# A Text Normalisation System for Non-Standard English Words

**Emma Flint**[1] **Elliot Ford**[1] **Olivia Thomas**[1] **Andrew Caines**[1] **Paula Buttery**[2]
[1] Department of Theoretical and Applied Linguistics
[2] Computer Laboratory
University of Cambridge, Cambridge, U.K.
{emf40|ef355|oft20|apc38|pjb48}@cam.ac.uk

## Abstract

This paper investigates the problem of text normalisation; specifically, the normalisation of non-standard words (NSWs) in English. Non-standard words can be defined as those word tokens which do not have a dictionary entry, and cannot be pronounced using the usual letter-to-phoneme conversion rules; *e.g.* lbs, 99.3%, #EMNLP2017. NSWs pose a challenge to the proper functioning of text-to-speech technology, and the solution is to spell them out in such a way that they can be pronounced appropriately. We describe our four-stage normalisation system made up of components for detection, classification, division and expansion of NSWs. Performance is favourabe compared to previous work in the field (Sproat *et al.* 2001, Normalization of non-standard words), as well as state-of-the-art text-to-speech software. Further, we update Sproat *et al.*'s NSW taxonomy, and create a more customisable system where users are able to input their own abbreviations and specify into which variety of English (currently available: British or American) they wish to normalise.

## 1 Introduction

The transfer of surface linguistic representations between the written and spoken form is known as 'text-to-speech' (TTS) in one direction and 'automatic speech recognition' (ASR) in the other. In TTS there is a need to map word tokens to a target pronunciation, enabling synthesized speech production. Depending on the text genre, many of the word tokens will map to sound symbols in a straightforward way. For instance, the Carnegie Mellon University Pronouncing Dictionary of English[1] (CMU's PDE) lists more than 134,000 tokens and their pronunciations in ARPAbet form[2] (Table 1).

| Entry | Pronunciation |
|---|---|
| AARDVARK | AA1 R D V AA2 R K |
| CAT | K AE1 T |
| MILK | M IH1 L K |
| PUG | P AH1 G |

Table 1: Example entries from the Carnegie Mellon University Pronouncing Dictionary of English

Tokens such as these may be thought of as the 'standard' set of words – those which have been curated, and continue to be curated, for TTS and ASR.

However, there is another type of word token that does not map straightforwardly to a pronunciation, either because it is an abbreviation or acronym (1), a number (2), a date or time (3), an amount (4), an asterisked profanity (5), a url or hashtag (6), or a spelling error (7).

(1) *kHz*, *Rt. Hon.*, *OED*

(2) *42, (Henry) VIII, 4/5*

(3) *15/04/1997*, *2016-12-31*, *09:30:01*

(4) *€500*, *2000¥*, *99.99%*

(5) *sh*t*, *f**k*, **ss*

(6) http://www.abc123.com, google.com, #summer2016

(7) *anoncement*, *caligaphy*, *helko*

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[2] http://fave.ling.upenn.edu/downloads/ARPAbet.pdf

There are no entries for these examples in CMU's PDE and hence they belong in the set of 'non-standard' words (NSWs) of English.

A normalisation system should automatically detect NSWs in a given text input, identify their type, and spell them out in full such that a TTS system may produce them in a human-interpretable fashion. A successful system must also be able to deal with ambiguities present in real text; the same NSW may be pronounced in multiple ways depending on the context. For example, the number *1985* would normally be pronounced 'one thousand, nine hundred and eighty five' when used as an amount, but 'nineteen eighty five' when used as a year, and "one nine eight five" if read as a sequence of digits. Does *M8* represent the 'm eight' motorway in Scotland or is it shorthand for 'mate'?

Here we present a text normalisation system which sorts an input text into standard and non-standard words, identifies NSW types where appropriate, and expands the NSW to a form ready for speech realisation. The NSW taxonomy is founded on the seminal work by Sproat et al. (2001), with amendments to deal with (a) overlap between class identification and expansion for several classes, (b) finer classification of the numeric NSW group, and (c) developments in web language. For example, the input text below (8) would be normalised as in (9) in order to be read out appropriately by a TTS system (NSWs in bold, expansions italicised):

(8) On the **13 Feb. 2007**, **Rt. Hon.** Theresa May **MP** announced on **ITV** News that the rate of **childhod** obesity had risen from **7.3-9.6%** in just **3** years, costing the **Gov. £20m #politics**.

(9) On the *thirteenth of February two thousand and seven*, *The Right Honourable* Theresa May *M P* announced on *I T V* News that the rate of *childhood* obesity had risen from *seven point three to nine point six percent* in just *three* years, costing the *government twenty million pounds hashtag politics*.

NSW normalisation systems enable a smoother transition between the written and spoken forms of language, rather than skipping NSW tokens, or attempting pronunciations in unexpected or incorrect ways. It is a vital prerequisite for TTS and other downstream natural language processing (NLP) tasks in which technology has been developed on the basis of standard language varieties (Plank, 2016). The system means that texts from a wide range of domains may be read aloud, including newswire, parliamentary proceedings, scientific literature, microblog texts, *etc*. We have made it straightforward to opt for a specific tokenizer, or input a new dictionary of abbreviations, meaning that the system is domain-modifiable whilst still being appropriately domain-general in its foundations. We make our normalisation system publicly available as a GitHub repository[3].

## 2 Related Work

Sproat et al. (2001) remains the single most influential piece of work in the normalisation of NSWs. They were the first to propose a comprehensive taxonomy of NSWs, as well as various heuristics for their expansion. Prior to this, text normalisation had been given limited attention in TTS, and was attempted through the construction of specific rules appropriate for the treatment of NSWs found in the desired domain.

Sproat et al. (2001) proposed an NSW taxonomy based on four distinct domains: newswire, a recipes newsgroup, a hardware-product-specific newsgroup, and classified property advertisements. Their corpora were predominantly U.S. English and an associated set of normalisation tools was made publicly available[4]. Their work has since inspired normalisation research for different text types such as short messaging service (SMS) texts (Aw et al., 2006), email (Moore et al., 2010), and microblog texts from Twitter (Han and Baldwin, 2011). Furthermore, Roark and Sproat (2014) focused on high precision abbreviation expansion, adopting a 'do no harm' approach. We attempt to incorporate some of the normalisation steps taken in these more recent papers, as internet and SMS text has developed in idiosyncratic ways which require normalisation heuristics of their own (Eisenstein, 2013).

We adopt the taxonomy outlined in Sproat et al. (2001) and adapt it to work in a more streamlined manner, and to cope with text domains which are much more prevalent in the present day than at the time of their work – namely the Internet domain. Furthermore, although our system aims to be domain-general, we also allow users the option to input their own dictionary of abbreviations, in

[3] http://github.com/EFord36/normalise
[4] http://festvox.org/nsw

order to tailor towards a specific domain. A further parameter allows the user to specify whether their input variety conforms to British (BrE) or American English (AmE), improving expansion of certain ambiguous tokens, such as dates; *02/03* represents 'the second of March' in BrE, but 'the third of February' in the AmE format. In future work we can incorporate normalisation variants from other Englishes, including 'outer' and 'expanding' circle varieties (Kachru, 1992).

The publicly available resources associated with the previous work required installation of the Festival speech synthesizer[5], were only intended as a "pre-alpha" release and have not been developed since version *0.2.1* in the year 2000. Furthermore, the source code is in Scheme, whereas we release software in the more commonly-used Python programming language.

## 3 Our Approach to Text Normalisation

Our system is made up of separate components for the detection, classification, division and expansion of NSWs. In this section, we outline our method of normalisation by describing each of these modules in turn.

### 3.1 NSW detection

After the input text has been tokenised (either by the user or with our basic tokenizer), non-standard words (henceforth NSWs) are detected. This is achieved by comparison of tokens against a word list, consisting of the set of all English words in a word list corpus, the set of all alphabetic words (greater than four characters) in the Brown Corpus (Francis and Kučera, 1964), and a set of proper names. The Brown Corpus contains 1.15 million words from 500 sources, hence we deemed it to be a good representation of different genres and styles of writing, from fiction to newswire to official documents. We recognise that, as it contains texts from the 1960s and 1970s, the Brown Corpus is by now a little dated. However, its benefits include availability, practicality and coverage. Additionally, we manually add a selection of lexemes which have been coined, or come into greater usage, since the corpus was compiled, such as common technological terms.

In order to facilitate detection of NSWs, we temporarily lower-case and lemmatise the input text (using the WordNet lemmatiser from NLTK (Bird et al., 2009)). This allows us to prevent words whose plural, inflected or capitalised form do not appear in our wordlist from being detected as NSWs. Furthermore, we exclude a number of common contractions which do not appear in our word list from NSW detection (e.g. *aren't*, *won't* and *you're*), on the basis that a normalisation system which expanded these tokens to their full forms would affect the register of the input text (*e.g.* from informal to formal), which is not the purpose of a TTS system. Single punctuation is also prevented from detection, as this provides meaningful information and is important for TTS, whereas nonsense sequences of characters should be detected, and later deleted.

To summarise, a token is detected as an NSW if it satisfies all four of the following conditions:

a. Its lower-cased form is not in the word list.

b. Its lemmatised form is not in the word list.

c. Its non-possessive form (with *'s* or *s'* removed) is not in the word list.

d. It is not single punctuation.

### 3.2 NSW classification

Following detection, NSWs are first classified into one of four general classes: ALPHA (for alphabetic tokens), NUMB (for numeric tokens), SPLT (for mixed tokens that require further division) or MISC (for everything else). Unlike in Sproat et al. (2001), where all NSWs are processed by a splitter, only tokens tagged as SPLT will form the input of our division algorithm. After initial classification and division of SPLT tokens, all NSWs are further classified and labelled with a specific tag to indicate how they should be expanded.

#### 3.2.1 A modified NSW taxonomy

A summary of tags assigned to various NSW tokens can be found in Table 2, a modified version of the taxonomy developed in Sproat et al. (2001), along with a description and examples.

Although our taxonomy is largely consistent with Sproat et al. (2001), a few changes and additions have been made. Sproat and colleagues' MSPL (misspelling), FNSP (funny spelling) and ASWD (read as word) tags have been conflated into a single category WDLK (wordlike), because the effort necessary to distinguish between these tokens is

[5] http://www.cstr.ed.ac.uk/projects/festival

| Class | Tag | Description | Examples |
|---|---|---|---|
| ALPHA | EXPN | abbreviation | *cm* (centimetres), *Dec.* (December), *addr.* (address) |
| | LSEQ | letter sequence | *BBC* (B B C), *U.K.* (U K) |
| | WDLK | word, misspelling | *beatiful* (beautiful), *sllloooow* (slow) |
| NUMB | NUM | cardinal number | *27* (twenty seven), *14.5* (fourteen point five), *2/3* (two thirds) |
| | NORD | ordinal number | *June 3* (third), *15th* (fifteenth), *Louis VI* (sixth) |
| | NRANGE | number range | *25-30* (twenty five to thirty) |
| | NTEL | telephone number | *+447892-739-562* (plus four four seven eight nine two...) |
| | NDIG | number as digits | *123* (one two three) |
| | NTIME | time | *2.45* (two forty five), *17:10* (five ten) |
| | NDATE | date | *19/03* (nineteenth of March), *07-07* (seventh of July) |
| | NADDR | address | *15 Hollybush Ave* (fifteen), *5000 Lensfield Rd.* (five thousand) |
| | NYER | year | *1980* (nineteen eighty), *70s* (seventies) |
| | MONEY | money | *£50* (fifty pounds), *100USD* (one hundred US dollars) |
| | PRCT | percentage | *23.5%* (twenty three point five percent) |
| | NSCI | scientific number | *63.2°N* (sixty three point two degrees north) |
| SPLT | SPLT | mixed | *ITV3* (I T V three), *500-yds* (five hundred yards) |
| MISC | PROF | profanity | sh*t (shit), cr*p (crap) |
| | URL | web address, email | *emf355@hotmail.co.uk* |
| | HTAG | hashtag | *#politics*, *#summer2016* |
| | NONE | not spoken | ?!*?!* |

Table 2: Non-standard word taxonomy.

equal to that used in expansion, rendering it redundant. This reduces the categories defined for alphabetic tokens from four to three.

In addition, SLNT (word boundary or emphasis character, e.g. *seriously*) and PUNC (non-standard punctuation, e.g. ?!*?!*) have been removed, as tokens previously corresponding to these tags can adequately be captured under NONE, given that all such tokens expand to nothing, emphi.e. are deleted and go unspoken. A further omission is that of the NIDE (identifier) tag in the numeric class – the distinction between this and NDIG was unclear.

Finally, we created several new tags in addition to those of Sproat and colleagues, to capture classes we believe to be both distinct and important. One major modernisation is the addition of HTAG, to reflect the growing usage of hashtags on social media platforms such as Twitter and Instagram. Such tokens are distinctive in that words are strung together without spaces or punctuation, making word boundaries (and subsequently the correct expansion) difficult to automatically determine. Additionally, NRANGE has been added to capture number ranges (e.g. *25-30*, *1990-1995*), NSCI to capture scientific numbers, including coordinates, and PROF to cover profanities, which often include an asterisk as a censor.

### 3.2.2 Further classification of ALPHA, NUMB and MISC tokens

The purpose of the classification stage is to assign to each NSW one of the specific tags predefined in our taxonomy (recall Table 2), e.g. EXPN, NRANGE, MONEY, URL etc. A separate classifier is used for each of the ALPHA, NUMB and MISC classes, which also include those NSWs retagged after the division step described below (Section 3.3). Our classification of ALPHA and NUMB tokens uses a semi-supervised label propagation algorithm, while the classification of MISC tokens is entirely rule-based.

We use a number of domain-independent features in training (13 for the ALPHA classifier, and 29 for the NUMB classifier). These look at properties of the token itself, as well as +/-2 surrounding tokens either side of the token in question. This information is important in cases where the class of the NSW is ambiguous, and its correct tag (and subsequently its expansion) can only be determined by the context. For example, a number should be tagged as an ordinal (NORD) when following or preceding a month (*e.g.* ‘On *16* June...’) but a cardinal (NUM) elsewhere (*e.g.* ‘There were *16* people...’).

Properties used in classification include –

- The length of the token.
- Case features: all upper, all lower, titlecase or mixed.
- Specific punctuation used within the token: forward slashes, hyphens, full stops, *etc.*
- The content of surrounding words, *e.g.* preceded by *on, at, from, to*, *etc.*

For the classification of MISC tokens, we use a rule-based method, as there is no (or at least very little) ambiguity in this class compared to ALPHA and NUMB. NSWs are tagged as either HTAG (hashtag), URL (web address) or PROF (profanity) if they conform to a pre-defined regular expression pattern. For example, tokens beginning with a single # character and followed by a series of alphanumeric characters are tagged HTAG. If tokens do not match any pattern, they are tagged as NONE (and later deleted), *e.g.* a series of nonsense characters. After classification, the assigned tag is used to determine how the NSW should be expanded.

### 3.3 Classification and division of SPLT tokens

Many NSWs are compound words made up of distinct subcomponents, which cannot be expanded as they are, but must be broken down for further processing. Examples include mixed alphanumeric tokens, such as acronym-number compounds (*ITV3*), tokens containing mixed upper and lower case letters (*iPlayer*) and hyphenated words (*100-mile*). By classifying into ALPHA, NUMB, MISC and SPLT prior to division, single tokens that would otherwise conform to the SPLT pattern, such as dates and number ranges, are prevented from being incorrectly divided.

With a predefined list of tokens to be split, the division process is relatively straightforward; the same patterns used in the classification of SPLT tokens are used to hypothesise split points. Tokens are split by punctuation (e.g. hyphens, forward slashes), at boundaries between alphabetic and numeric characters and at boundaries between upper and lower case letters. Emphasis characters, such as asterisks, which often surround a word of importance (such as *this*), are also removed.

One ambiguous case arises in words containing a transition from upper to lower case - subtokens here could be an uppercase word followed by a lowercase word (*BBCnews*), necessitating a split after the final uppercase character, or an uppercase word followed by a titlecase words (*BBCNews*), where the split should be before the final uppercase character. For tokens matching this pattern, we deal with the ambiguity by hypothesising both split points, and checking whether the resulting word is in our word list. If neither group is in the word list, we split before the final uppercase letter as a default. This was found to be the more common pattern by Sproat *et al* (2001).

After division, each part of the SPLT token is then retagged as ALPHA, NUMB or MISC for further classification and expansion.

### 3.4 NSW expansion

For the majority of NSW tokens, including all those tagged as NUMB, expansions are unambiguous, and pronunciations straightforward, once the tag is determined. Algorithms for number expansion are predominantly consistent with those in Sproat et al. (2001). However, in some cases where it was necessary to choose between multiple possible pronunciations for a single NSW, we looked at spoken data from the Spoken Wikipedia Corpus (Köhn et al., 2016) in order to make a principled, rather than arbitrary, decision. For example, for the pronunciation of years in the 2000s, we chose ‘twenty thirteen’ rather than ‘two thousand and thirteen’, based on our inspection of the corpus.

#### 3.4.1 Unsupervised expansion of EXPN tokens

EXPN tokens are first checked against a dictionary of common abbreviations, an amended version of a list taken from the *Oxford English Dictionary*[6]. Ambiguous abbreviations in the dictionary (those with more than one possible expansion) are disambiguated in the same way as previously unknown abbreviations (see below), but their candidates are taken from the dictionary rather than generated from the word list. A second dictionary is used for common measurements, and matching NSWs are only expanded as such if the previous token is digit-based, *e.g.* ‘two pounds’ for *2 lb*. This stage allows us to accurately capture the most common abbreviations, whilst still being sufficiently domain-general.

For unusual EXPN tokens whose expansions are not listed in the abbreviation dictionary, we use an unsupervised method to predict the most probable expansion given the abbreviation. The algorithm first generates a list of candidate expansions for the abbreviation. These candidates are words from the word list that include the (ordered) sequence of letters in the abbreviation, either at the start of the word (as in ‘address’ for *addr.*), inserting any numbers of characters before the final letter (‘government’ for *govt.*) or inserting any number of intervening vowels (‘function’ for *fnctn*). This follows from observations as to how abbreviations

[6] http://public.oed.com/how-to-use-the-oed/abbreviations

are most frequently formed. This list is then narrowed down by ruling out those candidates whose part-of-speech (POS) tag does not match the predicted POS tag for the abbreviation based on its syntactic context.

The final criterion for selection of an appropriate expansion uses a Corpus Lesk algorithm (Kilgarriff and Rosenzweig, 2000) to look at the overlap between the abbreviation and its possible expansions. Overlap is calculated by counting the number of words (ignoring stopwords, such as *the, at, of*, as well as the 100 most frequent words in Brown) shared by the context of the abbreviation and a signature generated for each candidate expansion, using contextual information from the Brown corpus, as well as WordNet (Fellbaum, 1998) definitions and examples. Candidates are ruled out if they overlap very little or not at all with the abbreviation. Where two candidates have equal overlap, we take the most frequent word (using frequency counts from Brown). This allows us to check that potential expansions are semantically, as well as syntactically, appropriate. As a candidate is only chosen if its character content, POS-tag and semantic context are consistent with the abbreviation, this allows us to be confident as to the accuracy of the expansion. In this way, we treat the problem of abbreviation expansion in a similar way to that of word sense disambiguation, where the task is to resolve an ambiguity between possible candidate expansions.

Since we use several criteria to predict the expansion of previously unseen abbreviations, this allows generalisation across many different domains. As a result, our method of abbreviation expansion represents a significant improvement over previous normalisation work, which was principally domain-specific.

# 4 Evaluation

Our normalisation system is evaluated against a gold standard corpus containing 1000 sentences taken from various websites including Wikipedia, Google News, Maths is Fun, Slate, the Urban Dictionary and the University of Cambridge. We refer to this corpus as NSW-GOLD and release it in the GitHub repository. NSW-GOLD contains 21,447 tokens in which NSWs were hand-labelled with an overall class (ALPHA, NUMB, MISC or SPLT) and a specific tag. It remains a matter of future work to add multiple gold-standard expansions for the whole corpus, though we did so for a subset, as explained below.

Evaluation was performed for detection, classification and expansion separately. As it is possible for a word to be correctly expanded whilst being incorrectly tagged (and vice versa), we evaluate the performance of each component separately.

## 4.1 NSW detection

As the first stage of our normalisation system (detection) is a binary task, labelling input tokens as either NSWs or standard words, we use simple precision and recall metrics for evaluation. Precision (1) is the number of true positives ($T_p$) over the number of true positives plus false positives ($F_p$), *i.e.* the proportion of tokens labelled 'NSW' that are truly NSWs. Recall (2) is the number of true positives over the number of true positives plus the number of false negatives ($F_n$), *i.e.* the proportion of NSWs in NSW-GOLD that are correctly detected as such.

$$P = \frac{T_p}{T_p + F_p} \quad (1) \qquad R = \frac{T_p}{T_p + F_n} \quad (2)$$

Our evaluation for NSW detection yielded scores of $95.1\%$ for precision and $97.4\%$ for recall. This means that just under $3\%$ of NSWs in NSW-GOLD went undetected but that tokens hypothesised to be NSWs were indeed NSWs 95 times out of 100. Note that we prioritize precision over recall in the detection stage, because if a word is incorrectly tagged as an NSW it should later be classified WDLK (wordlike) and expanded to itself, whereas if an NSW is not detected, it can never be expanded.

## 4.2 NSW classification

In order to evaluate both our overall classifier and our subclassifiers for ALPHA, NUMB and MISC tokens, we computed an accuracy score, where accuracy is the number of correctly labelled NSWs (those whose label matches that in NSW-GOLD) over the total number of NSWs. As tokens only proceed to be classified if they are tagged as NSWs, these accuracy scores do not take into account NSWs that were not detected at the initial stage, but are purely a measure of classification accuracy.

For our ALPHA classifier, accuracy was $89\%$ (Table 3). Within the ALPHA class, performance is high for LSEQ and WDLK but lower for EXPN, reflecting the ambiguity of NSWs tagging. Some NSW tokens, such as *LW*, could reasonably be read either as a letter sequence (LSEQ), or expanded to 'long wave' (EXPN).

For the NUMB class accuracy was found to be 89%. Whilst this performance is good, the task of assigning fine-grained labels to NSWs is much harder. Certain types, namely NUM and NYER may be tagged very accurately; others, such as, NRANGE and NTIME, are harder, whilst NTEL and NSCI were not identified at all. Improvement in identifying these NSW types remains a matter for future work. Nevertheless, in terms of expansion, provided that the NUMB class is correctly identified, many times the exact tag does not matter too much, since for several tag types the NSW will be spelled out like a number or as separate digits. This is fine for most numeric tags, and people are often willing to accept several different expansions of the same NSW, as is clear from human evaluation of expansion (next section). Moreover, this observation suggests we could collapse some of the numeric distinctions in a future review.

The SPLT class sees the lowest accuracy at 86%, which is understandable since – being of mixed content – these are inherently difficult to identify. Finally, for the MISC class accuracy was 92%. Within the MISC class, hashtags are identified without errors, but the PROF,URL,NONE types are identified less well. There are clear improvements to be made in this class in future work.

In all cases, the accuracy scores may be lower than if we had allowed for multiple tags per NSW in our evaluation, thereby reflecting the subjectivity of classification and the multi-functionality of linguistic tokens.

In Table 4 we show a confusion matrix for NSW tag types. It is apparent that errors tend to stay within class, or default to NONE (not spoken). Within the NUMB class, the NUM tag is dominant, which is tolerable as for most numeric types the expansion will be acceptable. It remains a matter for future work to improve our classifiers and add to NSW-GOLD for further evaluation.

### 4.3 NSW expansion – comparison to existing systems

In order to assess the accuracy of our overall system, we compared the output of our system to that of both Sproat et al. (2001)'s original system, and an online interface to the AT&T Natural Voices TTS system, which we believe to be derivative from Sproat and colleagues' work[7]. As Sproat et al. (2001)'s model uses training data from one of four specific domains (a recipes newsgroup, newswire, a PC-hardware newsgroup, and classified property advertisements), we evaluated against each domain separately.

| Class | Accuracy | Tag | Accuracy |
|---|---|---|---|
| ALPHA | 0.893 | EXPN | 0.60 |
| | | LSEQ | 0.90 |
| | | WDLK | 0.92 |
| NUMB | 0.89 | NUM | 1.0 |
| | | NORD | 0.72 |
| | | NRANGE | 0.56 |
| | | NTEL | 0 |
| | | NDIG | 0.12 |
| | | NTIME | 0.72 |
| | | NDATE | 0.34 |
| | | NADDR | 0.12 |
| | | NYER | 0.98 |
| | | MONEY | 0.80 |
| | | PRCT | 0.76 |
| | | NSCI | 0 |
| SPLT | 0.86 | SPLT | 0.86 |
| MISC | 0.92 | PROF | 0.66 |
| | | URL | 0.48 |
| | | HTAG | 1.0 |
| | | NONE | 0.66 |

Table 3: Accuracy of NSW classification by class and tag.

For this comparison, we used a subset of 102 sentences from NSW-GOLD, selected at random. The sample contained 291 NSWs, the expansion of which were hand-annotated as either correct or incorrect in the output generated by each of the five systems. Expansions were labelled as correct if they could realistically have been produced by a human, and would be acceptable if read out by a TTS system. In the case of ambiguity, we accepted both expansions as correct, *e.g.* either 'twenty ten' or 'two thousand and ten' for the year 2010. Here, accuracy is defined as the number of correctly expanded NSWs over the total number of NSWs ($n = 291$).

Our system was found to achieve an overall accuracy of 91.4%, much higher than that of the Wizzard TTS system (75.3%), or any of the domain-specific models (see Table 5). Whilst Sproat et al. (2001)'s system is sure to perform well given data from one of their four specific domains, the inapplicability of their supervised approach to new domains was evident here. For example, when using its classified property advertisements model,

[7] http://wizzardsoftware.com/text-to-speech-sdk.php

|  | Predicted labels |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  | ALPHA |  |  | NUMB |  |  |  |  |  |  |  |  |  |  |  |  | MISC |  |  |  |  |
| **Actual** | EXPN | LSEQ | WDLK | NUM | NORD | NRANGE | NTEL | NDIG | NTIME | NDATE | NADDR | NYER | MONEY | PRCT | NSCI | SPLT | PROF | URL | HTAG | NONE | *n/a* |
| EXPN | 30 | 12 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 0 |
| LSEQ | 0 | 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 |
| WDLK | 0 | 0 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |
| NUM | 0 | 0 | 0 | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NORD | 0 | 9 | 1 | 2 | 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| NRANGE | 0 | 0 | 0 | 16 | 0 | 28 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 |
| NTEL | 0 | 0 | 0 | 43 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 |
| NDIG | 0 | 0 | 0 | 42 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| NTIME | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 0 | 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NDATE | 0 | 0 | 0 | 19 | 0 | 4 | 0 | 10 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NADDR | 0 | 0 | 0 | 40 | 2 | 1 | 0 | 0 | 0 | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| NYER | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 49 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MONEY | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| PRCT | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 38 | 0 | 0 | 0 | 0 | 0 | 7 | 0 |
| NSCI | 0 | 0 | 0 | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 37 | 0 |
| SPLT | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 43 | 0 | 1 | 0 | 0 | 0 |
| PROF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33 | 0 | 0 | 17 | 0 |
| URL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 24 | 0 | 26 | 0 |
| HTAG | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 | 0 | 0 |
| NONE | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 33 | 15 |

Table 4: Confusion matrix for classification of NSW tags in NSW-GOLD.

| **Our system** | **AT&T** | Sproat et al. (2001) |  |  |  |
|---|---|---|---|---|---|
|  | Natural Voices | **Recipes** | **Hardware** | **News** | **Ads** |
| **0.914** | 0.753 | 0.460 | 0.536 | 0.601 | 0.574 |

Table 5: Cross-system comparison of NSW expansion accuracy

the system returned incorrect expansions such as 'kitchen H Z' for *kHz*. The total run time for our program compared to the original was also found to be significantly faster[8].

## 5 Conclusion and Future Work

We have presented a text normalisation system for NSWs, adopting and adapting the taxonomy designed by Sproat et al. (2001), and showed that our modular detection-classification-division-expansion system works to a high degree of accuracy across all NSW types and modules ($>$ 91%). This is an important step for TTS systems and other downstream NLP tasks. The system is made available as a GitHub repository[9] for non-commercial use under a GNU General Public License[10]. In contrast to a previous system written in Scheme (Sproat et al., 2001), the fact our system has been developed in the widely-used Python programming language makes it flexible to the unforeseen needs of other researchers. In addition, we have made it straightforward to opt for a specific tokenizer, or input a dictionary of abbreviations, meaning that the system is domain-modifiable whilst still being appropriately domain-general.

In future work, we intend to improve our system by addressing the NSW tag types for which performance was relatively poor, and by extending our taxonomy to include more tags specific to the web. We would also like to allow the generation of multiple expansions, to capture ambiguity and different pronunciation preferences. We can also further test our system against modern TTS systems available through Google Android Apps, Apple Macintosh OS, and Microsoft Office.

A further problem for normalisation is that the boundary between standard words and NSWs is not always rigid. Some words, such as proper nouns, foreign words or company names, may not have a dictionary entry (or pronunciation in the CMU), but should not (and cannot) be further expanded, thus a normalisation system would be un-

[8] Mean run time 1 minute 50 seconds for Sproat and colleagues' system; 22 seconds for ours, averaged over 100 runs on a Macintosh iMac with 2.7GHz Intel Core i5 processor and 8GB memory.

[9] http://github.com/EFord36/normalise

[10] https://www.gnu.org/licenses/gpl-3.0.en.html

able to aid TTS in these cases. This is an area in need of further investigation and system development.

Having updated the NSW taxonomy and adopted a rule-based approach to classification, the system remains vulnerable to further macro-scale shifts in language use such as that brought on by the Internet, and the kind of micro-scale non-standard neologisms which emerge (and recede) day-to-day. In future work we can therefore incorporate unsupervised methods of NSW classification and expansion, along the lines of the automatic dictionary construction method described by Han et al. (2012), and the distributional method described by Rangarajan Sridhar (2015).

Additional areas of future interest might be in developing a 'reverse text normalisation' system for Automatic Speech Recognition (ASR) – a backwards conversion of speech into non-standard text, *e.g.* numbers. Finally, a cognitive computational investigation comparing speech production errors and NSW classification errors is a research question of general interest.

## Acknowledgements

This paper reports on research supported by Cambridge English, University of Cambridge. We are thankful for support from Sidney Sussex College and Downing College, Cambridge. And we gratefully acknowledge Dimitrios Alikaniotis and Ernst for their help with this work. We thank the three reviewers for their very helpful comments and have attempted to improve the paper in line with their suggestions.

## References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Session*. Association for Computational Linguistics.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly Media.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of NAACL-HLT 2013*. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

W. Nelson Francis and Henry Kučera. 1964. *Manual of information to accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Braj Kachru. 1992. *The Other Tongue: English across cultures*. Chicago: University of Illinois Press.

Adam Kilgarriff and Joseph Rosenzweig. 2000. English senseval: Report and results. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)*. European Language Resources Association (ELRA).

Arne Köhn, Florian Stegen, and Timo Baumann. 2016. Mining the spoken wikipedia for speech data and beyond. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).

Stuart Moore, Sabine Buchholz, and Anna Korhonen. 2010. Annotating the enron email corpus with number senses. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

Barbara Plank. 2016. What to do about non-standard (or *non-canonical*) language in NLP. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*.

Vivek Kumar Rangarajan Sridhar. 2015. Unsupervised text normalization using distributed representations of words and phrases. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*.

Brian Roark and Richard Sproat. 2014. Hippocratic abbreviation expansion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Richard Sproat, Alan Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech and Language* 15:287–333.