Native Language Identification using Phonetic Algorithms

Charese Smiley
Indiana University
Bloomington, IN 47405
chsmiley@indiana.edu

Sandra Kübler
Indiana University
Bloomington, IN 47405
skuebler@indiana.edu

Abstract

In this paper, we discuss the results of the IUCL system in the NLI Shared Task 2017. For our system, we explore a variety of phonetic algorithms to generate features for Native Language Identification. These features are contrasted with one of the most successful type of features in NLI, character n-grams. We find that although phonetic features do not perform as well as character n-grams alone, they do increase overall F1 score when used together with character n-grams.

1 Introduction

Native Language Identification (NLI) is the task of automatically predicting the native language (L1) of a speaker given an unlabeled artifact such as a writing sample or speech transcript in a second language (L2). In a typical encounter with a non-native speaker, humans have a variety of contextual clues such as race, approximate age, style of dress, and accent to assist us in making a prediction about the person's native language. However, when predicting L1 relying on features that can be extracted from text alone, we must proceed without the assistance of these visual and acoustic signals. Acoustic cues can be an important source of information since speakers often transfer characteristics of their L1 onto L2. For example, a Japanese L1 speaker may transfer the rigid CV syllable structure onto English and epenthesize vowels into consonant clusters, which may also be reflected in writing. Thus, having phonetic information may prove useful in an NLI classification task. However, we need to make sure that the features we add can be acquired from text and do not contribute to data sparsity. For the IUCL system in the Native Language Identification Shared Task (Malmasi et al., 2017), we began with the premise that acoustic features lost in text are important for language identification and they can be reconstructed using pseudo-auditory features derived from phonetic algorithms that were developed for robust matching in text search. Additionally, we explore a dictionary lookup that provides a phonetic representation of the words in text.

English orthography is rich, complex, and at times idiosyncratic. Using phonetic algorithms, we can reduce some of this complexity by producing a phonetic representation of a word through a series of transformations that map characters and character sequences with similar pronunciation to a single representation such as mapping both <ph> and <f> $\rightarrow$ <f>. To our knowledge, phonetic algorithms have not been explored to generate features for NLI.

2 Related Work

The first NLI Shared Task was part of the 2013 Building Educational Applications (BEA) workshop (Tetreault et al., 2013). Participants received the training portion of the TOEFL11 corpus and were asked to identify the native language of the essay writer from among a closed set of 11 languages available in the corpus. Scoring was based on classification accuracy on an unseen test set in 3 tasks: 1 closed training task where only training data provided in the TOEFL11 corpus could be used, and 2 open training tasks. In the first open training task, researchers could use any training data except for the TOEFL11 corpus. In the second task, they could use any training data *including* the TOEFL11 corpus.

Both character-level and word-level n-grams have featured prominently in past work. Character n-grams ranging from lengths 1-9 have been used (Tetreault et al., 2013). Early work featuring

character bigrams is that of Tsur and Rappoport (2007), which achieved 66% accuracy in 5-way classification. They suggested that character features can serve as a proxy for phonology and that learners' word choices in essays are influenced by their native language. That is, learners gravitate to words in the target language whose phonology matched that of their native language while avoiding words that do not. This tendency can be captured at the character level.

Word-level n-grams have been widely used in a variety of approaches (e.g. Bykh and Meurers, 2012; Jarvis et al., 2013). Traditionally, shorter n-grams with lengths of 1-3 characters are more useful for computational tasks due to the data sparsity that ensues as the length of the n-gram increases. Bykh and Meurers (2012), however, used longest recurring n-grams that appeared in 2 or more essays with good results, perhaps capturing longer collocations and set phrases used by learners from specific L1s. Wong and Dras (2009) and Jarvis et al. (2013) found that both character features and lexical features are effective but classification accuracy deteriorated when both feature types are used together.

Part-of-speech n-grams also feature widely in previous work (Koppel et al., 2005a,b; Wong and Dras, 2009). Lexical n-grams have been shown to outperform POS n-grams for classification accuracy (Bykh and Meurers, 2012). The traditional motivation for the use of POS n-grams is based on the assumption that they abstract away from the confounding effect of essay topic (Koppel et al., 2005a,b; Wong and Dras, 2009). However, POS tag sequences may still be topic dependent. For instance, an opinion piece may contain more personal pronouns than a scientific paper. Brooke and Hirst (2011) suggest that the essay prompts may lend themselves to responses in different registers and the register may manifest itself beyond the lexicon.

Additionally, a number of studies have used syntactic features: context-free grammar (CFG) production rules (Wong and Dras, 2011; Bykh and Meurers, 2014), Tree Substitution Grammar (TSG) fragments (Swanson and Charniak, 2012), and Stanford Dependencies (Malmasi and Cahill, 2015).

3 Data

For the 2017 shared task, similar to the 2013 shared task (Tetreault et al., 2013), the data consists of essays from the same 11 L1s, with the test data drawn from a similar distribution as the original TOEFL11 corpus. In addition to the written text, transcripts of speech and i-vector acoustic features were included in the data release as they have shown promising results for dialect identification (Malmasi et al., 2016; Zampieri et al., 2017). The NLI 2017 shared task contains tracks for essay, speech transcript, and i-vectors alone as well as a fusion task combining all features. The IUCL system focuses exclusively on the essay task.

This dataset consisted of 11 L1s: Arabic (ARA), Chinese (CHI), French (FRE), German (GER), Hindi (HIN), Italian (ITA), Japanese (JPN), Korean (KOR), Spanish (SPA), Telugu (TEL), and Turkish (TUR). There were a total of 11,000 training essays (1,000 for each L1) and 1,100 development essays (100 for each L1). Additionally, the data contained the test taker id, essay prompt, and speech prompt. The distribution of essays by essay prompt for the development set, shown in Figure 1, varies by L1 with Arabic and Korean having the most balanced distribution among prompts and Turkish and Italian having the least.

4 Acoustic Features

Our system utilizes phonetic features for NLI. We explore 3 algorithms that were developed for robust matching: Soundex (section 4.1), Double Metaphone (DMETA) (section 4.2), and the New York State Identification and Intelligence System (NYSIIS) (section 4.3).[1] Soundex relies on simple conversion rules that mostly ignore vowels and groups consonants together by place of articulation in the mouth. The approach abstracts over issues of 1-to-many sound-symbol correspondence in English. For example, the sound /ks/ can be written as both <ks> and <x> as in *tacks* and *tax*. Soundex converts the two spellings to two different representations. In contrast, the NYSIIS and the Metaphone family of algorithms (Philips, 1990, 2000) go a step further to incorporate more of the peculiarities of English spelling, which is necessary for better mappings of homophones.

[1] Features were generated using the Fuzzy library https://pypi.python.org/pypi/Fuzzy

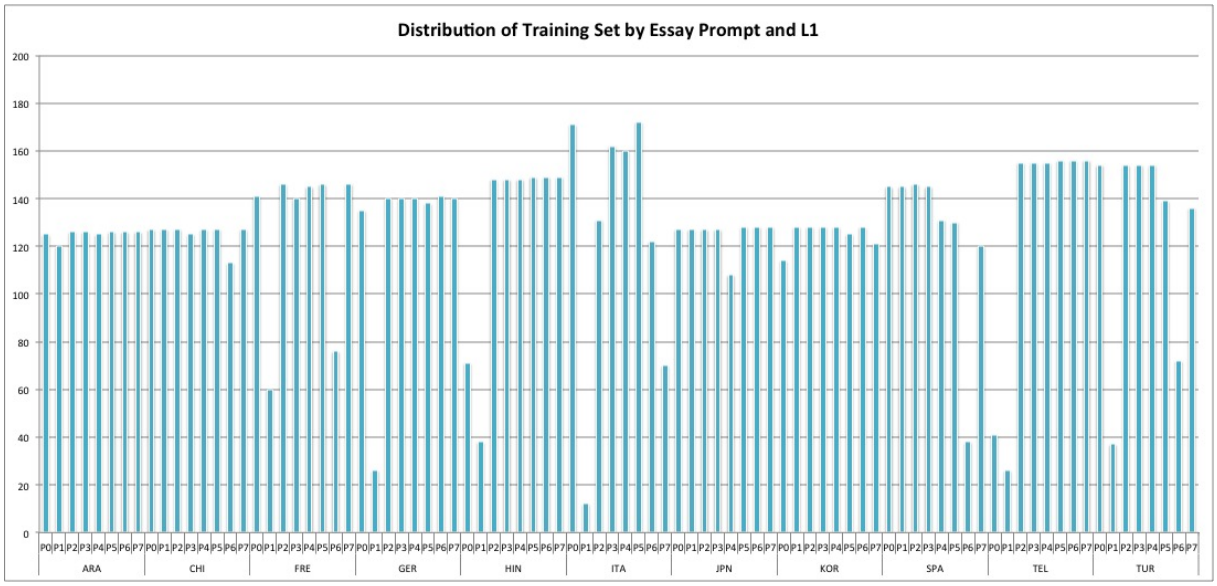

Figure 1: Distribution of data in the development set by number of essays per prompt for each L1.

Thus, all of these algorithms abstract away from specific types of acoustic distinctions, but they differ in which distinctions they ignore.

Finally, we also use the Carnegie Mellon University Pronouncing Dictionary (CMU) (section 4.4) which provides a lookup for the pronunciation of known words. This has the added benefit of providing more accurate mappings than a rule-based converter would and thus a better handling of known words. Moreover, the CMU dictionary can differentiate different vowel sounds where the other algorithms cannot.

4.1 Soundex

Soundex (Knuth, 1973) is an early algorithm first patented in 1918. Under Soundex, the first letter of a word is retained (including all vowels and consonants), all other consonants are mapped to the numbers 1-6, and all other vowels, along with consonants <h>, <w>, and <y>, are dropped. Consonants are converted to numbers within the algorithm as follows: 1: *b, f, p, v*, 2: *c, g, j, k, q, s, x, z*, 3: *d, t*, 4: *l*, 5: *m,n*, and 6: *r*.

This conversion ensures that the consonants are divided roughly along places of articulation with 1 for labials, 2 for coronals and dorsals (excluding<l> and <r>), 3 for dentals, 4 for laterals, 5 for nasals, and 6 for rhotics. Repeated numbers after conversion, such as <mn>, which becomes 55, are reduced to a single number (e.g., 5). All words are normalized to a starting letter plus 3 digits by either omitting any remaining characters for longer words or appending zeros until there are 3 digits for shorter words. Table 1 shows an example sentence from the training set written by a Turkish speaker converted to keys using Soundex and the other algorithms in this paper. One of the the advantages of the Soundex algorithm is that it is easy to implement the small number of rules mapping from letters to numbers. However, since it does not take into account English spelling rules, words that do not sound very similar can end up mapped to the same key. For example, *Cajun* and *Cigna* both map to C250 (Philips, 1990). For our purposes, this means that adaptations of vowels along with adaptations where the place of articulation does not change are not represented in the features.

4.2 Double Metaphone

The original Metaphone phonetic algorithm (Philips, 1990), uses an inventory of 16 consonants, 0BFHJKLMNPRSTWXZ, where 0 stands for /θ/ and *X* for /ʃ/ or /tʃ/. All 21 orthographic English consonants are mapped to these 16, by collapsing some letters like <d> and <t> to <t>. Metaphone contains a number of improvements over Soundex. For example, the letter <c> is sometimes pronounced as /s/ and sometimes as /k/ and the Metaphone algorithm covers many of such cases whereas Soundex does not due to its more simplistic mapping strategy.

Building on the Metaphone algorithm, the Double Metaphone (DMETA) algorithm (Philips,

Algorithm	Key
Original	Furthermore in the past since the mothers were frequently hausewifes, they were able to follow their chidren's education.
Soundex	F636 I500 T000 P230 S520 T000 M362 W600 F625 H212 T000 W600 A140 T000 F400 T600 C365 E323
DMETA	FR0R AN 0 PST SNK 0 M0RS AR FRKN HSFS 0 AR APL T FL 0R XTRN ATKX
NYSIIS	FARTARNAR IN T PAST SANC T MATAR WAR FRAGANTLY HASAF TAY WAR ABL T FAL TAR CADRAN EDACATAN
CMU	FER1DHER0MAO2R IH0N DHAH0 PAE1ST SIH1NS DHAH0 MAH1DHER0Z WER0 FRIY1KWAH0NTLIY0 HHAOSUWAYFS DHEY1 WER0 , EY1BAH0L TUW1 FAA1LOW0 DHEH1R CHIHDRAHNZ EH2JHAH0KEY1SHAH0N

Table 1: Sample sentence represented using various phonetic algorithms

2000) includes many changes and improvements over the original algorithm. Following Soundex, DMETA originally retains the first vowel in words and returns a key with a maximum of 4 letters. Additionally, it collapses all vowels to the letter *A*, and as such the words *Auto* and *Otto*, for example, are mapped to the same key. It also combines the letters <p> and <b>, treats <y> and <w> as vowels (thus eliminating them in post word-initial contexts) and includes a number of modifications to account for spelling influences from foreign words. Finally, the Double Metaphone algorithm returns multiple keys for words that could be pronounced in alternate ways. However, for the use in the IUCL system, we only choose the first key provided since the algorithm favors the most common American pronunciation over other alternatives. As an example, consider Spanish borrowings with <ll> that could be pronounced in an Americanized way as /l/ (e.g. *armadillo, flotilla*) or in the Spanish way as /j/ (e.g. *tortilla, paella*). This is an issue for a rather small number of words, roughly 10% based on Phillips' sample, but should not be of much consequence for our application.[2]

4.3 NYSIIS

Similar to the Double Metaphone algorithm, the NYSIIS algorithm extends Soundex by encoding each letter with consideration for English spelling nuances rather than strict reliance on place of articulation. Unlike the previous two algorithms, NYSIIS maintains the position of vowels within the word but collapses them by converting all vowels to <A>. Also, some versions of NYSIIS maintain the entire key rather than limiting it to the first *N* letters (e.g. exactly 4 letters for Soundex and 1-4 for DMETA). However, word final <s> and <a> are removed. For the IUCL system, we use the non-truncated version of the key. Thus, NYSIIS retains more information in comparison to the previous two conversion algorithms.

4.4 CMU Pronunciations

The Carnegie Mellon University Pronouncing Dictionary (CMU)[3] is an open source dictionary that contains pronunciations for more than 134,000 English words using a set of 39 phonemes and stress markers for vowels. The previous algorithms were designed to minimize the differences between homophones and near homophones. However, since this dictionary was developed for use in automatic speech recognition (ASR) applications rather than text search, the mappings are based on the common pronunciations of American English words. For unknown words, we the LOGIOS Lexicon Tool[4] which generates a CMU pronunciation using letter-to-sound rules. Using this tool, the misspelled word *hausewifes* in Table 1 is converted into the the pronunciation HHAOSUWAYFS.

Soundex, DMETA, and NYSIIS all fall under the category of fuzzy matching algorithms. The original intention for these algorithms were to improve recall when searching for names where the exact spelling is unknown or uncertain and they

[2] There is a newer version of the Metaphone algorithm, Metaphone 3 which is available for commercial use but the details remain unpublished.

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

[4] http://www.speech.cs.cmu.edu/tools/lextool.html

still enjoy widespread use in search applications. The main advantage for using algorithmic phonetic converters such as these is that they can produce a key as output no matter word is given as input. This has an advantage over dictionary-based methods like CMU which, under its pure implementation, fails when a word (such as a proper name or misspelling) falls outside of its internal list, however large. On the other hand, dictionary methods like CMU have the advantage of producing a more nuanced and accurate key for known words. This is especially true when it comes to representation of vowels. All of the other phonetic algorithms mentioned in this paper either collapse all vowels to a common representation or eliminate them in non-word-initial positions whereas in Table 1 we see that the CMU output most closely resembles the original words.

4.5 Experimental Setup

For all experiments, we use the C-Support Vector Classifier implementation in scikit-learn with a linear kernel.[5] For feature values, rather than using a frequency count matrix for features in the document, the TF-IDF score for the term is used instead. TF is the term frequency, i.e., the number of occurrences of a term in a document. IDF is the inverse document frequency, which is calculated by dividing the total number of documents in a corpus by the number of documents containing term t (if t is not equal 0). The application of TF-IDF weighting has been used to good effect in previous research (Gebre et al., 2013).

The benefit of using TF-IDF weighting in this task is that it dampens the effect of terms that are well dispersed throughout the corpus while emphasizing terms that occur less frequently and only within a smaller set of documents. For NLI, TF-IDF weighting is useful for capturing and amplifying the effects of vocabulary choices that are L1-specific. When using binary features, for example, only the presence or absence of a feature is recorded. This effectively weights rare features, such as low frequency words and spelling errors, the same as common features, such as function words. In contrast, TF-IDF gives a measure of the informativeness of a word since a word that appears in many documents will have a lower IDF than one that rarely occurs. Therefore, terms that receive a high TF-IDF score will occur with high frequency in a small number of documents.

All experiments were conducted using character n-grams of length 2-9. Additionally, we restricted the minimum document frequency to 5 documents and the maximum to 5% of documents in the training set.

[5] http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

System	F1 (macro)	Accuracy
Random baseline	0.0909	0.0909
Essay baseline	0.7104	0.7109
Soundex	0.7455	0.7473
CMU	0.7629	0.7627
DMETA	0.7697	0.7727
NYSIIS	0.7830	0.7836
Char†	0.8206	0.8209
Char+CMU	0.8147	0.8145
Char+NYSIIS	0.8190	0.8191
Char+DMETA†	**0.8262**	**0.8264**
Char+Soundex	**0.8300**	**0.8300**

Table 2: Essay track results. Systems marked by † were submitted as part of the official NLI Shared Task. The remaining systems were submitted outside of the official testing phase.

5 Results

We show the results of the different feature sets in Table 2. All approaches perform well above the random baseline of 0.0909 (1/11) and above the shared task baseline of 0.7104 for the 11-way classification task. Results for the single feature runs show that none of the algorithms outperformed basic character n-grams (Char) features, which result in an F1 of 0.8206. Of the phonetic algorithms, NYSIIS shows the highest performance while Soundex and the CMU dictionary approach do not fare well, reaching an F1 of 0.7455 and 0.7697 respectively. This shows that the closest modeling of pronunciation is not helpful for the task. When we combine character-based features with the acoustic features, we observe that all combinations show improvements over the acoustic-only features. Additionally, combining with DMETA and Soundex results in higher performance than character n-grams alone, reaching an F1 of 0.8262 and 0.8300 respectively. The highest performing acoustic-only model, NYSIIS, shows the least improvement, which indicates that the features extracted from this particular conversion are harmful when combined with character n-grams. One

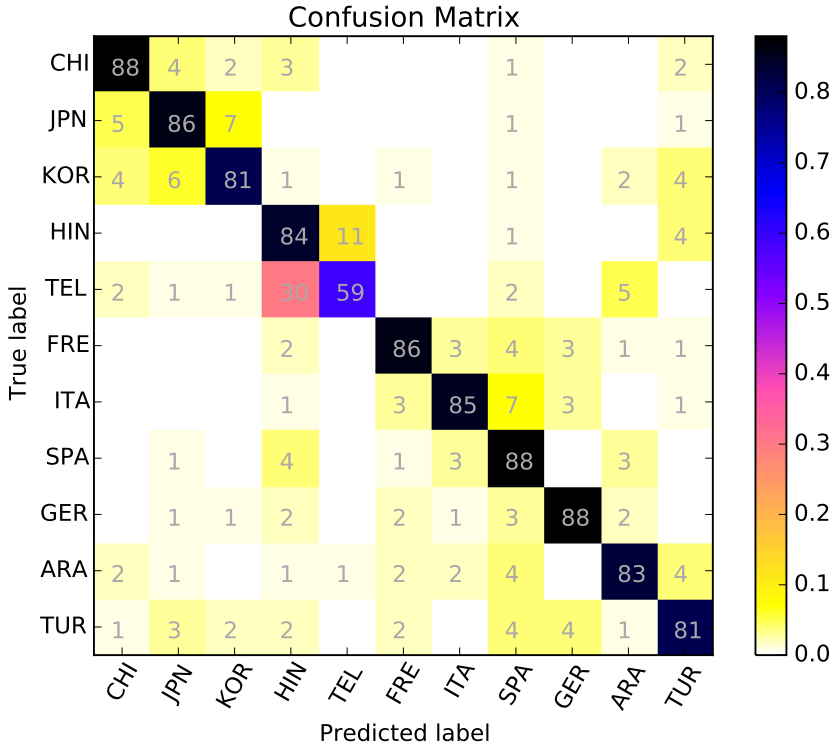

Figure 2: Confusion matrix for best official run, Char+DMETA

possible reason can be found in the higher number of features provided by NYSIIS as opposed to Soundex and DMETA since NYSIIS does not truncate the acoustic representation. However, this requires further investigation.

6 Discussion

Overall, we see that phonetic algorithms do not perform as well as character n-grams for NLI. One reason for this could be that by mapping characters to a simpler representation, important information is lost. For example, one of the most common misspellings for Arabic speakers is *becouse* but it would be rendered the same as *because* by all of the phonetic algorithms except for CMU. Information losses such as this could account for the reduced performance of phonetic algorithms as compared to character n-grams.

On the other hand, the advantage of phonetic algorithms is that commonly used phrases such as *for many reasons* (common among Arabic speakers) can be collapsed into one feature and captured even when minor spelling differences exist (especially if those errors have to do with vowels or consonant sounds that are close in place of articulation).

One of the most informative phrases in for Turkish L1 writers was ATKX SSTM (DMETA) and E323 S235 (Soundex) which translates to 'education system'. Table 3 shows all of the variants of both words found in the training and development data with the variants for Turkish L1 writers shown in italics. Both DMETA and Soundex capture a wide range of variants including spelling errors and various parts-of-speech. As Table 3 demonstrates, Soundex is much more aggressive in combining words that do not sound as similar (e.g. *sixteen, scouting, skidding*) into a single key (S235). Overall, the power of DMETA and Soundex is that used in conjunction with other features, such as character n-grams, these types of features are able to take advantage of longer phrases even when they include spelling errors.

We reach the highest results in the official testing phase of the NLI Shared Task 2017 with the combination of character n-grams with DMETA features, which surpasses the character features by about 0.5% absolute and the DMETA features by about 5.6% absolute. Outside of the testing phase, our best run combined character n-grams with Soundex, surpassing character features by 0.9% absolute and Soundex alone by roughly 8.5% absolute. This shows that the phonetic conversion plus abstraction provides novel information that is

DMETA	education	educationaly, aducation, eeducational, educuation, *education*, educational, edication, educationnal, *educations*, educationally, eduacation, education, ediocation, educationals, *educational*, edecationat
	system	systmatting, *systems*, *systematic*, *systematical*, sistm, systamatically, sistematically, systamatic, *systematically*, syustem, sistem, sistemsm, sistems, *system*, sysytem, systeme, systam, systme
Soundex	education	educationnal, educuation, educaction, *educating*, educaters, edicted, *educations*, edged, etcetera, educaded, edcation, educates, educationalisties, edcucation, eduactional, ethusiastic, *educated*, edecationat, *educaton*, euthusiastic, *educate*, educathion, educationally, eduacation, educoated, educatoion, *education*, ettiquittes, etcetra, educatin, *educational*, edcated, educationaly, ediocation, educative, educaed, edcucate, edication, educat, edcuate, eductional, eductions, eduacte, ethusiastically, educationals, *eductaion*, *educatinal*, edxtra, eduction, eduaction, eeducational, etctec, educatipn, educateted, educatiuonal, educatied, *educators*, *educator*
	system	sixteens, sustaining, sestem, sistuations, seesighting, sstem, systemic, sostinable, *systems*, *systematic*, sixteen, *suggesting*, scitients, schedume, *sugestions*, *systematical*, *sustainable*, skidding, sustained, *sustainablity*, sketing, seggestion, sistematically, systamatic, sustan, sostitution, scation, sustanable, societyhence, sighting, sighteeing, *systematically*, skaiting, sustantiated, sustanining, *sections*, sistem, *succeeding*, sucseedments, sistemsm, sistems, section, *sugestion*, sightings, sustaine, *system*, sistm, sstems, sysytem, sustainment, *suggestions*, *succeding*, systamatically, skating, *sustainability*, *sustain*, sustances, ssudents, *section*, sexteen, systmatting, scating, sostantable, suggetions, sesation, systeme, scouting, systam, systme, suggesstion, syustem, *suggestion*, sistuation, skatting, sixtenn, systen, suceeding, sastained, successding, suggestiong

Table 3: Variants of "education system" in the corpus that are collapsed by DMETA and Soundex. Words in italics are taken from Turkish L1 essays.

not captured in spelling directly.

We had a closer look at the errors that our system makes. Figure 2 shows a confusion matrix for the best setting using character and DMETA features. The table shows that the main weak point of the learner lies in confusing Hindi and Telugu. This is not surprising given the fact that Indians are often multilingual and speak more than two languages. Additionally, English is often used as a lingua franca on the Indian subcontinent with frequent contact from speakers from a variety of L1s resulting in highly similar linguistic patterns.

7 Conclusion and and Future Work

This paper explored NLI using feature sets derived from 3 phonetic algorithms and one dictionary-based lookup. We have shown that our system IUCL can profit from having access to acoustic features in addition to character n-grams. In the future, we plan to further explore variants of phonetic conversions in which we do not abbreviate the words but rather segment them into acoustic n-grams. Will will also explore how features derived from phonetic algorithms can be combined with other lexical and syntactic features.

References

Julian Brooke and Graeme Hirst. 2011. Native language detection with 'cheap' learner corpora. In *Conference of Learner Corpus Research (LCR2011)*. Louvain-la-Neuve, Belgium.

Serhiy Bykh and Detmar Meurers. 2012. Native language identification using recurring n-grams – investigating abstraction and domain dependence. In *Proceedings of COLING 2012*. Mumbai, India, pages 425–440. http://www.aclweb.org/anthology/C12-1027.

Serhiy Bykh and Detmar Meurers. 2014. Exploring Syntactic Features for Native Language Identification: A Variationist Perspective on Feature Encoding and Ensemble Optimization. In *Proceedings*

of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. Dublin, Ireland, pages 1962–1973.

Binyam Gebrekidan Gebre, Marcos Zampieri, Peter Wittenburg, and Tom Heskes. 2013. Improving native language identification with tf-idf weighting. In *the 8th NAACL Workshop on Innovative Use of NLP for Building Educational Applications (BEA8)*. pages 216–223.

Scott Jarvis, Yves Bestgen, and Steve Pepper. 2013. Maximizing Classification Accuracy in Native Language Identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Atlanta, Georgia, pages 111–118.

Donald E Knuth. 1973. *The Art of Computer Programming*, Addison-Wesley, Reading, MA, volume 3, chapter Sorting and Searching.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005a. Automatically determining an anonymous author's native language. In *International Conference on Intelligence and Security Informatics*. Atlanta, GA, pages 41–76.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. 2005b. Determining an author's native language by mining a text for errors. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. Chicago, IL, pages 624–628.

Shervin Malmasi and Aoife Cahill. 2015. Measuring feature diversity in native language identification. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*. Denver, CO, pages 49–55.

Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A Report on the 2017 Native Language Identification Shared Task. In *Proceedings of the 12th Workshop on Building Educational Applications Using NLP*. Copenhagen, Denmark.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the VarDial Workshop*. Osaka, Japan.

Lawrence Philips. 1990. Hanging on the metaphone. *Computer Language* 7(12).

Lawrence Philips. 2000. The double metaphone search algorithm. *C/C++ Users Journal* 18(6):38–43.

Benjamin Swanson and Eugene Charniak. 2012. Native Language Detection with Tree Substitution Grammars. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Jeju Island, Korea, pages 193–197. http://www.aclweb.org/anthology/P12-2038.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A Report on the First Native Language Identification Shared Task. In *Proceedings of the Eighth Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Atlanta, GA, USA.

Oren Tsur and Ari Rappoport. 2007. Using Classifier Features for Studying the Effect of Native Language on the Choice of Written Second Language Words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*. Association for Computational Linguistics, Prague, Czech Republic, pages 9–16. http://www.aclweb.org/anthology/W/W07/W07-0602.

Sze-Meng Jojo Wong and Mark Dras. 2009. Contrastive Analysis and Native Language Identification. In *Proceedings of the Australasian Language Technology Association Workshop 2009*. Sydney, Australia, pages 53–61. http://www.aclweb.org/anthology/U09-1008.

Sze-Meng Jojo Wong and Mark Dras. 2011. Exploiting Parse Structures for Native Language Identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK., pages 1600–1610. http://www.aclweb.org/anthology/D11-1148.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial Evaluation Campaign 2017. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Valencia, Spain, pages 1–15.