt
›lio/1590/Julius Caesar.txt
›lio/1590/King Henry the Fifth.txt
›lio/1590/King John.txt
›lio/1590/King Richard the Second.txt
›lio/1590/King Richard the Third.txt
›lio/1590/Love labours lost.txt

Threshold (/100%): 50

□ Original Format (Changes Unmarked)
☑ XML Format (Changes Tagged)

Output Folder: :sktop/corpora/Shakespeare First Folio

Figure 4: Screenshot of VARD 2.2 batch processing interface.

Once texts have been standardised with VARD 2 they can be saved with equivalents appearing in place of variants. The original variant forms are always maintained in the form of an XML tag, for example:

```
<replaced orig="companie">company</replaced>
```

The majority of automated corpus linguistic tools analysing the standardised text will only ‘see’ the new equivalent form and effectively ignore the original spelling. Because the orthography analysed will be closer to standard Modern English, texts will be easier to process and accuracy will be improved. A reader of the texts will always have access to the original spelling forms and it is a simple scripting process to switch between the original and equivalent forms.

Users can customise VARD 2 to deal with a particular corpus, text-type or even languages other than English. This can be achieved through training the tool on samples of their corpus in the interactive mode; this results in the software 'learning' which of its methods are most appropriate, having variant equivalents it has not seen before saved for future use and having its modern lexicon edited: i.e. words which are commonly variants can be removed and 'correct' spellings of words can be added. Furthermore, users can 'plug in' their own modern lexicon (i.e. for another language), rules list (from DICER), variant equivalents list and formatting structure (e.g. if text in certain structures should be ignored, such as headings) to customise VARD 2 for their corpus.

4 Standardising Early Modern English Texts

As VARD 2 was initially developed for use with texts from the Early Modern English period, we first wished to evaluate its performance with a corpus from this time period. We were particularly interested in how effectively VARD 2 deals with spelling variation and what effect different levels of training have on the tool. To measure VARD 2's effectiveness, recall and precision statistics can be calculated; recall being the proportion of variants that have been successfully standardised and precision being the proportion of standardisations made that are correct.

4.1 Data Used for Training and Evaluation

For an effective evaluation of VARD 2 a reasonably large corpus is required which has been manually standardised; the manual standardisations can be used in two ways: to train the tool so it learns which of its standardisation methods are most appropriate for the corpus, and to compare against automatic standardisations made to produce recall and precision statistics. Fortunately, we gained access to such a corpus in the form of the Innsbruck Letters Corpus, which is part of the ICAMET corpus (Markus, 1999). The corpus contains 469 complete personal letters dated between 1386 and 1688, totalling 182,000 words. As well as being a valuable resource to corpus linguistic studies of the Early Modern English period, the corpus is of particular use to us as it contains parallel line pairs where the first line is the original text and the second line is a manually-checked standardised version, for example:

```
$I schepyng at thys day, but be the grace of God I am avysyd for
$N shipping at this day, but by the grace of God I am advised for
```

A processing tool was developed to process these parallel lines and produce a version of the corpus similar to VARD 2's format with in-line xml tags, so the example above becomes:

```
<replaced orig="schepyng">shipping</replaced> at <replaced orig=
"thys">this</replaced> day, but <replaced orig="be">by</replaced>
the grace of God I am <replaced orig="avysyd">advised</replaced> for
```

This allows the corpus to be used as if it was standardised using VARD 2. To fairly train and test the software, the entire corpus was split into equally sized samples (of 100 and 1000 tokens) containing randomly selected short sequences from a variety of randomly selected letters. This process prevented any bias from letters containing more spelling variation due to the date of writing (levels of spelling variation reduced over the Early Modern English Period, as shown by Baron et al. (2009b).), the writer or any other factor. The set of samples was then split into two halves, the first half would be the base for training the tool, whilst the second half would be used to test how close VARD 2's automatic standardisation came to the manual standardisation after different levels of training.

4.2 Initial Results with VARD 2.2

The evaluation presented throughout this paper was completed using a complimentary tool developed especially for evaluation purposes. The tool uses the VARD 2 software library to process texts but instead of producing standardised texts it outputs a statistical analysis of various elements of VARD 2's performance. These statistics are outputted textually and in the form of graphs, in this paper we will present the graphs outputted at different stages of the evaluation and discuss the performance of VARD 2. In each test VARD 2 begins with no previous training and is set up with the default data (developed for Early Modern English) which comes as standard with the tool.

The latest publicly released version of the software, VARD 2.2, was evaluated first, this version is described in detail in the previous section. The first test was run with 1,000 word samples from the training data; VARD 2's performance was tested after no training data whatsoever and then after each 1,000 word sample. The processing of each sample simulates a user manually processing the sample in the interactive mode of the tool with VARD 2 'learning' from decisions made. A confidence threshold of 50% was used, meaning a candidate replacement must have a confidence score of at least 50% for it to be used (see page 8).

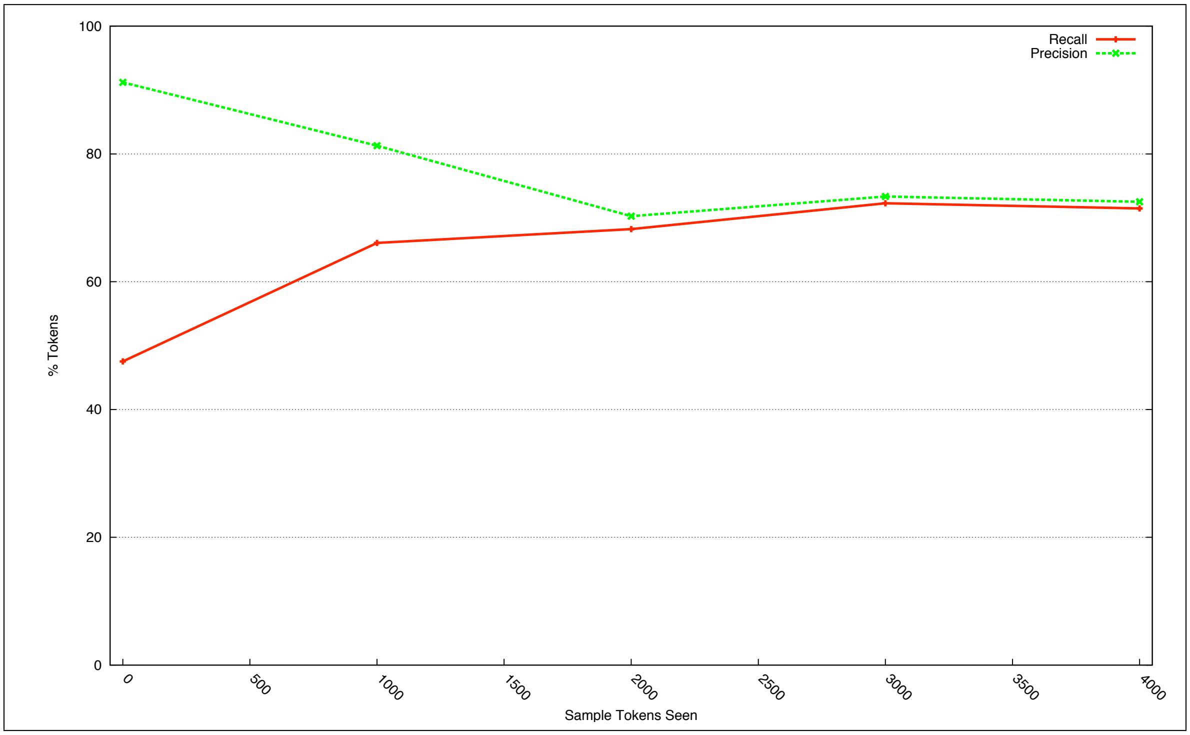


Figure 5: Initial precision and recall scores with VARD 2.2.

On running the initial evaluation, shown in Figure 5, it soon became apparent that the training methods were not performing as well as hoped. If only looking at the recall statistic, the performance looks fairly promising with an initial recall percentage of nearly 50% quickly increasing to 65% and then 70%, this shows that the training method is increasing the number of variants dealt with. Unfortunately, this increase in recall comes at a substantial cost of precision, which is initially fairly high at over 90%, but after training actually reduces to just over 70%. This means that although variants are being standardised more frequently, less variants are being standardised correctly. In many applications of natural language processing a trade-off between recall and precision is acceptable and often expected; in the case of spelling standardisation precision is generally of much greater concern than recall, incorrect standardisations are of little use and will more than likely be counterproductive, having a negative effect on the accuracy of tools processing the standardising text. Ideally, we wish for VARD 2 to have a high precision score which is maintained after training and improved recall.

So why is this reduction in precision occurring? VARD 2's training method works well for small amounts of text, it will quickly learn which of its methods are most applicable to the text and favour these methods when suggesting replacements. However, with larger levels of training the 'best' method generally overpowers the other methods, meaning that if a replacement is not found with that method its score will never reach the threshold required for replacement. In the case of the Innsbruck Letters corpus, the phonetic matching algorithm is of greater effectiveness (in terms of recall) over the other methods, so with increased training it develops a high weighting. Because phonetic matching is typically high recall with low precision (Hodge and Austin, 2001) and VARD 2 is relying more and more upon this method, the tool's overall precision drops at the expense of recall.

Clearly, improvements were needed in VARD 2's training technique. In VARD 2.2, training only produced improvements in terms of recall, it is important that both precision and recall are taken into account when assigning weights to standardisation methods. Another problem is that one method could too easily overpower other methods; instead of adjusting method weights against each other, methods need to have weights individually adjusted, independent of any other methods.

4.3 Training Technique Improvements

To improve the performance of VARD 2's training capability, the training technique was overhauled for the next version of VARD 2. In order to take into account precision as well as recall, each method will now have an average precision and recall score attached to it. Each time a replacement is made, these precision and recall scores will be recalculated; for example, if a method is consistently suggesting the chosen replacement and with few alternative candidates the method will have a high recall and high precision, if a method is consistently suggesting the chosen replacement but with a lot of alternative candidates the method will have high recall but low precision, finally if a method is consistently suggesting many candidates and the chosen replacement is rarely in this set of candidates the method will have low recall and low precision. Because each method has its average recall and precision calculated independently of other methods the high performance of one method will not have a detrimental effect on the calculated reliability of other methods.

This intermediate version of VARD 2 will find candidate replacements for variants using all methods as in VARD 2.2, however, the confidence score for each candidate will now be based on a predicted recall and precision. This will be calculated by combining scores for each method offering the candidate; the past-evidence recall average, the past-evidence precision average and the precision score for the current candidate (i.e. of how many alternatives is the candidate offered by the method) are all taken into account. These predicted recall and precision scores will then be combined into an F-measure (van Rijsbergen, 1979), calculated as shown below, and formed into a percentage which can be compared to the replacement threshold (see page 8) in the same way as in VARD 2.2.

$$F_\beta = \frac{(1+\beta^2)\cdot(precision \cdot recall)}{(\beta^2 \cdot precision + recall)} \qquad (F_\beta \text{ Measure})$$

With β, a non-negative real-number, dictating the balance between precision and recall; for example, $F_{\frac{1}{2}}$ weights precision twice as much as recall whilst F_2 weights recall twice as much as precision. F_1, shown below, is used to balance recall and precision equally, this is used in the intermediate version of VARD 2 being evaluated.

$$F = \frac{2 \cdot precision \cdot recall}{precison + recall} \qquad (\text{F}_1\text{-Measure})$$

The effect of these changes to VARD 2's accuracy and training ability are shown in Figure 6. The results are much more promising than the initial results (Figure 5); whilst the recall score is lower (40% rising to 50% through training), precision is slightly improved (92–93%) but more importantly is maintained despite the increase in recall. This shows that the changes made to the training technique have been beneficial.

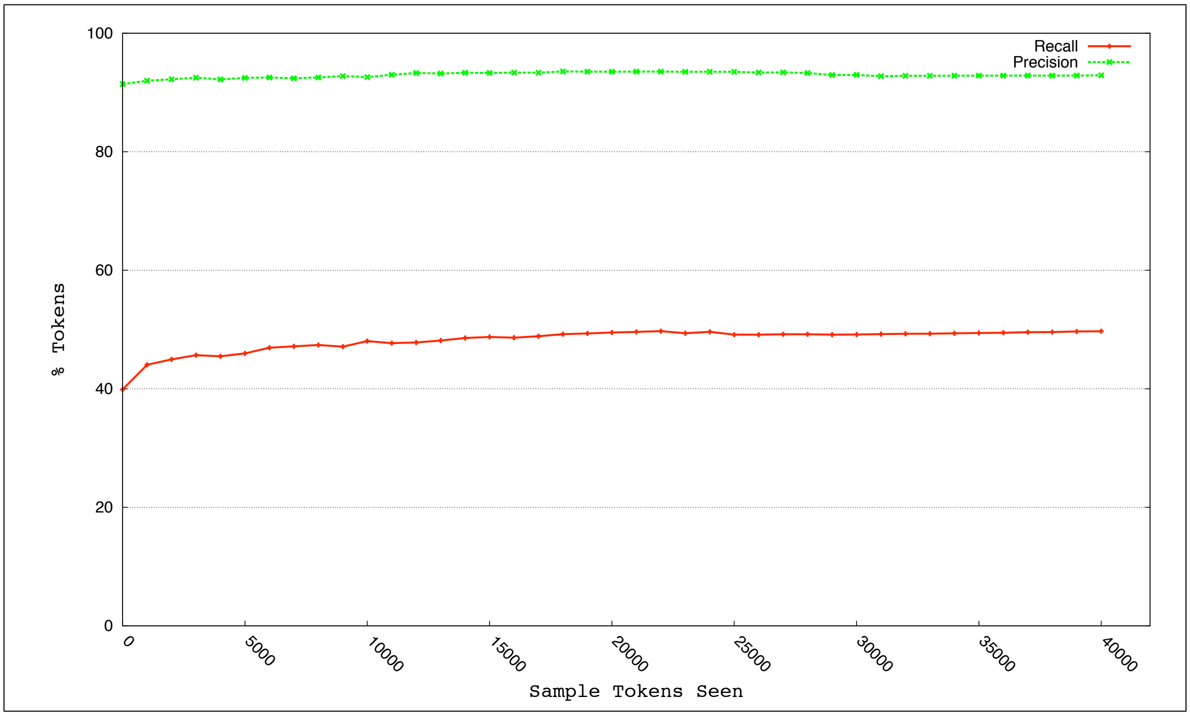


Figure 6: Precision and recall scores of VARD 2 after improvement of training technique.

4.4 VARD 2 Method Improvements

Whilst the results shown in Figure 6 were promising, it was felt that further advances could be made. Development of the next version of the software, VARD 2.3, has seen improvements made to the individual methods that VARD uses to find candidate replacements for variants. For the final results of our evaluation VARD 2 will be tested with these improvements in place.

In VARD 2.2, edit distance was used differently than the other methods to contribute to the confidence score for each candidate (see page 5). As confidence scores are now calculated in terms of predicted precision and recall it no longer seems logical to use Edit Distance in this way. Instead, it would be desirable for edit distance to contribute to the confidence score in the same way as the other three methods. The Levenshtein Distance (Levenshtein 1966, cited by Kukich, 1992) used currently returns an integer value, i.e. the number of edits required to transform the variant string into the candidate string. This is problematic for two reasons: firstly the length of the two strings is not taken into account, common sense dictates that an edit distance cost of 2 between two strings of length 4 has more impact than a cost of 2 on two strings of length 10; secondly, in order to calculate precision and recall scores in a comparable manner to the other methods, a score between 0 and 1 is required. To solve these problems an edit distance simply normalised by the length of the two strings being compared was used, this returns a similarity score between 0 and 1; 0 meaning the two strings bear no similarity and 1 meaning the strings are exactly the same. The formula, shown below, was considered by (Sampson and Babarczy, 2003) in their study of parsing accuracy. Levenshtein Distance continues to be used to calculate edit distance.

$$similarity = 1 - \frac{distance(x, y)}{length(x) + length(y)} \qquad \text{(Normalised Edit Distance)}$$

As well as taking into account the length of the strings, this also allows precision and recall averages to be calculated based upon chosen replacements, and to add evidence when calculating the predicted recall and precision for new candidates. The similarity score can be regarded as a confidence score for edit distance, the recall is therefore just the similarity score between the variant and the candidate (a true positive)[8] and the precision is calculated by summing the similarity scores for all other alternative candidates (false positives)[9]. The reasoning behind this is that if the normalised edit distance is a strong method for identifying the correct replacement in the current environment (i.e. for the current corpus' spelling standardisation) there will be a high similarity score between the variant and the correct replacement and low similarity scores between the variant and alternative candidates.

The letter replacement rules algorithm and the phonetic matching algorithm have also seen an improvement. In VARD 2.2 candidate searches with these methods relied solely on the modern lexicon to find potential word forms, this has been extended to include the manually derived list of variants and their equivalents. For the letter replacement rules algorithm this means that if the variant form being standardised can be transformed into a variant in the known list, the equivalent form which the known variant maps to will be offered as a candidate. For the phonetic matching algorithm, each variant in the known list now has a phonetic code mapped to it, the variant currently being standardised also has a phonetic code calculated, any matches to this code in the known list will result in the mapped equivalents being offered as candidates. These improvements give more scope for recall to both methods, there will also be the added benefit that the two methods will profit from any additions to the known variants list.

4.5 Final Results

With the enhancements described in sections 4.3 and 4.4, the evaluation shown in Figures 5 and 6 was processed once more with the revised version of VARD 2. The final recall and precision scores are shown in Figure 7. In order to show more detail of the early stages of training the first five 1,000 word samples have been broken down into 100 word samples[10]. The results show that the method improvements have been rewarding; without any training whatsoever a recall score of 45% is achieved, an improvement of 5%, with no detraction of the precision score which stays at 92%. The graph also shows the benefit of training with the recall climbing rapidly at first to 50% by 1,000 tokens of training, and then steadily onto 60% by about 12,000 tokens. At this point improvement decelerates, finally reaching 65% at the end of training, after 40,000 tokens have been seen. Throughout this improvement in recall, very little change is recorded for the precision score which maintains a respectable 92–93%.

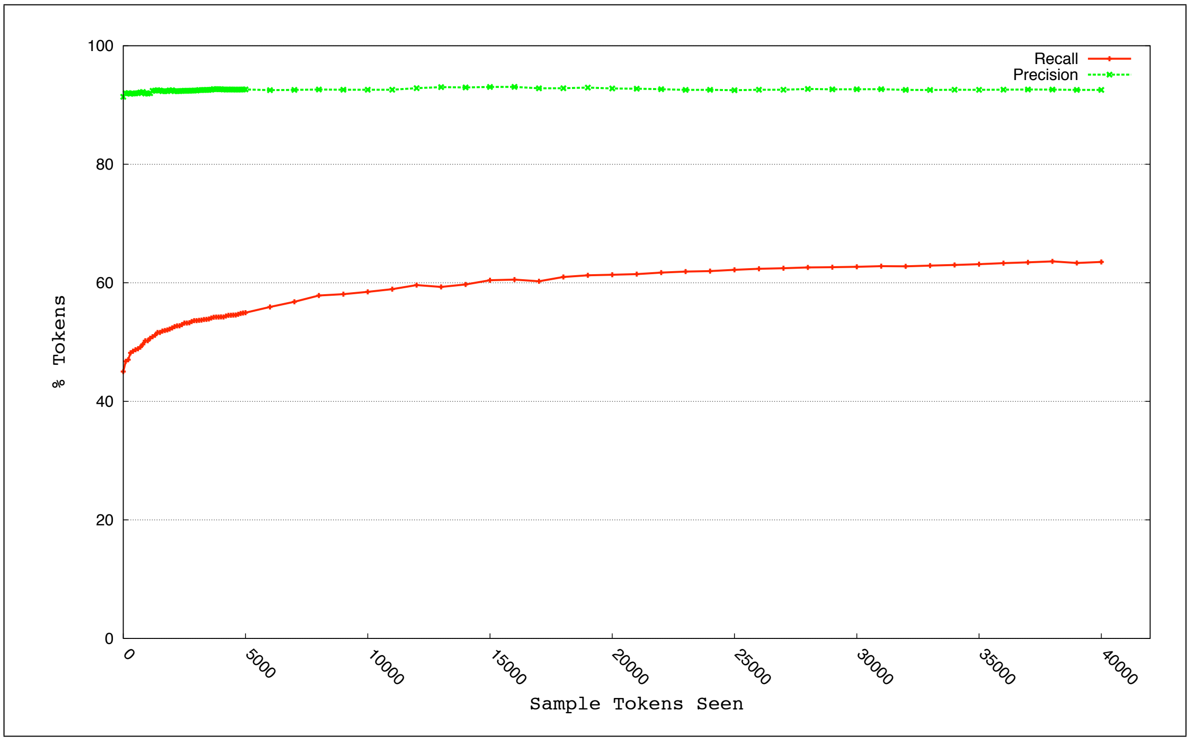


Figure 7: Precision and recall scores of VARD 2 after method improvements.

4.6 Replacement Threshold

One feature of VARD 2 which has not been evaluated formally is the replacement threshold which one can set to control how ‘confident’ the software should be of a replacement candidate when standardising variants. In the evaluation graphs shown previously a threshold of 50% was used, a further test of the newly developed version of VARD 2 was undertaken, this time with thresholds set from 0–100% at 10% intervals. The results are shown in Figure 8 for recall and in Figure 9 for precision. As expected, lower thresholds yield greater recall scores (although no increase in performance is seen below a threshold of 40%) and higher thresholds yield greater precision scores. A threshold of 50% produces a slightly inferior recall and slightly superior precision to that produced for 40% and lower. 60% and 70% eventually converge to nearly 55% recall and 95–96% precision, with greater differences between the two during earlier training. Increasing the threshold to 80% sees a further drop in recall to 50%, but a rise in precision to 96.5% after training. Interestingly, increased training has a negative impact on the recall at 80% (although precision continues to rise), at 90% this is even more apparent with recall dropping to 35%, from a high of 50%, whilst precision continues to rise with increased training.

So which threshold should be used? It is clear that lower thresholds increase recall and higher thresholds increase precision. Furthermore, with increased training lower thresholds continue to improve in terms of recall but weaken in terms of precision, whilst higher thresholds improve in terms of precision and weaken in terms of recall. To take both recall and precision into account simultaneously the scores were combined into an F_1-Measure (see page 12), as shown in Figure 10. Here, 40% (and lower) out-performs larger thresholds, but there is very little reduction in performance with 50% and there is no difference to be seen for thresholds of 70% and lower until after 3,000 tokens of training.

The F_1 measure gives precision and recall equal weighting, however, for standardisation it is likely that precision will be of greater importance. By weighting the F-Measure to consider precision of greater value we can evaluate the performance of different thresholds with precision taking greater importance. Figure 11 shows precision at twice the weight of recall, Figure 12 shows precision at three times the weight of recall and Figure 13 shows precision at four times the weight of recall. At $F_{\frac{1}{2}}$ (Figure 11) thresholds of 70% and lower converge to very similar scores, 70% has a very slight advantage in the early stages of training, but 40% (and lower) are just superior after further training. However once precision is three times the weight of recall (Figure 12) 70% threshold is clearly ahead, with 80% also overtaking 50% and lower. 60% threshold begins worse than 80% but through increased training overtakes 80% to converge with 70%. With further weight given to precision (Figure 13) 50% and lower drop further, even 90% ranking higher until 24,000 tokens of training. At this stage, 80% is the leading threshold with 70% very close behind.

To conclude, which threshold to choose depends greatly on the balance a user wishes to have between recall and precision. If precision is of equal or slightly greater importance to recall, then 50% and lower would appear to be an appropriate threshold to use. If precision is of utmost importance then choosing a higher threshold would be wise, although using a threshold above 80% would have a seriously detrimental effect on recall. Generally, a threshold of 70% would appear to be a sensible compromise.

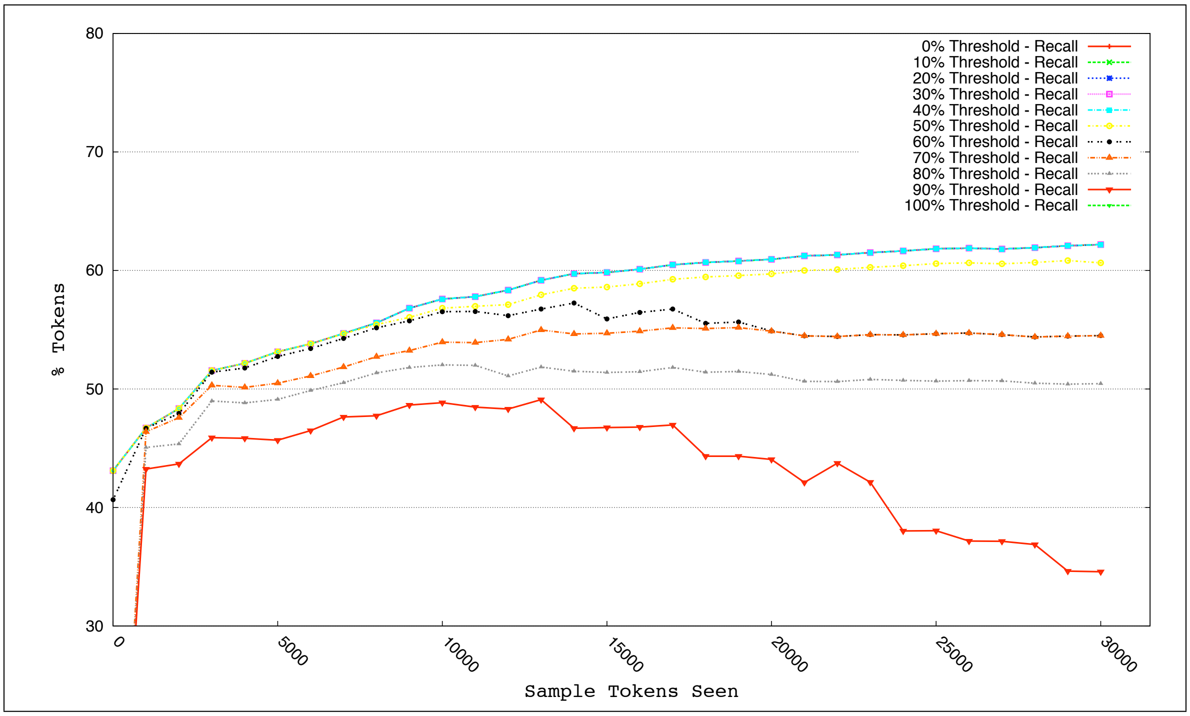


Figure 8: Recall score measured with different replacement thresholds. *Note: y-axis has been shortened to show detail.*

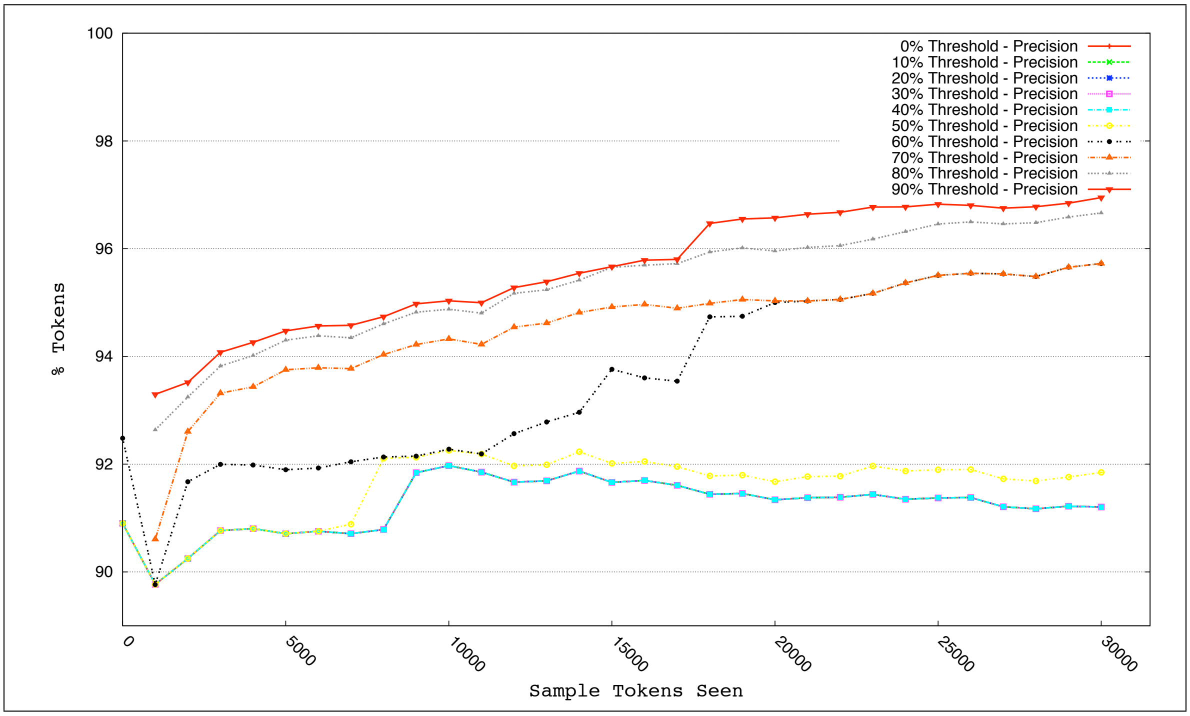


Figure 9: Precision score measured with different replacement thresholds. *Note: y-axis has been shortened to show detail.*

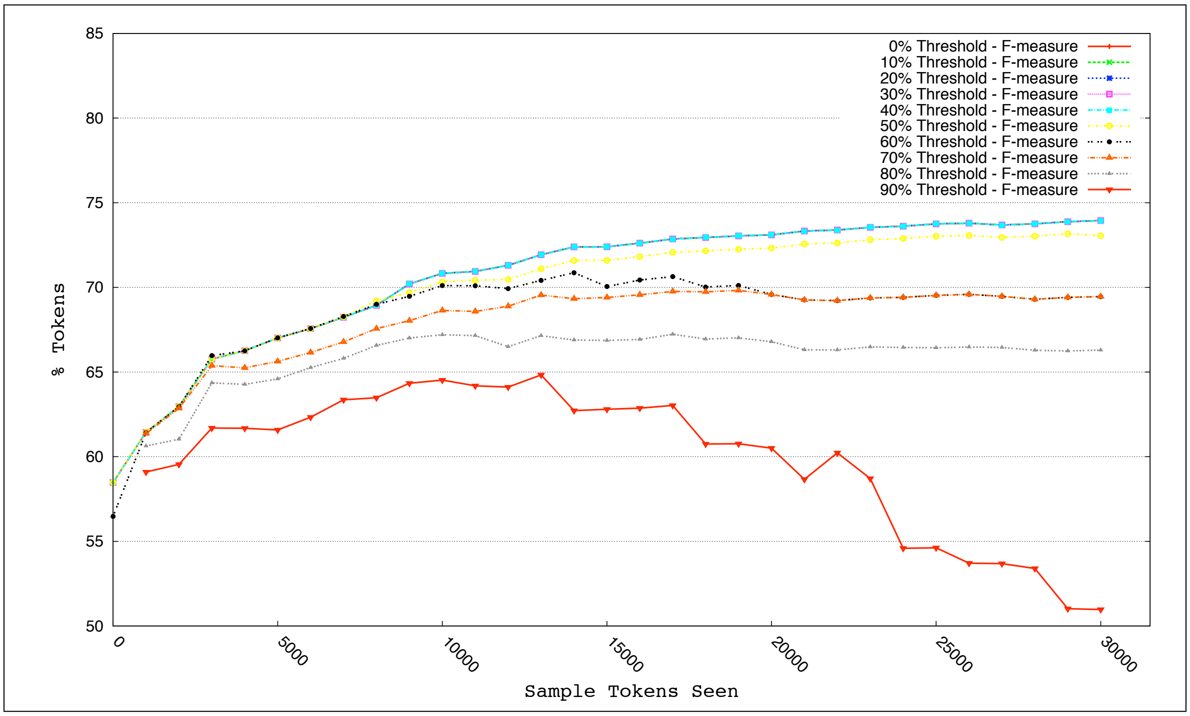


Figure 10: F_1 Measure with different replacement thresholds. *Note: y-axis has been shortened to show detail.*

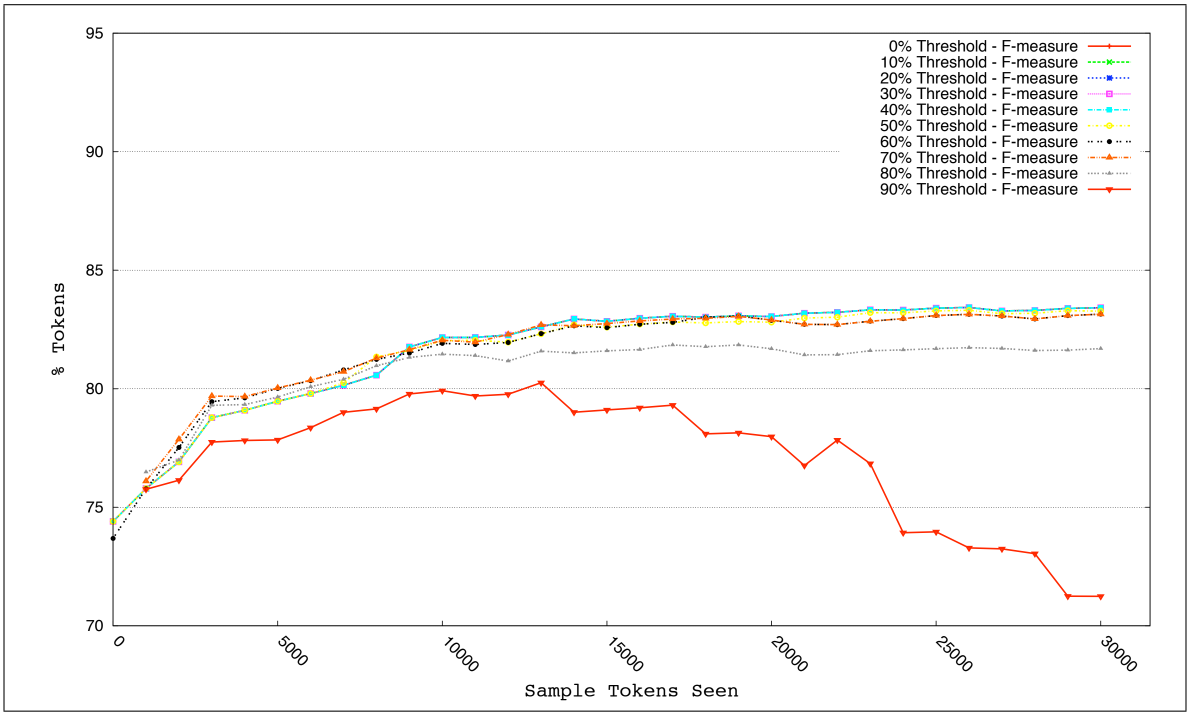


Figure 11: $F_{\frac{1}{2}}$ Measure with different replacement thresholds. *Note: y-axis has been shortened to show detail.*

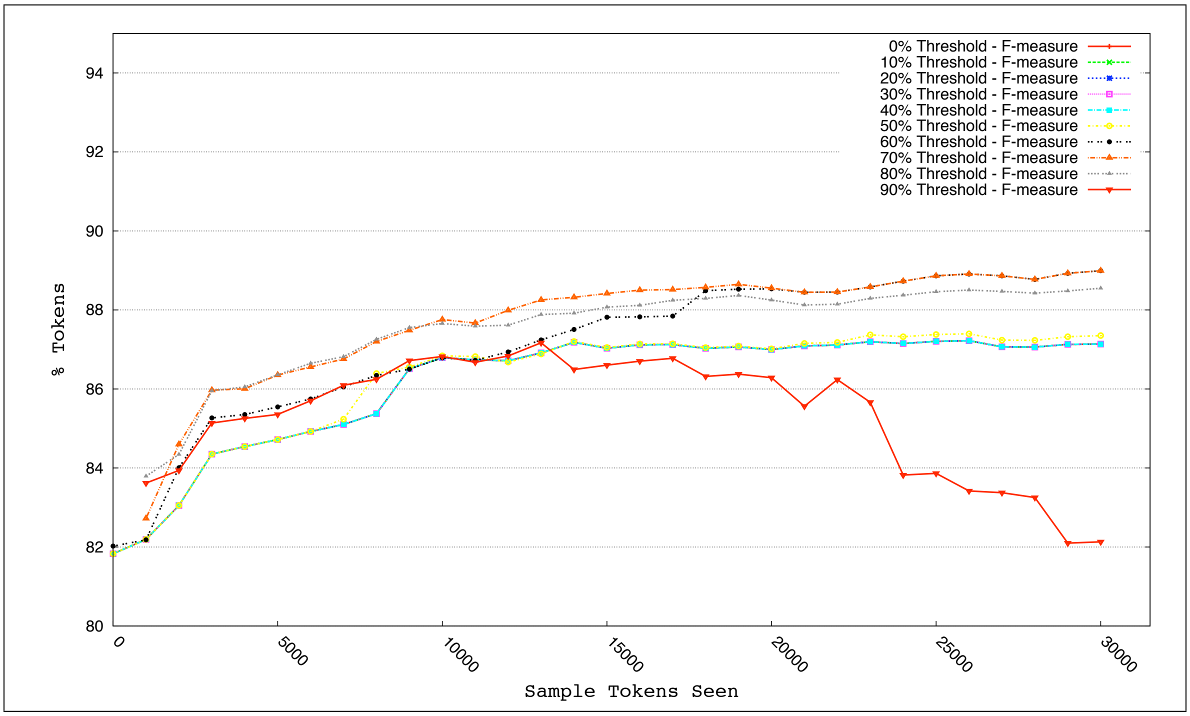


Figure 12: $F_{\frac{1}{3}}$ Measure with different replacement thresholds. *Note: y-axis has been shortened to show detail.*

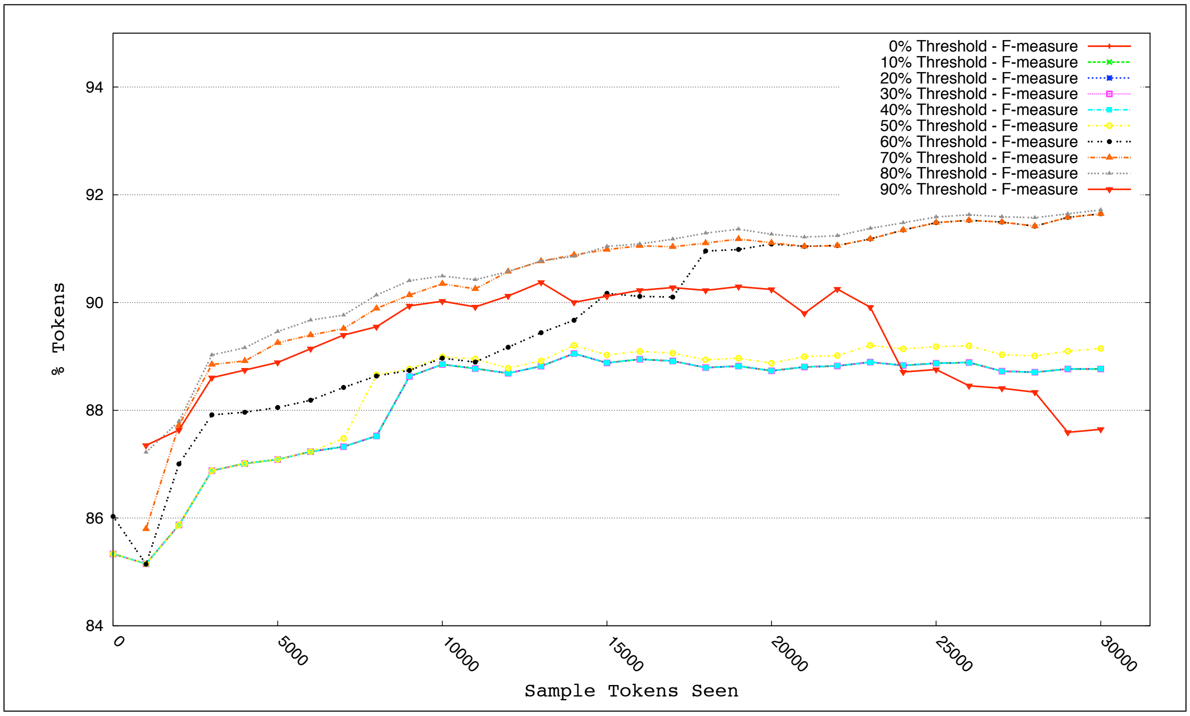


Figure 13: $F_{\frac{1}{4}}$ Measure with different replacement thresholds. *Note: y-axis has been shortened to show detail.*

5 Standardising Child Language Texts

In addition to the Early Modern English experiment described above, we wished to test the VARD 2 tool on a significantly different language variety. Our aim was to have a much clearer idea of the tool's learning capabilities and how robust they are when applied to a completely different type of language. Although the Early Modern English dataset described above had not been used for creating the saved data that comes with VARD 2, it was obviously much closer in terms of variety to the target use of the system. To challenge VARD 2 with a very different character of spelling variation we selected a child language corpus for our experiment.

5.1 Data Used for Training and Evaluation

Ongoing at Lancaster, is a pilot project[11] to undertake the digitisation of the Nuffield Foundation Child Language Survey conducted originally in the 1960s and to collect a modern equivalent for comparison purposes. The original material exists only in hardcopy transcribed format along with the original reel-to-reel audio tapes, and some handwritten essays transcribed into typed hardcopy form. As part of the pilot project, new handwritten short essays were collected from children at a number of schools and transcribed into machine readable format. The essays we utilised for the experiment described here were written by children aged 8–11 years old. As part of the transcription process, spelling errors have been manually corrected and the originals retained. Therefore, the resulting corpus of 253 texts totalling nearly 50,000 tokens formed a suitable gold-standard corpus for our experiments. For details of the transcription format and guidelines, see Pooley et al. (2008). As before, the corpus was split into random samples and half of the data was used to train the tool and the other half was used to test VARD 2's recall and precision after each training sample.

5.2 Results and Discussion

The experiment was performed using the newly developed version of VARD 2 as used in section 4.5 and 4.6, the results are shown in Figure 14. Obviously, the scores are going to be far from the levels found for Early Modern English, however, the results for precision look promising as they show that initially around 80% of standardisations are predicted correctly by VARD 2 and although there is some fluctuation, the reasonably high precision is largely maintained during training. Recall, however, is much lower; with no training the score is below 10%, but through training steady progress is made until a final figure slightly below 20% is reached. The results are not surprising given the small amount of training material used for the experiments and the lack of any customisation of VARD 2 other than automatic training. The promising aspect of the results is that improvements are made with training and with some further development and increased training it is feasible that scores approaching those achieved for Early Modern English are possible.

As with the Early Modern English data, we also experimented with varying the level of VARD 2's replacement threshold. These results can be seen in Figures 15 and 16 for recall and precision respectively. The scores show a similar trend to those for Early Modern English (Figures 8 and 9); lower thresholds yield a greater recall but reduced precision, whilst higher thresholds yield an improved precision at the expense of recall. For Early Modern English it was decided that 70% appeared to be a sensible threshold for general use, here this seems to be the case again, with an admirable precision of 89% being reached by the end of training, whilst recall reaches 16%.

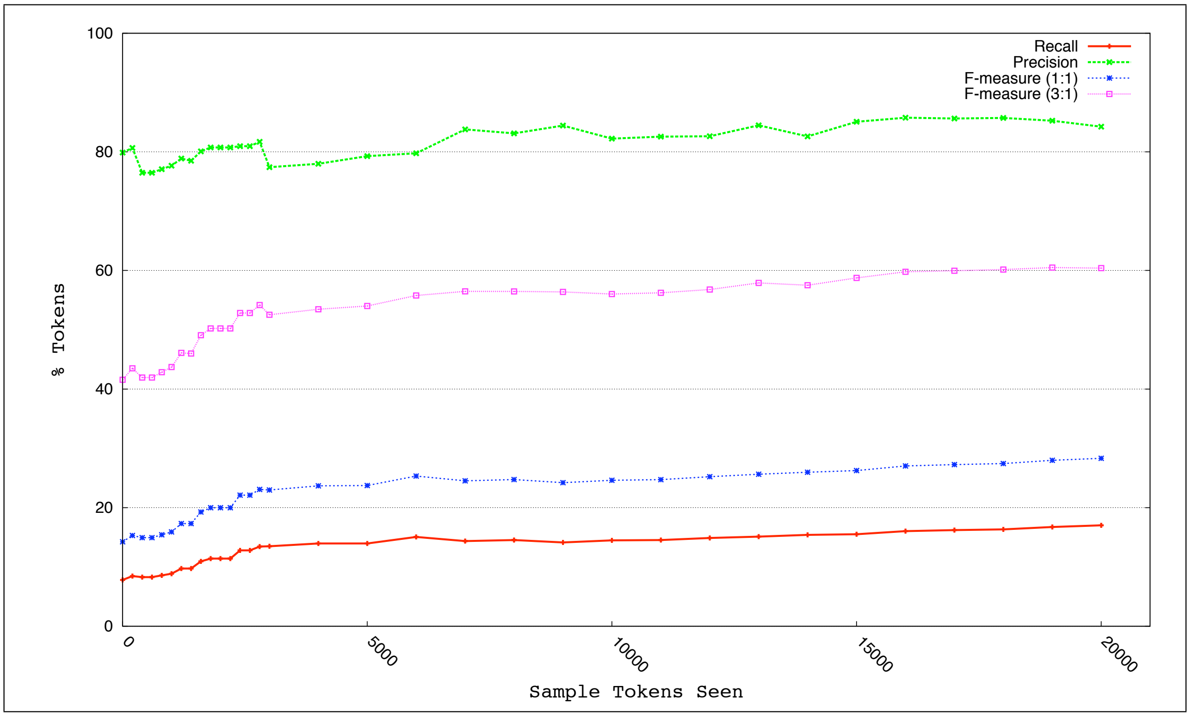


Figure 14: Precision, Recall and F-Measure scores for VARD 2 on child language data.

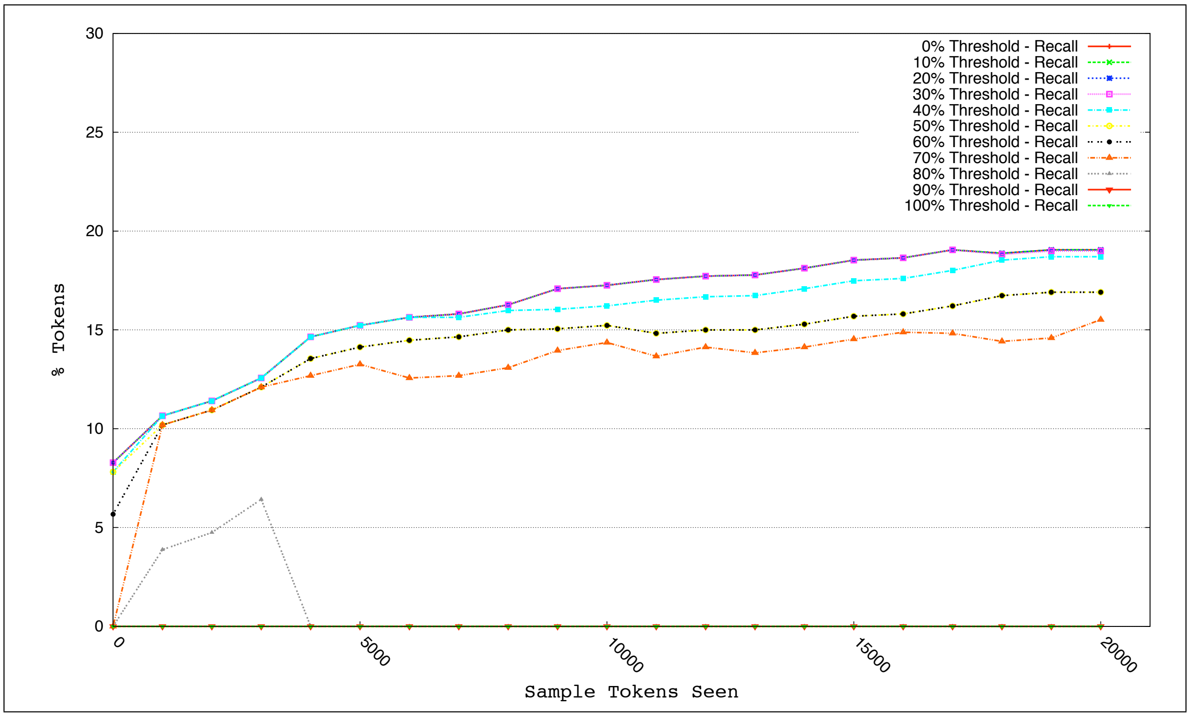


Figure 15: Recall scores for VARD 2 with different replacement thresholds on child language data. *Note: y-axis has been shortened to show detail.*

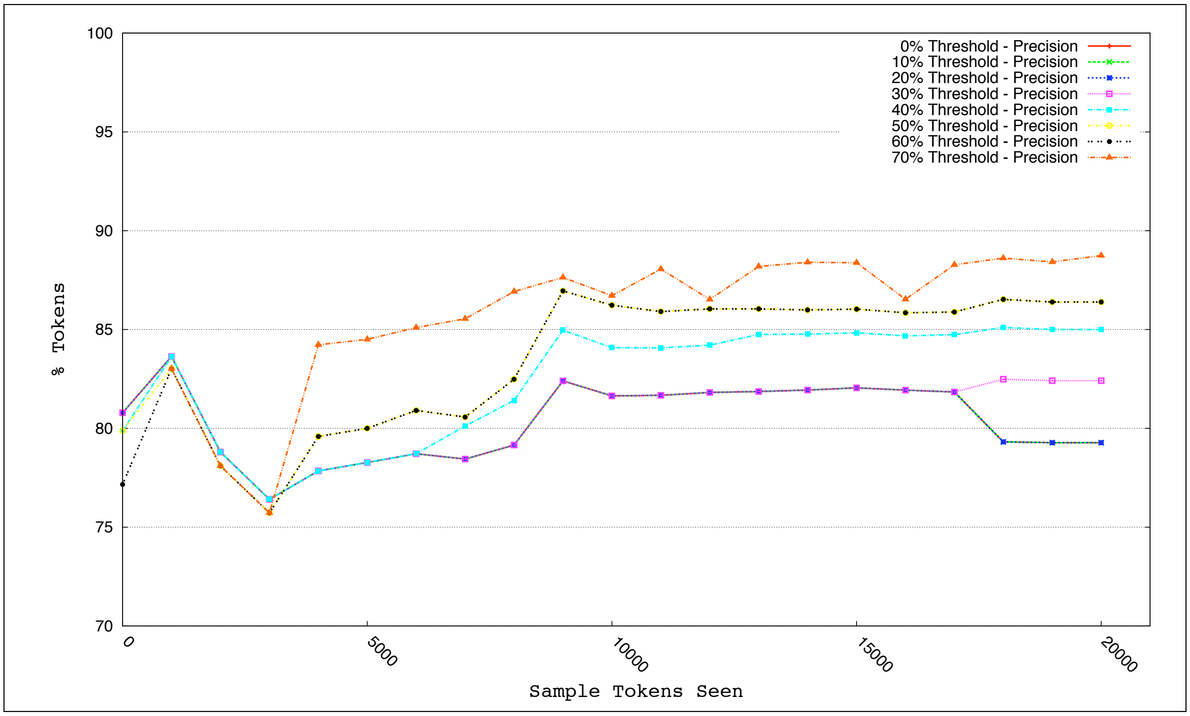


Figure 16: Precision scores for VARD 2 with different replacement thresholds on child language data. *Note: y-axis has been shortened to show detail.*

6 Conclusions and Future Work

We have shown that VARD 2 can be used to automatically standardise spelling variation in Early Modern English with high precision and reasonably high recall. Training of the software increases the performance in terms of recall with no detrimental effect to precision. We need to answer the question “How much training data do you need?”. For Early Modern English, the training and test data we used showed that the first 2–3,000 tokens of training yielded a steep increase in performance, after this a not quite so steep increase in performance continues until 10–12,000 tokens. After this stage, each 1,000 tokens only increases the tool’s performance by a small amount so extra training is not really of significant enough benefit.

With the child language data, VARD 2’s performance is some distance from reaching the levels it achieves for Early Modern English (for which it was designed). However, a reasonably high precision is available and even though the recall scores are low (under 20%), the automatic standardisation should still be of some benefit. Unfortunately, a substantial amount of manual processing will be necessary to reasonably standardise the spelling in such corpora. The training statistics for this data show that the first 3–4,000 tokens of training yielded the majority of improvement in performance, after this point only relatively small increases in the tool’s recall can be achieved.

Even though we can be fairly satisfied with the results of our evaluation, there is a lot of scope for further development of VARD 2 to improve performance further, especially for other language varieties than Early Modern English. The main area currently under development is a further improvement of the letter replacement rules method. We have shown elsewhere (Baron et al., 2009a) that the introduction of information from the DICER tool (Figure 1, page 6) can improve VARD 2’s recall scores substantially. Work is currently underway to amalgamate VARD 2 and DICER, this will allow for a new letter replacement rules algorithm which builds statistics for each individual rule in different contexts, it will also be possible for the tool to dynamically learn new rules as new spelling variants are encountered. This advancement will be of particular use when VARD 2 encounters new language varieties as the tool will no longer be relying upon methods which are specialised for Early Modern English.

Another area of research which could help VARD 2's performance is an investigation into the effect of text source on spelling variation, i.e. the date of publication, the genre, and the author. We have already shown that spelling variation levels curtail through the Early Modern English period (Baron et al., 2009b), it would be beneficial to investigate if there are any changes in specific spelling trends, i.e. certain letter replacement rules becoming more or less prevalent. Nevalainen (2006: 4-6) indicates that such properties will be apparent. With this information, and similar information for other meta-data, it will be possible to build *spelling profiles* which could be used to standardise individual texts more specifically based on source information.

Further evaluation of VARD 2's training capability is planned, its performance could be tested on other text types containing spelling variation; such as, non-native language, SMS messaging, weblogs, emails and chat data. It is also feasible that VARD 2 could be used with other languages than English with some customisation. The amount of extra effort and training needed to accomplish reasonable recall and precision scores will need to be investigated.

Notes

[1] http://ota.ahds.ac.uk

[2] http://books.google.com

[3] http://eebo.chadwyck.com/home

[4] VARD 2 is freely available for academic research from http://www.comp.lancs.ac.uk/∼barona/vard2/.

[5] See http://wordlist.sourceforge.net/scowl-readme.

[6] Various outputs from DICER are available at http://juilland.comp.lancs.ac.uk/dicer.

[7] The minimum edit distance is 1, unless the variant and equivalent are the same, this results in a score always being less than 100%, which is sensible as the system should never be 100% confident of a variant equivalent.

[8] $recall = \frac{tp}{tp+fn}$, fn (false negatives) here is $1 - tp$.

[9] $precision = \frac{tp}{tp+fp}$.

[10] It was infeasible to run more tests at the 100-word rate as the processing time required to process each sample and then test VARD 2 on the test sub-corpus was too expensive when repeating over 100 times.

[11] http://www.lancs.ac.uk/fass/faculty/activities/540/

References

Archer, D., T. Mcenery, P. Rayson, and A. Hardie. (2003). "Developing an automated semantic analysis system for early modern english". In Archer, D., P. Rayson, A. Wilson, and T. Mcenery (eds), *Proceedings of Corpus Linguistics 2003*, 22–31.

Archer, D., A. Ernst-Gerlach, S. Kempken, T. Pilz, and P. Rayson. (2006). "The identification of spelling variants in english and german historical texts: manual or automatic?". In *Digital Humanities 2006*, The Sorbonne, Centre Cultures Anglophones et Technologies de l'Information, Paris, France.

Baron, A. and P. Rayson. (2008). "Vard 2: A tool for dealing with spelling variation in historical corpora". In *Proceedings of the Postgraduate Conference in Corpus Linguistics*, Birmingham. Aston University.

Baron, A., P. Rayson, and D. Archer. (2009a). "Automatic standardization of spelling for historical text mining". In *Proceedings of Digital Humanities 2009*, Maryland, USA. University of Maryland.

Baron, A., P. Rayson, and D. Archer. (2009b). "Word frequency and key word statistics in corpus linguistics". *Anglistik: International Journal of English Studies*, 20(1), 41–67.

Culpeper, J. (2002). "Computers, language and characterisation: An Analysis of six characters in Romeo and Juliet". In Merlander-Marttala, U., C. Ostman, and M. Kytö (eds), *Conversation in Life and in Literature: Papers from the ASLA Symposium*, 15, 11–30. Universitetstryckeriet, Uppsala.

Culpeper, J. and M. Kytö. (1997). "Towards a corpus of dialogues, 1550-1750". *Language in Time and Space. Studies in Honour of Wolfgang Viereck on the Occasion of His 60th Birthday*, 60–73.

Görlach, M. (1991). *Introduction to Early Modern English*. Cambridge University Press, Cambridge.

Grazia, M. d. and P. Stallybrass. (1993). "The materiality of the shakespearean text". *Shakespeare Quarterly*, 44(3), 255–283. URL `http://www.jstor.org/stable/2871419`.

Hodge, V. J. and J. Austin. (2001). "An evaluation of phonetic spell checkers". Technical Report YCS338, Department of Computer Science, University of York.

Kukich, K. (1992). "Techniques for automatically correcting words in text". *ACM Computing Surveys*, 24(4), 377–439.

Kytö, M., M. Rissanen, and S. Wright (eds). (1994). *Corpora across the Centuries: Proceedings of the First International Colloquium on English Diachronic Corpora*, St. Catherine's College, Cambridge. Rodopi, Amsterdam.

Lass, R. (1999). *The Cambridge History of the English Language: Volume III, 1476-1776*, chapter Phonology and Morphology. Cambridge University Press, Cambridge.

Leech, G., P. Rayson, and A. Wilson. (2001). *Word Frequencies in Written and Spoken English: based on the British National Corpus*. Longman, London.

Markus, M. (1999). “Innsbruck computer-archive of machine-readable english texts”. In *Innsbrucker Beitraege zur Kulturwissenschaft, Anglistische Reihe*, 7. Leopold-Franzens-Universitaet Innsbruck, Institut fuer Anglistik, Innsbruck.

Mitton, R. (1996). *English Spelling and the Computer*. Studies in Language and Linguistics. Longman, London and New York.

Nevalainen, T. (1997). “Ongoing work on the corpus of early english correspondence”. *Language and Computers*, 18, 81–90.

Nevalainen, T. (2006). *An Introduction to Early Modern English*. Edinburgh Textbooks on the English Language. Edinburgh University Press, Edinburgh.

Pfeifer, U., T. Poersch, and N. Fuhr. (1996). “Retreival effectiveness of proper name search methods”. *Information Processing and Management*, 32(6), 667–679.

Pooley, N., K. Alcock, K. Cain, A. Hardie, S. Hoffman, and P. Rayson. (2008). “Variability in child language”. In *Posters at ICAME 2008 Conference*, Ascona, Switzerland.

Rayson, P. (2009). “Wmatrix: a web-based corpus processing environment.”. URL `http://ucrel.lancs.ac.uk/wmatrix/`.

Rayson, P., D. Archer, and N. Smith. (2005). “Vard versus word: A comparison of the ucrel variant detector and modern spell checkers on english historical corpora”. In *Proceedings of Corpus Linguistics 2005*.

Rayson, P., D. Archer, A. Baron, J. Culpeper, and N. Smith. (2007). “Tagging the bard: Evaluating the accuracy of a modern pos tagger on early modern english corpora”. In Davies, M., P. Rayson, S. Hunston, and P. Danielsson (eds), *Proceedings of Corpus Linguistics 2007*.

Rayson, P., D. Archer, A. Baron, and N. Smith. (2008). “Travelling through time with corpus annotation software”. In Lewandowska-Tomaszczyk, B. (ed.), *Corpus Linguistics, Computer Tools, and Applications – State of the Art. PALC 2007*, 17 of *Studies In Language*, 29–46, Frankfurt am Main. Peter Lang.

Richardson, M. (1980). “Henry v, the english chancery, and chancery english”. *Speculum*, 55(4), 726–750.

Rissanen, M. (1999). *The Cambridge History of the English Language: Volume III, 1476-1776*, chapter Syntax. Cambridge University Press, Cambridge.

Sampson, G. and A. Babarczy. (2003). “A test of the leaf-ancestor metric for parse accuracy”. *Journal of Natural Language Engineering*, 9(4), 365–380.

Schmied, J. (1994). “The lampeter corpus of early modern english tracts”. In Kytö, M., M. Rissanen, and S. Wright (eds), *Corpora across the Centuries: Proceedings of the First International Colloquium on English Diachronic Corpora*, St. Catherine’s College, Cambridge. Rodopi, Amsterdam.

Scott, M. (2004). *WordSmith Tools version 4*. Oxford University Press.

Sebba, M. (2007). *Spelling and Society*. Cambridge University Press, Cambridge.

Singh, I. (2005). *The History of English*. Hodder Arnold, London.

Taavitsainen, I. and P. Pahta. (1997). “Corpus of early english medical writing 1375-1750”. *ICAME Journal*, 21, 71–81.

Vallins, G. H. and D. G. Scragg. (1965). *Spelling*. André Deutsch, London.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth, London, 2nd edition.

Yannakoudakis, E. J. and D. Fawthrop. (1983). “The rules of spelling errors”. *Information Processing and Management*, 19(2), 87–99.

Zobel, J. and P. Dart. (1996). “Phonetic string matching: Lessons from information retreival”. In *Proceedings of the 18th ACM SIGIR International Conference on Research and Development in Information Retreival*, 166–173, Zurich, Switzerland.