AUTOMATIC DETECTION OF ANGLICISMS FOR THE PRONUNCIATION DICTIONARY GENERATION: A CASE STUDY ON OUR GERMAN IT CORPUS

Sebastian Leidig, Tim Schlippe, Tanja Schultz

Cognitive Systems Lab, Karlsruhe Institute of Technology (KIT), Germany

ABSTRACT

With the globalization more and more words from other languages come into a language without assimilation to the phonetic system of the new language. To economically build up lexical resources with automatic or semi-automatic methods, it is important to detect and treat them separately. Due to the strong increase of Anglicisms, especially from the IT domain, we developed features for their automatic detection and collected and annotated a German IT corpus to evaluate them. Furthermore we applied our methods to Afrikaans words from the *NCHLT* corpus and German words from the news domain. Combining features based on grapheme perplexity, grapheme-to-phoneme confidence, *Google* hits count as well as spell-checker dictionary and *Wiktionary* lookup reaches 75.44% f-score. Producing pronunciations for the words in our German IT corpus based on our methods resulted in 1.6% phoneme error rate to reference pronunciations, while applying exclusively German grapheme-to-phoneme rules for all words achieved 5.0%.

Index Terms— Foreign entity detection, lexical resources, pronunciation modeling, Anglicisms

1. INTRODUCTION

As English is the prime tongue of international communication, English terms are widespread in many languages. This is particularly true for the IT sector but not limited to that domain. For example, African and Indian languages [1][2], of which many are still under-resourced for speech technology, use a lot of borrowed English words. Anglicisms – i.e. words borrowed from English into another language (the so called *matrix language*) – nowadays come naturally to most people but this mix of languages poses a challenge to systems dealing with language or speech. With more than 6,900 languages in the world, the biggest challenge today is to rapidly port speech processing systems to new languages and domains with low human effort and at reasonable cost. This includes the creation of qualified pronunciation dictionaries. Automatic speech recognition (ASR) and speech synthesis systems need correct pronunciations for these words and names of English origin. However, a grapheme-to-phoneme (G2P) model of the matrix language, which is usually employed to rapidly and economically generate pronunciations, does not often give appropriate pronunciations for these words. Nowadays many people are fluent in English and pronounce Anglicisms according to their original pronunciation. An automatic detection of Anglicisms enables us to use more adequate English pronunciation rules to generate pronunciations for them. Adding pronunciation variants for foreign words to the dictionary can reduce the word error rate (WER) of ASR systems as shown in [3]. Therefore we develop new methods to automatically detect Anglicisms from word lists of different matrix languages and advance existing approaches. The term *matrix language* designates the main language of a text from which we try to distinguish the inclusions of English origin.

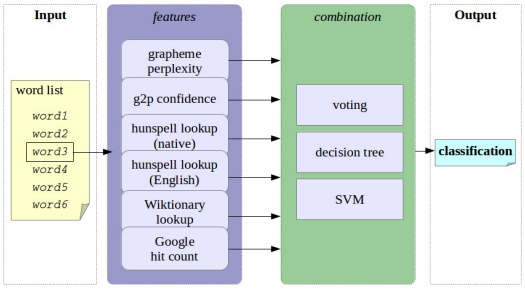

Fig. 1. Overview of the Anglicism detection system

We evaluated our features on the two matrix languages German and Afrikaans. However, our methods can easily be adapted to new matrix languages.

In our scenario we receive single words from texts in the matrix language as input. As output we produce a classification between the classes *English* and *native*. While some related work relies on information about the word context (e. g. part-of-speech), we concentrate on context-independent features of the examined word itself to classify it. This flexibility is usefull, as dictionaries are usually based on lists of most frequent words [4]. Other features which use information about the word context can still be integrated in future work. As shown in Fig. 1, we developed and evaluated a set of different features to detect English words in word lists:

- Grapheme perplexity
- G2P confidence
- Hunspell spell-checker dictionary lookup
- Wiktionary lookup
- Google hit count

Those features were separately tuned and evaluated before we proceeded to combine them. For the combination we experimented with different methods:

- Voting
- Decision Tree
- Support Vector Machine (SVM)

We leverage different sources of expert knowledge and unannotated training text to create features that are mostly language-independent and cheap to set up. By developing features based on commonly available training data like unannotated word lists, spell-checker or pronunciation dictionaries, we avoid the expensive step of hand-annotating Anglicisms directly in lots of training data. This also enables to use available frameworks for the implementation of our approaches (e.g. the SRI Language Modeling Toolkit [5] for the *Grapheme Perplexity Feature* or Phonetisaurus [6] for the *G2P Confidence Feature*). A more expensive resource which boosts our

Category	
English	All English words were tagged as "English". This comprises all types of words including proper names and also pseudo-Anglicisms. Words that could be German as well as English (homomorph words) were not tagged as English (e.g. *Admiral*, *Evolution*, ...). Neither are loan translations tagged as English. Words that contain an English part (see *Hybrid foreign word*) were tagged as English because a monolingual German G2P model cannot generate correct pronunciations for those words.
abbreviation	Abbreviations were tagged as "abbreviation". We did not distinguish between English and German abbreviations as our focus is to detect whole words with English part. Therefore no abbreviations were tagged as English.
other foreign word	Foreign words that are neither German nor English were tagged as "foreign". As we limit our algorithms to classify exclusively between the categories *English* and *native* (in this case German), these words fall into the *native* category.
hybrid foreign word	Words that contain an English plus a German part were tagged as "hybrid" in addition to "English". This covers for example compound words with a German and an English part (e.g. "Schad*software*") and grammatically conjugated forms of English verbs (e.g. "*download*en").

Table 1. Annotation Guidelines for the German test sets.

G2P Confidence Feature may be a pronunciation dictionary of the matrix language. For English and many other languages, dictionaries are available. To account for scenarios where a pronunciation dictionary is not available or of poor quality, we also evaluated our *G2P Confidence Feature* in simulated situations with pronunciation dictionaries containing only a small number of entries.

2. RELATED WORK

Specific treatment of foreign inclusions for the pronunciation dictionary generation improved text-to-speech system performance [7] and ASR performance. [3] added pronunciation variants for automatically detected foreign words and reduced the WER of a Finnish ASR system by up to 8.8% relative. [1] reduced the WER of a Swahili ASR system from 26.9% to 26.5% by adding English pronunciations variants for the almost 9% of the words in their Swahili dictionary that also appeared in the English CMU dictionary [8]. Moreover, a foreign word detection improves part-of-speech parsing as reported in [9]. A simple approach to detect foreign words in word lists and generate different pronunciations for them has already been patented [10]. There have been many approaches based on grapheme-level methods, mostly based on grapheme n-gram likelihoods. [3] focused on the effects of pronunciation variants on ASR and used a simple grapheme perplexity (PPL) threshold, treating the 30% of words with the highest PPL as foreign word candidates. [11] and [12] compared syllable probabilities between a Korean and a foreign model and extracted the foreign word stem. [13] developed a "Cumulative Frequency Addition" that distinguishes between a number of different languages. Thereby grapheme n-gram frequencies within each language model (LM) and over all languages are calculated to classify a word. While [14] worked with word-based Hidden-Markov-Models (HMMs), [15] switched to character-level HMMs thereby achieving high error reduction. We compare grapheme n-gram probabilities after converting them to PPLs for our *Grapheme Perplexity Feature*. Another common approach to foreign word detection is a dictionary lookup. Even if grapheme n-grams performed better than lexicon lookup, their combination gave the best results in [16]. [17] used a dictionary lookup to reduce the number of English word candidates before applying more costly features. [18, 19] use the open source spell-checker and morphological analyzer Hunspell. Our *Hunspell Lookup Feature* uses a similar approach, also basing its classification on Hunspell lookups. An innovate method is the comparison of the number of search engine results found for different languages [17] which we reimplemented for our *Google Hit Count Feature*. [2] interpolated probabilities of grapheme and phoneme LMs for English and Bangla. Their classification is based on a comparison between those probabilities. The phoneme sequences are generated with a G2P converter producing the pronunciations for Bangla and English transliterated words. Our *G2P Confidence Feature* uses a similar approach, also basing its classification on a combination of phoneme- and grapheme-level information. For our feature we compare probabilities of graphone-level models. For named entity recognition, often the local context or specific trigger words are used. Part-of-speech (POS) tags, capitalization and punctuation are also common features as shown in [20] and [21]. The detection performance is usually evaluated in terms of f-score with equal weight for precision and recall (f_0-score) [22]. Results vary for the different methods and setups in related work. [12] achieve 88.0% f-score detecting foreign transliterations in Korean. [13] reach 79.9% distinguishing between several language pairs. Detecting English inclusions in German text, [17]'s experiments are very similar to ours and give comparable results of up to 77.2% f-score.

3. EXPERIMENTAL SETUP

3.1. German IT Corpus

Our German IT corpus *Microsoft-de* contains about 4.6k word types crawled from the German website of Microsoft `www.microsoft.de`. To reduce the effort of hand-annotating, this word list only contains frequent types that occured more than once in the crawled text. Before extracting the types for our word list, some normalization and cleanup was performed on the crawled text. We removed all HTML tags, sentences containing more than 80% capital letters and replaced punctuations marks including hyphens with spaces.

3.1.1. Annotation Guidelines

In our German word lists English words and some additional word categories for further analyses were annotated. Like [17], we base our annotation on the agreement of the annotators. In case of disagreement we consulted the well-known German dictionary Duden (`www.duden.de`) and checked the context in which the word occured in the text. The annotation of the German word lists follows the guidelines described in Tab. 1.

3.1.2. Reference Dictionary

We selected 824 sentences, containing 2,276 unique words (types) from the *Microsoft-de* corpus. For this vocabulary we created a reference pronunciation dictionary. The pronunciations for words annotated as English were generated with an English G2P model that was built from the *CMU Pronouncing Dictionary* (*CMUdict*) [8]. The pronunciations for non-English words were generated with a German G2P model which was generated from the German *GlobalPhone* dictionary (*GP-de*) [23]. The pronunciations for hybrid English words were created manually. To avoid ambiguous pronunciations for abbreviations, we only selected sentences that do not contain any abbreviation. For the whole dictionary we use the German phoneme set from *GP-de*. Pronunciations generated with the

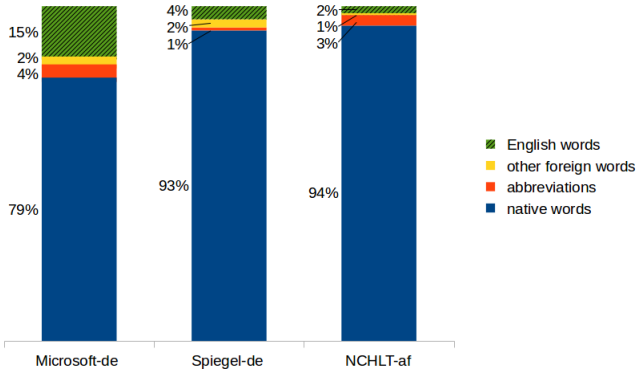

Fig. 2. Foreign words in different word lists

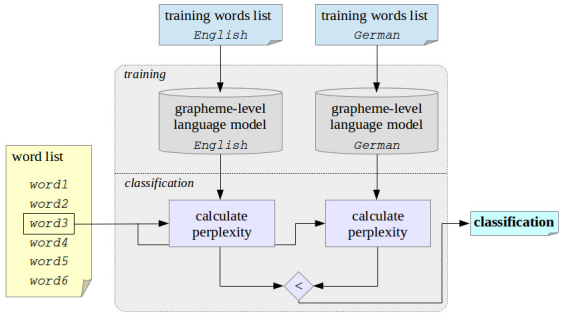

Fig. 3. Classification with the Grapheme Perplexity Feature

English G2P model were mapped to this German phoneme set based on the IPA scheme [24].

3.2. Other Domains and Languages

To compare the detection performance on different domains and languages, we use two more annotated word lists. The general news domain word list *Spiegel-de* contains about 6.6k types from 35 articles covering the domain of German political and business news. The texts were manually taken from the website of the German news journal Spiegel `www.spiegel.de`. The texts have not been crawled automatically to keep the word list clean of advertisements, user comments and other unwanted content. The punctuation marks were removed. The *NCHLT-af* word list contains about 9.4k types taken from the Afrikaans part of the *NCHLT* corpus [25], which contains a collection in the eleven official languages of South Africa. In our Afrikaans test set English, foreign words and abbreviations have been annotated by [26]. The authors kindly provided this annotated word list for our experiments.

3.3. Distribution of the Word Categories

Fig. 2 demonstrates the distribution of the word categories in our four word lists. Especially, in the IT domain we find many *foreign words* and *abbreviations*. Those are more than 21% of the *Microsoft-de* word list, where 15% of all words are *English*. In the general news domain (*Spiegel-de*) we find only approximately 4% *English* words. About 10% of the English words in our German word lists from each domain are *hybrid* words, consisting of German and English parts (e.g. "Schad*software*"). The Afrikaans *NCHLT* corpus contains only 2% *English* words and 1% *abbreviations*.

4. EXPERIMENTS AND RESULTS

4.1. Features for Anglicism detection

To detect Anglicisms we advanced existing methods and developed entirely new features. In addition to the evaluation of new approaches, an important goal was an inexpensive setup and the portability to new languages. In contrast to standard supervised machine learning, our features do not rely on training data that is annotated specifically for the task of Anglicism detection. The test sets presented in the previous section are only used for evaluation of our single features and never for their training. Instead we use common resources like word lists and pronunciation or spell-checker dictionaries. Exceptions are only our feature combinations, as described in Section 4.2, which are trained in a cross-validation on the test sets. To avoid supervised training of thresholds, for most features we base our classification on the difference between results calculated on an English model and a model of the matrix language. This also improves the detection performance significantly.

4.1.1. Grapheme Perplexity Feature

The grapheme-level detection of foreign words is based on the assumption that common grapheme sequences differ depending on the language. For example, in our German word list 25.8% of the words end with "en" but only 1.7% in our English word list. A grapheme (or character) n-gram is a sequence of n graphemes. Grapheme-level LMs are trained from lists of training words. These models are a statistical representation of grapheme n-grams over all training words. In addition to the graphemes, word boundary symbols are included to specifically identify the grapheme n-grams at the beginning and end of words. We used the SRI Language Modeling Toolkit (Available at `http://www.speech.sri.com/projects/srilm/`) [5] to build our models. The detection based on grapheme n-gram models deals well with conjugations and small variations of words. Unknown forms of a word can still be recognized because the overall grapheme sequences stay similar. Therefore many works in the field of Named Entity Recognition and Foreign Entity Recognition are based on grapheme n-grams ([15], [3], [11], [12], [13]).

We experimented with different training word lists and parameters to build grapheme-level LMs. The best detection performance was achieved using case-insensitive 5-gram models built from lists of unique training words. To train the grapheme LMs, we used 116k word types from the *CMU Pronouncing Dictionary* (*CMUdict*) [8] for English and 37k from the German *GlobalPhone* dictionary [23] for German. The Afrikaans model was trained with 27k word types crawled on the Afrikaans news website `www.rapport.co.za`. To port this feature to another language, an unannotated word list from that language is sufficient as long as grapheme sequences of that language are more likely in this word list than in the English one. Our approach of using the PPL difference between two models allows us to have an unsupervised classification based on a direct comparison of PPLs for an English model and a model of the matrix language. Furthermore it enables a focus on Anglicisms instead of a broad recognition of uncommon words. Fig. 3 depicts the steps of our *Grapheme Perplexity Feature*:

1. Preparation: Training of grapheme-level LMs from training word lists for English and the matrix language
2. Calculation of the PPL on the English model and the model of the matrix language for a word from the test set
3. Comparison of the two PPLs and classification towards the model whose PPL is lower for the word

Test set	Threshold = 0	Optimal threshold	
	F-score	F-score	Threshold
Microsoft-de	67.17%	68.56%	0.5
Spiegel-de	36.00%	45.61%	6.5
NCHLT-af	25.75%	29.87%	2.5

Table 2. Detection performance with different thresholds for the grapheme perplexity difference

The feature uses the difference of the English and matrix language PPLs. We calculate

$$d = \text{ppl}_{matrixlang.}(w) - \text{ppl}_{English}(w)$$

and classify a word w as English if the difference d is greater than zero. We generically assume a threshold of zero, which leads to a simple comparison of which PPL is smaller. This is not an optimal choice as shown in Tab. 2. We still make this trade-off to refrain from supervised training of a better threshold. The different optimal thresholds seem to be related to the portion of Anglicisms in the test set. *Microsoft-de* contains almost four times as many Anglicisms as *Spiegel-de*. A further normalization by standard score (z-score) over the PPLs of all words of the test set led to worse results.

4.1.2. G2P Confidence Feature

We use *Phonetisaurus* [6], an open source weighted finite state transducer based G2P conversion toolkit, for our experiments. *Phonetisaurus* takes the following steps to predict pronunciations:

1. Alignment of graphemes and phonemes in the training dictionary (creating graphones)
2. Training of a graphone-level LM
3. Prediction of pronunciations for novel words

In the alignment step graphemes are combined with the phonemes from the corresponding pronunciation. The resulting grapheme-phoneme clusters are usually named *graphones* in literature [27]. Then a 7-gram graphone-level LM is trained from all graphone sequences of the training dictionary. To predict pronunciations, *Phonetisaurus* searches the shortest path in the G2P model which corresponds to the input grapheme sequence. As path costs the graphones' negative log probabilities are summed up. This value can be interpreted as a confidence measure: It is used to rank different pronunciation variants. In our experiments, we use this G2P confidence to measure the "sureness" between a word's pronunciation variants generated from different G2P models. To train G2P models, we use 133k word-pronunciation pairs from the *CMU Pronouncing Dictionary* (*CMUdict*) [8] for English, 38k from the German *GlobalPhone* dictionary (*GP-de*) [23] for German and the Afrikaans pronunciation dictionary (*dict-af*) created by [28] (42k word-pronunciation pairs). Our *G2P Confidence Feature* is conceptually similar to our *Grapheme Perplexity Feature*. We only compare scores for a word at graphone-level instead of grapheme-level. The steps to detect Anglicisms based on G2P confidence are like in Fig. 3 for the *Grapheme Perplexity Feature*, only replacing the grapheme-level model with a G2P model and PPL with the graphone log probability (*Phonetisaurus* G2P confidence):

1. Preparation: Training of G2P (graphone) models from English and matrix language pronunciation dictionaries
2. Prediction of pronunciation for a word from the test set
3. Comparison of G2P confidence and classification towards the language for which the confidence is better

Test set	Threshold = 0	Optimal threshold	
	F-score	F-score	Threshold
Microsoft-de	70.39%	71.40%	1.0
Spiegel-de	40.56%	45.00%	1.0
NCHLT-af	23.94%	40.23%	10.0

Table 3. Detection performance with different thresholds for the G2P confidence difference

Dictionary size (entries)	Microsoft-de	Spiegel-de
200	21.76%	14.22%
500	46.96%	16.90%
1k	51.02%	19.04%
5k	60.02%	25.82%
10k	64.26%	31.02%
full dict (37k)	70.39%	40.56%

Table 4. Detection performance (f-score) with different dictionary sizes for default threshold of zero

As described we use *Phonetisaurus* for pronunciation prediction and rely on its confidence measure, the negative log probability of the graphone sequence. The G2P confidence of the first-best pronunciation for a word is used, while the generated pronunciation itself is discarded. The feature uses the difference of the G2P confidence for English and the matrix language. We calculate

$$d = \text{G2Pconf}_{matrixlang.}(w) - \text{G2Pconf}_{English}(w)$$

and classify a word w as English if the difference d is greater than zero. We generically assume a threshold of zero, which leads to a simple comparison of which G2P confidence is smaller. Like for the grapheme PPL difference, this is not an optimal choice as shown in Tab. 3. Again we make this trade-off to refrain from supervised training of a better threshold.

For the two German test sets *Microsoft-de* and *Spiegel-de*, the optimal threshold is equally at 1. This value seems to be depending on the dictionary as we only reach a good detection performance from much higher thresholds for the Afrikaans test set. To account for scenarios with low lexical resources in the matrix language, we also evaluated this feature in simulated situations with pronunciation dictionaries containing only a small number of entries. Tab. 4 illustrates the detection performance with different amounts of word-pronunciation pairs to train the G2P model of the matrix language.

4.1.3. Hunspell Lookup Features

Hunspell (`hunspell.sourceforge.net`) is an open source spell-checker and morphological analyzer used in software like OpenOffice. It supports complex compounding and morphological analysis and stemming. The word forms are recognized based on rules defined in the spell-checker dictionary of a language. *Hunspell* spell-checker dictionaries are freely available for more than 60 languages, including English, German and Afrikaans. For our features we used those *Hunspell* resources: The American English dictionary (*en_US*), the "frami" version of the German dictionary (*de_DE-frami*) and the Afrikaans dictionary (*af_ZA*). Our *Hunspell Lookup Features* simply check whether a word is found in the dictionary of the language. The lookup includes an automatic check if the word in question can be derived by the morphological or compound rules in the dictionary. We use two independent features with this concept:

- *English Hunspell Lookup*
  If the word is found or derived from the English dictionary, it is classified as *English*, otherwise as *native*. This feature is

Fig. 4. Entry of German *Wiktionary* containing a paragraph about the word's origin ("Herkunft") and language ("Deutsch" meaning German)

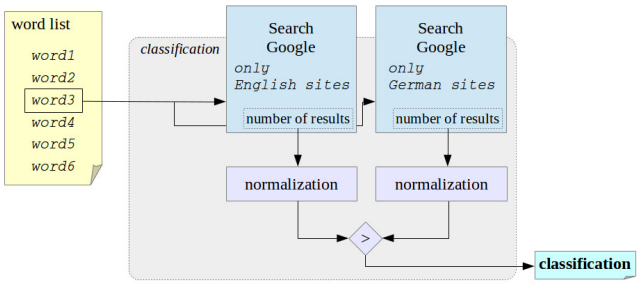

Fig. 5. Classification with the *Google Hits Count Feature*

language independent and can be used without modification for any matrix language.

- *Matrix language Hunspell Lookup*
  The *Matrix Language Hunspell Lookup Feature* does a lookup in the spell-checker dictionary of the matrix language. In this case a word found or derived from the matrix language dictionary is classified as *native*, otherwise as *English*.

The *Matrix Language* and the *English Hunspell Lookup Feature* are independently evaluated. Their classifications can disagree if a word is found in both dictionaries or in neither dictionary. We also experimented with combinations of both features: We only classified a word as *native* if it was in the matrix language dictionary, while not being in the English dictionary. All other words were classified as *English*. However, this did not lead to better results.

4.1.4. Wiktionary Lookup

Wiktionary (`www.wiktionary.org`) is a community-driven online dictionary. Like Wikipedia, the content is written by volunteers. *Wiktionary* is available for over 150 languages but scope and quality in the different languages vary [29]. While the English and French *Wiktionary* each contain more than a million entries, the German *Wiktionary* currently has approximately 355,000 entries and the Afrikaans *Wiktionary* less than 16,000. However, the *Wiktionary* project is growing rapidly which is an advantage for our approach because information about recently introduced words is likely to be added in the future. *Wiktionary* provides a wide range of information. For example, [29] have used *Wiktionary* to extract pronunciations. For most words, *Wiktionary* contains a paragraph about the word's origin. The *Wiktionary* edition of one language does not only contain words from that language. Foreign words including the name of the source language are also added. The snapshot in Fig. 4 shows the Anglicism "downloaden" as a German word ("Deutsch" meaning German) which is originating from English (explained in the section "Herkunft" meaning origin). To detect Anglicisms, we only use the information from the matrix language's *Wiktionary* version. A word is classified as *English* if:

- There is an entry for this word belonging to the matrix language and the origin section contains a keyword indicating English origin.
- There is no entry belonging to the matrix language but an entry marked as "English" in the matrix language's *Wiktionary*.

Unfortunately, entries of the *Wiktionary* versions from different languages do not have a common style and structure. Therefore some language-dependent fine-tuning is necessary. In the German *Wiktionary* we check for the keywords "englisch", "engl.", "Anglizismus" and special *Wiktionary* markups indicating that the word is English. To avoid false positives for loan translations or ancient common origins, we exclude words that contain keywords like "Übersetzung" (translation) and "altenglisch" (Old English) in the origin section. The German *Wiktionary* also contains many conjugations and word forms that are linked to their principal form. We follow such links and classify a word based on the *Wiktionary* entry of its principal form. The Afrikaans *Wiktionary* is not as comprehensive. A section about word origin is not available. Therefore we can only rely on the *Wiktionary* markup indicating that an entry describes an English word. Words that are not found at all in *Wiktionary* are treated as *native* words in our evaluation. When we combine all features (see Section 4.2), we give those words a neutral value. To speed up the procedure and reduce the load on the *Wiktionary* servers, we used a *Wiktionary* dump, which is available to download all content of a language's *Wiktionary*. First we extracted the relevant parts about the words' language and origin from the *Wiktionary*. From this smaller file the actual *Wiktionary Lookup* of the words from our test sets can be done faster. As the German *Wiktionary* also contains many word forms, we have almost 75% coverage of all words from our German test sets. More than half of the annotated Anglicism also have entries in the German *Wiktionary*. In contrast, the Afrikaans *Wiktionary* has very few entries and we could find only 3.45% of the words from our test set.

4.1.5. Google Hits Count Feature

Our *Google Hits Count Feature* is an implementation of the *Search Engine Module* developed by [9]. They use the method to detect English words in a two step approach, first filtering potential English words with a dictionary lookup. Many search engines offer the advanced option to exclusively search on websites of a specific language. Given a correct language identification by the search engine, the assumption is that an English word is more frequently used in English, while a German or Afrikaans word is more frequently used in its language [9]. [30] notes that because current information is dynamically added, this web-based approach also deals well with unknown words like recent borrowings that have not yet been entered into dictionaries. Fig. 5 illustrates the process of the *Google Hits Count Feature*:

1. Search of a word from the test set with search results restricted to English
2. Search of a word from the test set with search results restricted to the matrix language
3. Normalization of the number of search results from (1.) and (2.) with the estimated size of the web in each language
4. Comparison and classification towards the language for which the normalized number of search results is higher

As there is much more English than German content on the Web – and for Afrikaans it is just a fraction –, the raw number of search

Language	Estimated size of web	Ratio to English
English	3,121,434,523,810	
German	184,085,953,431	1 : 17
Afrikaans	6,941,357,100	1 : 450

Table 5. Estimated size of the web in different languages

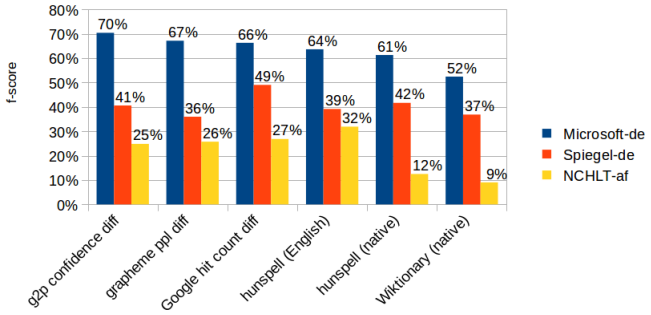

Fig. 6. Anglicism detection performance of all features

results has to be normalized before comparison. The normalized number of hits of a word w returned for the search in each language L is calculated as:

$$\text{hits}_{\text{norm}}(w, L) = \frac{\text{hits}_{\text{absolute}}(w, L)}{\text{web-size}(L)}$$

$\text{hits}_{\text{norm}}(w, \text{'English'})$ and $\text{hits}_{\text{norm}}(w, \text{'matrix language'})$ are compared to classify the word w depending on which normalized score is higher. Following [9], we need the estimated size of the web corpus that is accessible through the search engine in a specific language. This number, web-size(L), is used to normalize the search hits, as shown above. The estimation method was developed by [31]:

1. The frequencies of the 20 most common words are calculated within a large text corpus of the language.
2. The search engine limited to pages of the language is queried for each of these most common words.
3. The number of search hits for each word is divided by its frequency in the training text. The resulting number is an estimate of the total number of search results in that language.

Like [31], we then remove the highest and the lowest estimates as potential outliers. The average of the rest of the estimates is the final estimation of the web corpus size in the language. For English and German we used the most common words and frequencies from [31] and calculated the web corpus sizes based on new *Google* hits counts for these words. For Afrikaans this information was not provided by [31]. Therefore we calculated the 20 most common words and their frequencies from the Afrikaans bible. The normalization based on the bible text resulted in better detection performance than with a normalization on current news articles from `www.rapport.co.za`. Tab. 5 shows our estimations of the total number accessible search results in each language.

4.1.6. Results

Fig. 6 gives an overview of the Anglicism performance for all features. In particular our *G2P Confidence Features* perform well. The large performance gap between the two German domains *Microsoft-de* and *Spiegel-de* occurs throughout all our features. This is caused by the different portions of Anglicisms in the test sets. In the precision measure, which makes up part of the f-score, the portion of false positives is weighted much less on *Microsoft-de* because the features there detect a higher absolute number of Anglicisms (true positives).

Test set	Threshold = 0	Optimal threshold	
	F-score	F-score	Threshold
Microsoft-de	75.44%	75.44%	0
Spiegel-de	56.78%	61.54%	1
NCHLT-af	35.33%	51.66%	4

Table 6. Detection performance of Voting with different thresholds

4.2. Combination of Features

For the combination we experimented with *Voting*, *Decision Tree* and *Support Vector Machine* (SVM) methods.

4.2.1. Voting

To reach a classification based on all features, all Boolean detection hypotheses of the separate features are summed up in a *Voting*:

1. Separate classification by all features of a word from test set
2. Calculation of the sum of all separate classification results
3. Final classification by comparing the vote count to a threshold

We consider a feature classifying the word as *English* with $+1$ and a feature classifying the word w as *native* as -1. An exception is the *Wiktionary Lookup Feature* (see Section 4.1.4): Its contribution in the *Voting* can also be 0 if the word is not found in the native *Wiktionary*. The vote(w) for a word is calculated as:

$$\text{vote}(w) = \text{Classified}_{\text{English}}(w) - \text{Classified}_{\text{native}}(w)$$

The final hypothesis of the *Voting* is based on a threshold T for this vote. With $T > 0$ more than half the features need to vote for a word to be *English*. The threshold was chosen through a 10-fold cross-validation on each test set. Tab. 6 compares the optimal thresholds, which vary for our different test sets. Particularly the detection performance on *NCHLT-af* is significantly improved if some features are not included in the vote. For the final method for Afrikaans, we therefore use a *Voting* without the *Google Hit Counts Feature*.

4.2.2. Decision Tree and Support Vector Machine

We also experimented with *Support Vector Machines* (SVM) with a linear kernel as a powerful state-of-the-art classification method [32] and *Decision Trees* as a common classification method [33]. For those algorithms we used the default parameters of Matlab and trained each test set separately in a 10-fold cross-validation. The information from the single features is given as Boolean input. Like for the *Voting*, the input from a feature classifying the word as *English* is $+1$ and from a feature classifying the word as *native* is -1. The *Wiktionary Lookup Feature* can also be 0 if the word is not found in the native *Wiktionary*. With Boolean features as input for the decision tree, we discard some information. It may help the Anglicism detection if the decision tree receives the "confidence" of each feature's hypothesis. Therefore we additionally experimented with continuous feature input. This improves the detection performance on *Microsoft-de* but deteriorates it on *Spiegel-de*. Consequently, we do not use continuous features in our final combination method.

4.2.3. Results

The performance of the different combination approaches are shown in Fig. 7. For two out of our three test sets our simple *Voting* gives the best overall results – although we only use training data to fine-tune the vote threshold, whereas *Decision Tree* and *SVM* learned more complex relations from the input features. As we did not spend much time on fine-tuning the parameters of the *SVM* some further improvements may be possible. The improvements compared to the best single feature are striking, almost doubling the f-score on the

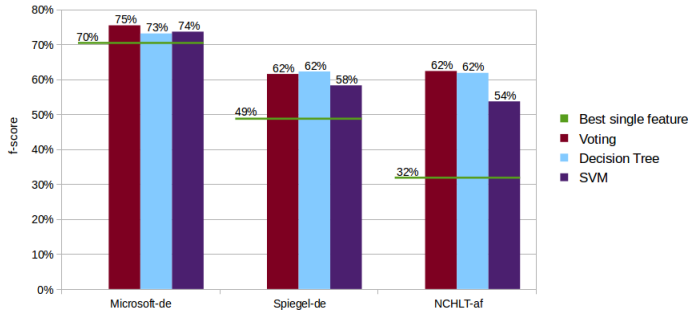

Fig. 7. F-scores of the feature combinations and relative improvement from the best single feature to the best combination

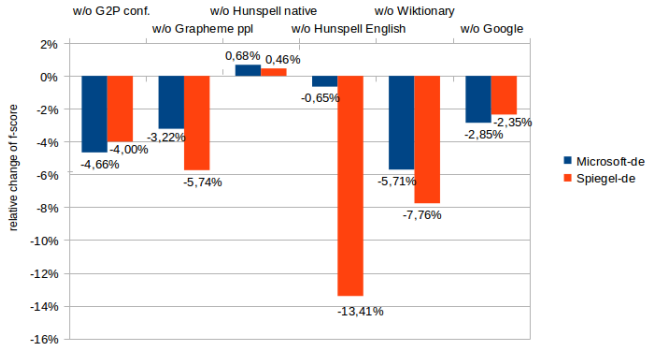

Fig. 8. Relative f-score change of Voting if one feature is left out

NCHLT-af test set. Especially on the Afrikaans test set, for which all our single features had poor detection performances, the combination gives a massive relative improvement of 44.74%. As shown in Fig. 8, the *Wiktionary Lookup* provides important additional information and supports the detection in feature combinations. On both German test sets the *Wiktionary Lookup Features* is an important part of the *Voting*. Apart from the *German Hunspell Lookup*, which gives a very minor improvement if left out, all features contribute to good performance of the *Voting*.

4.3. Challenges: Abbreviations, Hybrid, Other Foreign Words

We have annotated *hybrid English words*, *other foreign words* and *abbreviations* in our German test sets. The classification of these words is somewhat ambiguous because they are either both German and English (*hybrid English words*) or not clearly any of the two (*abbreviations*, *other foreign words*). In oracle experiments we removed these types of words from the test sets before evaluating the f-score. The results show the performance of our features only on the unambiguous test words. Fig. 9 compares the results of our *Voting* when one or all of those word categories are removed from the test set. After manually removing those words, we achieve a relative improvement of up to 47.70%. The varying contribution of the different word categories depends on the composition of the test set. *Spiegel-de* has – relative to the whole test set – more *other foreign words* and less *abbreviations* and *hybrids*. A lot of potential improvement remains in handling these special word categories. We did not experiment with this but word stemming and compound splitting algorithms seem a good way to deal with *hybrid English words*. *abbreviations* might be filtered using regular expressions or an absolute grapheme PPL threshold.

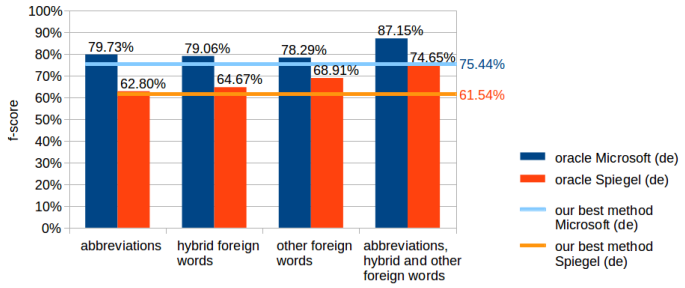

Fig. 9. Performance of Voting after removing difficult word categories from the test sets

	PER
Automatic Anglicism Detection	1.61%
German G2P Model	4.95%
Mixed Language 80:20	4.97%
Mixed Language 50:50	5.46%
English G2P Model	39.66%

Table 7. Phoneme error rates of different approaches to generate pronunciations for the German IT corpus

4.4. Pronunciation Dictionary Generation

We produced pronunciations for the 2,276 words in our German IT corpus based on our Anglicism detection and compared their phoneme error rate (PER) to reference pronunciations. Then we compared the resulting quality to the quality of pronunciations which have been exclusively produced with our German G2P model. Furthermore we compared it to the quality of pronunciations which have been generated with mixed language G2P models trained with different fractions of word-pronunciation pairs from the English *CMUdict* and German *GP-de* dictionary. All pronunciations are mapped to the German phoneme set of *GP-de* before training the G2P model. The *Mixed Language 50:50* model was built with all 40k word-pronunciation pairs from *GP-de* and additional randomly selected 40k word-pronunciation pairs from *CMUdict*. The *Mixed Language 80:20* model was built from all 40k word-pronunciation pairs from *GP-de* and additional randomly selected 10k word-pronunciation pairs from *CMUdict*. This ratio is close to the actual portion of Anglicism contained in the corpus. Our approach to generate pronunciations based on automatic Anglicism detection selects the appropriate G2P model for each word:

- Pronunciations of words detected as *English* are generated by the English G2P model based on *CMUdict*
- Pronunciations of words detected as *not English* are generated by the German G2P model based on *GP-de*

As shown in Tab. 7, our approach to use automatic Anglicism detection produces a far lower PER than generically using a German G2P model. The PER is reduced from 4.95% to 1.61%. The mixed language G2P models produce a PER even slightly higher than the single German G2P model.

5. CONCLUSION AND FUTURE WORK

To detect Anglicisms in text of a matrix language, we developed a set of features and combined those to further improve the performance. For evaluation, we built two German test sets. One from

the IT domain and one from general news articles. We annotated Anglicisms and special word categories in those test sets to allow detailed analyses. Our features are based on grapheme perplexity, G2P confidence, native *Hunspell* lookup, English *Hunspell* lookup, *Wiktionary* lookup, and *Google* hits count. With the *G2P Confidence Feature* we developed an approach which incorporates information from a pronunciation dictionary. This was our most successful single feature. The *Wiktionary Lookup Feature*, leveraging web-derived information, is also a new approach that especially supported the performance of feature combinations. None of our single features rely on text with Anglicisms annotated for training. The features are instead based on other resources like unannotated word lists or dictionaries and are portable to other languages. The combination of the diverse set of features boosted detection performance considerably, especially for the test sets on which the separate features did not bring satisfactory results. A separate handling of the detected Anglicisms from the matrix language words based on our results enhanced our automatic pronunciation dictionary generation process. To develop this approach to Anglicism detection further, we primarily see two areas for future work: (1) Improvement of how *abbreviations*, *hybrid words* and *other foreign words* are handled, and (2) the development of additional independent features. Moreover the effect on ASR and TTS systems should be evaluated further.

6. ACKNOWLEDGEMENT

The authors would like to thank Willem D. Basson for useful comments and the other members of the speech group from the North-West University, South Africa, for providing the NCHLT corpus.

7. REFERENCES

[1] H. Gelas, L. Besacier, and F. Pellegrino, "Developments of Swahili Resources for an Automatic Speech Recognition System," in *SLT-U*, 2012.

[2] B. Kundu and S. Chandra, "Automatic Detection of English Words in Benglish Text," in *IHCI*, 2012.

[3] A. A. Mansikkaniemi and M. Kurimo, "Unsupervised Vocabulary Adaptation for Morph-based Language Models," in *NAACL-HLT*, 2012.

[4] T. Schlippe, M. Volovyk, K. Yurchenko, and T. Schultz, "Rapid Bootstrapping of a Ukrainian Large Vocabulary Continuous Speech Recognition System," in *ICASSP*, 2013.

[5] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," in *ICSLP*, 2002.

[6] J. R. Novak, N. Minematsu, and K. Hirose, "WFST-based Grapheme-to-Phoneme Conversion: Open Source Tools for Alignment, Model-Building and Decoding," in *FSMNLP*, 2012.

[7] B. Ahmed, *Detection of Foreign Words and Names in Written Text*, Ph.D. thesis, Pace University, 2005.

[8] Carnegie Melon University, "The CMU Pronouncing Dictionary," `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`, 2007, accessed on 20.01.2014.

[9] *Automatic Detection of English Inclusions in Mixed-lingual Data with an Application to Parsing*, Ph.D. thesis, University of Edinburgh, 2008.

[10] N. Alewine, E. Janke, R. Sicconi, and P. Sharp, "Systems and Methods for Building a Native Language Phoneme Lexicon Having Native Pronunciations of the Non-Native Words Derived from Non-Native Pronunciations," 2011, US 7472061 B1.

[11] K. S. Jeong, S. H. Myaeng, J. S. Lee, and K.-S. Choi, "Automatic Identification and Back-Transliteration of Foreign Words for Information Retrieval," *Information Processing and Management*, vol. 35, pp. 523–540, 1999.

[12] B. Kang and K. Choi, "Effective Foreign Word Extraction for Korean Information Retrieval," *Information Processing and Management*, vol. 38, 2002.

[13] B. Ahmed, S.-H. Cha, and C. Tappert, "Detection of Foreign Entities in Native Text Using N-gram Based Cumulative Frequency Addition," in *Student/Faculty Research Day, CSIS, Pace University*, 2005.

[14] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a High-Performance Learning Name-finder," in *Conference on Applied Natural Language Processing*, 1997.

[15] D. Klein, J. Smarr, H. Nguyen, and C. Manning, "Named Entity Recognition with Character-Level Models," in *NAACL-HLT*, 2003.

[16] G. Andersen, "Assessing Algorithms for Automatic Extraction of Anglicisms in Norwegian Texts," *Corpus Linguistics*, 2005.

[17] B. Alex, "An Unsupervised System for Identifying English Inclusions in German Text," in *ACL Student Research Workshop*, 2005.

[18] S. Ochs, M. Wölfel, and S. Stüker, "Verbesserung der automatischen Transkription von englischen Wörtern in deutschen Vorlesungen," in *ESSV*, 2008.

[19] S. Ochs, "Verbesserung der automatischen Transkription von englischen Wörtern in deutschen Vorlesungen," Bachelor's thesis (Studienarbeit), KIT, ISL, Germany, 2009.

[20] R. Munro, D. Ler, and J. Patrick, "Meta-learning Orthographic and Contextual Models for Language Independent Named Entity Recognition," in *NAACL-HLT*, 2003.

[21] F. Wolinski, F. Vichot, and B. Dillet, "Automatic Processing of Proper Names in Texts," in *EACL*, 1995.

[22] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[23] T. Schultz, N. T. Vu, and T. Schlippe, "GlobalPhone: A Multilingual Text & Speech Database in 20 Languages," in *ICASSP*, 2013.

[24] *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*, Cambridge University Press, 1999.

[25] C. V. Heerden, M. Davel, and E. Barnard, "The Semi-Automated Creation of Stratified Speech Corpora," in *PRASA*, 2012.

[26] W. Basson and M. Davel, "Category-Based Phoneme-To-Grapheme Transliteration," in *Interspeech*, 2013.

[27] M. Bisani and H. Ney, "Joint-Sequence Models for Grapheme-to-Phoneme Conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[28] H. Engelbrecht and T. Schultz, "Rapid Development of an Afrikaans-English Speech-to-Speech Translator," in *International Workshop of Spoken Language Translation*, 2005.

[29] T. Schlippe, S. Ochs, and T. Schultz, "Web-based Tools and Methods for Rapid Pronunciation Dictionary Creation," *Speech Communication*, vol. 56, pp. 101–118, 2014.

[30] B. Alex, "Comparing Corpus-based to Web-based Lookup Techniques for Automatic English Inclusion Detection," in *LREC*, 2008.

[31] G. Grefenstette and J. Nioche, "Estimation of English and non-English Language Use on the WWW," in *RIAO*, 2000.

[32] I. Steinwart and A. Christmann, *Support Vector Machines*, Information Science and Statistics. 2008.

[33] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.