OTEANN: Estimating the Transparency of Orthographies with an Artificial Neural Network

Xavier Marjou
xavier.marjou@gmail.com

Abstract

To transcribe spoken language to written medium, most alphabets enable an unambiguous sound-to-letter rule. However, some writing systems have distanced themselves from this simple concept and little work exists on measuring such distance. In this study, we use an Artificial Neural Network (ANN) model to evaluate the transparency between written words and their pronunciation, hence its name Orthographic Transparency Estimation with an ANN (OTEANN). Based on datasets derived from Wikimedia dictionaries, we trained and tested this model to score the percentage of false predictions in phoneme-to-grapheme and grapheme-to-phoneme translation tasks. The scores obtained on 15 orthographies were in line with the estimations of other studies. Interestingly, the model also provided insight into typical mistakes made by learners who only consider the phonemic rule in reading and writing.

***K*eywords** Orthography · Transparency · Multilingual · Artificial Neural Network · Datasets

1 Introduction

An alphabet is a standard set of letters that represent the basic significant sounds of the spoken language it is used to write. When a spelling system (also referred as *orthography*) systematically uses a one-to-one correspondence between its sounds and its letters, the encoding of a sound (also referred as *phoneme*) into a letter (also referred as *grapheme*) leads to a single possibility; similarly the decoding of a letter into a sound leads to a single possibility as well. Such orthography is thus *transparent* with regards to phonemes with the advantage of offering no ambiguity when writing or reading the letters of a word, as illustrated in Figure 1.

In real life, no existing orthography is fully transparent phonemically. One reason is that a word spoken alone is sometimes different from a word spoken in a sentence. An even more consequential reason is that some orthographies like English[1] and French[2] have incorporated deeper depth rules that have moved them away from a transparent orthography [1]; this has created ambiguities when trying to write or read phonemically, as illustrated in Figure 2.

Many studies have discussed the degree of transparency of orthographies [2]. These studies are mainly motivated by the estimation of the ease of reading and writing when learning a new language [3]. Finnish, Korean, Serbo-Croatian and Turkish orthographies are often referred as highly *transparent* [4][5][6][7], whereas English and French orthographies are referred as *opaque* [8]. However, little work exists about measuring the level of transparency of an orthography. One noticeable exception is the work of van den Bosch et al. [8] who have created grapheme-to-phoneme scores and tested them on three orthographies (Dutch, English and French).

This study extends such work with a method called Orthographic Transparency Estimation with an ANN (OTEANN), which models a word-based *phoneme-to-grapheme* task and a word-based *grapheme-to-phoneme* task using an Artificial Neural Network (ANN). For the sake of simplicity, the former task is called a *writing* task while the latter task is called a *reading* task. The goal is not to build a perfect spelling translator or a spell checker. Instead the

[1] https://en.wikipedia.org/wiki/English_orthography#Spelling_patterns
[2] https://fr.wiktionary.org/wiki/Annexe:Prononciation/français

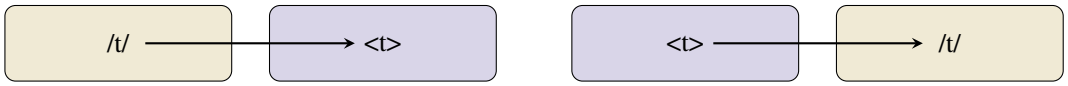

Figure 1: Example of unambiguous correspondence during writing and reading tasks in Esperanto.

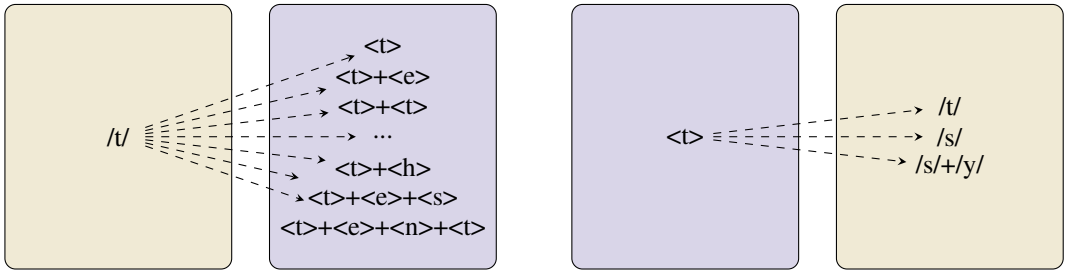

Figure 2: Example of ambiguous correspondence during writing and reading tasks in French. The /t/ phoneme can correspond to multiple graphemes, depending on the nature of the word and also depending on the nature of neighboring words in the sentence or even in a previous sentence. Similarly, the <t> grapheme can correspond to multiple phonemes.

goal is to build a translator which can indicate the degree of phonemic transparency of an orthography, thus allowing to compare several of them.

Interestingly, recent years have seen tremendous progress regarding Natural Language Processing (NLP) with ANNs [9]. In particular, Sutskever et al. [10] have proposed an ANN called a Sequence-to-sequence (seq2seq) model that has proven to be very successful on language translation tasks. Considering writing a word and reading a word as two translations tasks allows re-using the seq2seq model for our work. Of course, some new ANNs with mechanisms such as *attention* [11] or *transformers* [12][13] have recently outperformed seq2seqs. But since we are focusing on short sequences of characters where the translation of each element depends only on its very close neighbours, one seq2seq should be sufficient. Indeed, given that we don't aim at building a perfect spelling translator, we do not have to translate a sequence of words into another sequence of words as in [10]; our seq2seq only requires translating a spoken word into a spelled word (for a writing task) and a spelled word into a spoken word (for a reading task). In other words, while [10] operates at the word level within a sequence of words, our ANN operates at the character level within a sequence of characters. Both the pronunciation and spelling of the word are encoded as a sequence of Unicode characters; a pronounced word is encoded with the characters belonging to the set of phonemes of the target language, whereas a spelled word is encoded with the characters belonging to the alphabet of the target orthography.

Figure 3 illustrates the seq2seq architecture model used by OTEANN. Two other differences with [10] is that OTEANN does not use word embeddings and uses a single layer of Long Short Term Memory (LSTM) [14] units in the seq2seq encoder and decoder. We chose to use a single LSTM layer because our first experiments had shown better results than with multiple LSTM layers; this is likely due to the fact that a phonemic translation does not involve high-level patterns as in word sentences.

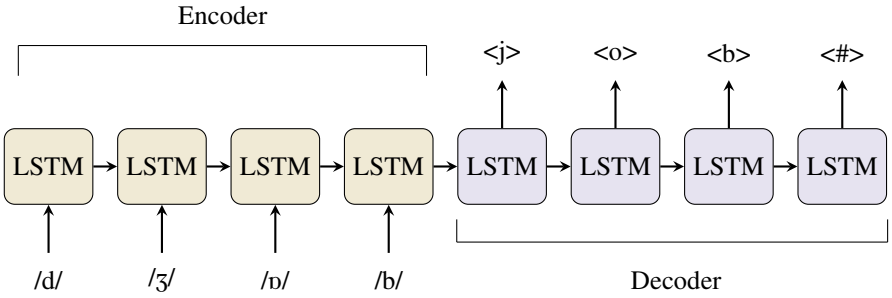

Figure 3: A seq2seq ANN used for a reading task (the input is a spoken word, the target is the spelled word). The hash character acts as a stop character in the decoder.

We used OTEANN to test fifteen orthographies in order to evaluate their degree of phonemic transparency. Fourteen of them are the official orthographies of their respective language (Arabic, Breton, German, English, Spanish, Finnish, French, Italian, Korean, Dutch, Portuguese, Russian, Serbo-Croatian and Turkish) whereas another one is an alternative orthography called Ortofasil proposed for French [3].

For each orthography, two ANN instances were trained independently: one for the writing task and the other for the reading task. Each instance was then tested with new samples, which allowed calculating an average percentage of correct translations. A score of 0% of correct translations represented a fully opaque orthography (no correlation between the input and the target), whereas a score close to 100% represented a fully transparent orthography (full correlation between the input and the target).

Our study first confirms that orthographies like Finnish, Korean and Turkish are highly transparent whereas other ones like French and English are highly opaque. For example, when solely based on a phoneme-grapheme correspondence, we estimated the chances of correctly writing a French word at 5%; similarly, when solely based on a grapheme-phoneme correspondence, we estimated the chances of correctly pronouncing an English word at 18%. For Dutch, English and French reading tasks, our obtained ranking is in line with the one of van den Bosch et al. [8]. One unexpected finding is that OTEANN also allows discovering certain mistakes performed by a new learner during writing and reading.

Remarkably, our method should apply to any other alphabetical orthography, provided a dataset as specified in Section 2.1 is available.

2 Methodology

In order to evaluate the level of transparency of an orthography two main steps were necessary: obtaining datasets and performing the experiments with the ANN.

2.1 Datasets

As displayed in Table 1, a dataset contained only two features per sample: the spelled word and its pronunciation. The spelled word was a sequence of graphemes whereas its pronunciation was a sequence of phonemes. Both were encoded in Unicode characters. The characters representing phonemes are also called International Phonetic Alphabet (IPA) characters. One dataset was needed for each orthography.

Spelled word *(encoded in Unicode characters)*	**Spoken word** *(encoded in Unicode characters)*
job	dʒɒb

Table 1: Dataset features

We also imposed three additional constraints on the needed dataset:

- To have ANN instances comparable in size, we excluded orthographies with more than 100 phonemes or 100 graphemes;
- To obtain the pronunciation feature, the IPA pronunciation of the words had to be either directly available from a dictionary or indirectly available using third-party tools like *phonemizer*[4];
- To have acceptable ANN training performances, we required $10,000$ samples per orthography. Early experiments had shown better scoring results with $20,000$ training samples, but requiring as many training samples would have ruled out certain orthographies such as Arabic, Breton, Dutch or Portuguese.

2.1.1 Baseline Datasets

We first looked for candidates representing a fully transparent orthography and a fully opaque orthography.

Regarding a fully transparent orthography, a promising candidate was Esperanto since its phonemene-to-grapheme correspondence[5] is nearly bijective. Unfortunately, with less than 5000 IPA descriptions, the Esperanto Wiktionary

[3] `https://fr.wikipedia.org/wiki/Systèmes_orthographiques_alternatifs_du_français`
[4] `https://github.com/bootphon/phonemizer`
[5] `https://en.wikipedia.org/wiki/Esperanto_orthography#Alphabet`

did not contain enough samples and was therefore discarded. In order to have a fully transparent baseline, we therefore created a new artificial orthography called Entirely Transparent ('ent') orthography by taking the IPA pronunciation of the English words and mapping each of their phonemes to a single grapheme in order to generate the associated spelled word. Given that OTEANN used the same dataset for writing and reading, each grapheme also corresponded to a single phoneme, which made 'ent' orthography bijective.

Regarding a fully opaque orthography, Chinese orthography, and to a lesser extent Japanese orthography, also appeared as good candidates. Unfortunately, both utilize more than 100 graphemes. We therefore also created a new artificial Entirely Opaque ('eno') orthography by taking the IPA pronunciation of the English words and mapping each of their phonemes to a randomly assigned grapheme, which generated an artificial orthography having no correlation with the word pronunciation.

2.1.2 Studied Datasets

A dataset was created for each of the following languages: Arabic ('ar'), Breton ('br'), German ('de'), English ('en'), Spanish ('es'), Finnish ('fi'), French ('fr'), Italian ('it'), Korean ('ko'), Dutch ('nl'), Portuguese ('pt'), Russian ('ru'), Serbo-Croatian ('sh') and Turkish ('tr').

We incorporated the words from the corresponding Wiktionary[6] dump[7], with the exception of the following ones:

- Words containing space characters;
- Words containing more than 25 characters;
- Words containing capital letters (except for German words);
- Words containing non-standard characters with regard to the orthography's alphabet.

Two languages required additional processing:

- For German, proper nouns were discarded and the capital letter of common nouns was transformed into lower case;
- For Korean, the syllabic blocks words were converted in a series of two or three letters (one vowel and one or two consonants) pertaining to the Korean alphabet with *ko_pron*[8] Python library.

Regarding pronunciation, we directly extracted the IPA pronunciation when available in the associated Wiktionary dump, which was the case for 'br', 'de', 'en', 'es', 'fr', 'it', 'nl' and 'sh'. For Portuguese ('pt'), the IPA pronunciation was extracted from the English Wiktionary dump as there was no IPA is the Portuguese Wiktionary. For the others ('ar', 'ko', 'ru', 'fi', 'tr'), we had to derive it from the spelled word with additional software. For Russian, the Russian Wiktionary dump did not contain the IPA. We thus used *wikt2pron ru_pron* module[9] to obtain a pronunciation similar to the one displayed in the Russian Wiktionary web pages. For all orthographies, we discarded the words whose pronunciation contained an IPA symbol encountered less than 100 times throughout the whole dictionary.

Extracting the phonemic pronunciation from Wiktionary may raise concerns given than IPA symbols can be used both for phonetic and phonemic notations and that there is no unified consistency between the different dictionaries. When processing the IPA strings, we nonetheless took care of preserving the highest surface pronunciation as possible: most pitches were removed since they represent no useful hint during the writing task (i.e. no consequence on the spelled word) and especially since they are generally impossible to predict when translating the spelled word into a pronunciation during the reading task. Nevertheless the /ː/ pitch was noticed as indispensable for some orthographies, for instance for predicting double vowels in the spelling of Finnish words or the *alif* letter in Arabic. Regarding the /ˈ/ pitch, it can slightly influence Spanish translation scores: it can lead to a better writing score as it can be a hint for predicting accented letters, but it can also lead to a lower reading score.

Another interesting candidate was a proposal of a new orthography for French that is known as French Ortofasil ('fro')[10]. Although not being entirely bijective (e.g. both /o/ and /ɔ/ map to <o> letter), it seemed highly transparent. We therefore used it to generate a dataset of the 'fro' orthography.

Table 2 summarizes the datasets obtained.

[6] https://wiktionary.org
[7] https://dumps.wikimedia.org/
[8] https://pypi.org/project/ko-pron
[9] https://wikt2pron.readthedocs.io/en/latest/_modules/IPA/ru_pron.html
[10] http://fonétik.fr/v0/faq-en.html#mapping-table

Orthography	Samples	Phonemes	Graphemes	Mean Nb. of Phonemes	Mean Nb of Graphemes
ar	12,058	32	48	8.0	8.9
br	17,343	45	29	6.7	7.5
de	529,742	41	30	10.2	11.6
en	42,217	50	29	7.4	7.6
es	40,830	35	33	8.2	8.7
fi	105,353	27	27	10.4	10.4
fr	1,214,262	35	41	9.0	11.3
fro	1,214,262	35	32	9.0	8.6
it	26,798	34	32	9.2	9.1
ko	64,669	41	67	10.6	8.3
nl	13,344	45	28	7.8	8.6
pt	13,722	43	36	7.0	7.2
ru	304,525	30	33	10.6	10.7
sh	98,576	40	27	9.1	8.9
tr	117,850	36	34	10.2	10.2
eno	42,217	42	42	7.4	7.4
ent	42,217	42	50	7.4	7.4

Table 2: Summary of the datasets. For each dataset, a line indicates the number of samples available, the number of different phonemes, the number of different graphemes, the mean number of phonemes in words, and the mean number of graphemes in words.

2.2 Experiments

2.2.1 ANN architecture

We used a seq2seq ANN similar to Sutskever et al. [10]. However, our seq2seq model was configured at the character level and not at the word level given that our translation tasks operate at the word level and not at the sentence level. Our model consisted of two LSTM stacks: one in the seq2seq encoder and another in the seq2seq decoder. Each stack contained 512 units. During the first experiments, we tried more LSTM stacks but they led to less good results, probably because a phonemic translation only depends on few consecutive characters (generally less than three characters), which does not require any hierarchical patterns. In the end, all of our ANN instances resulted in approximately $5,000$ units and $8,000,000$ parameters.

The model was built on top of TensorFlow[11]. The ANN code is available on Github[12]. As the model contained LSTMs, a GPU support offered significant time savings. For instance, training OTEANN on $10,000$ samples ($12,500$ samples with a TensorFlow-Keras training validation split of 0.2 and 59 epochs) for the 'en' writing task took one hour and forty-one minutes with our XLA_CPU (i7-7700HQ CPU @ 2.80GHz-16GB on a MacBook) whereas it took only five minutes and four seconds (i.e. 20 times faster) with an XLA_GPU (Tesla P100-PCIE-16GB on Google Colab).

2.2.2 Performance metric

As we were not aware of similar work, we created a simple score in order to assess the performance of the ANN prediction during the testing step. When all the predicted characters were equal to those of the true target, a prediction was considered successful, which allowed calculating a mean score indicating the percentage of successful predictions at the end of the test step.

2.2.3 Orthographic test

For each dataset, we then trained the ANN to try to make it learn how to spell a word based on its sole pronunciation, and conversely to learn how to pronounce a word based on it sole spelling. For each task of each orthography, a new ANN instance was thus trained with $10,000$ samples. For a writing task, the input was the word pronunciation while the target was its associated spelling, as depicted in Figure 3. Conversely, for a reading task, the input was the spelled word while the target was its associated pronunciation. After training, the ANN was evaluated with $1,000$ new samples. Interestingly, future work may use more test samples to gain a statistical insight on the different types of errors depending on the orthography at hand.

[11] `https://www.tensorflow.org/`

[12] `https://github.com/marxav/oteann`

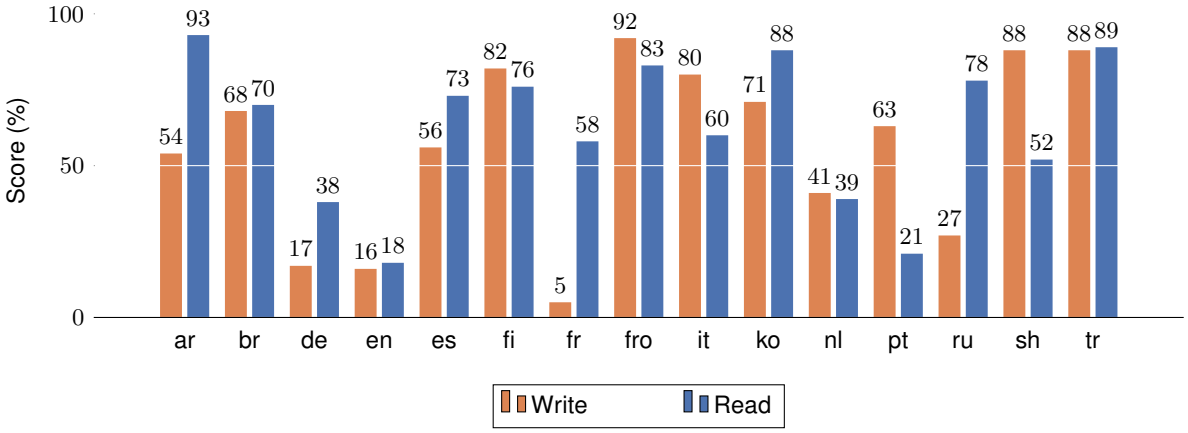

Figure 4: Phonemic transparency scores of 15 orthographies.

3 Results

First, with regards to the results of the baseline orthographies, the 'eno' opaque orthography obtained a score of 0% in both writing and reading, which was in line with the expectations given that there was no correlation between its phonemes and its graphemes. On the other hand the 'ent' transparent orthography obtained a score of 92% in writing and 92% in reading, which indicated a high level of correlation between its phonemes and its graphemes. The use of $20,000$ training samples would have further increased the scores by about 5%, but unfortunately not all the Wiktionaries have allowed us to derive more than $10,000$ training samples. Although future versions of this work may improve the ANN performance with a more complex model to allow obtaining a score closer to 100%, we considered our ANN satisfactory for our objective of comparing the performance of different orthographies.

Figure 4 presents our main results. They are notably different between writing and reading given these task are generally not symmetrical. Three features are likely to influence the symmetry, and therefore the efficiency of the two tasks. As recalled by Figure 2, the most important feature would undoubtedly be the number of possible phoneme-to-grapheme and grapheme-to-phoneme ambiguities per tested orthography. Unfortunately we did not possess such data. Another feature that might influence the performance of an ANN task is the mean number of characters in the target sequence with respect to the input sequence; indeed, the ANN is likely to have more difficulties predicting a long sequence of characters than a short one. At last, another impacting feature may be the number of possible values (graphemes or phonemes) for a given target character. Typically the higher the number of values, the harder the prediction is for the ANN. Future work should investigate the relative importance of these features on the OTEANN performances.

Comparing our reading results of OTEANN with those of van den Bosch et al. [8], OTEANN first seems to naturally assimilate the grapheme complexity (e.g. for French, it successfully learnt that "cadeau" should be pronounced /kado/). Regarding grapheme-to-phoneme complexity (*G-P complexity*), they ranked English (*G-P complexity*=90%) more complex than Dutch (*G-P complexity*=25%) which, in turn, was more complex than French (*G-P complexity*=15%). OTEANN results preserved the same ranking with transparency scores of 18%, 39% and 58% for English, Dutch and French. Admittedly, OTEANN's scores were different in terms of scale but OTEANN had to deal with more orthographies as well as with the writing task.

Analyzing the detailed results, we can also investigate which phonemic correspondences were typically learnt or not learnt by an ANN. For instance, when testing reading, the Italian ANN successfully predicted that the word "cerchia" should be pronounced /ʧerkja/ , hence having successfully learned that <c>, when followed by <e>, should be pronounced as /ʧ/ and also that <c>, when followed by <h>, should be pronounced as /k/. On the other hand, the French ANN unsuccessfully predicted that /bɔ̃ʒuʁ/ should be written as "bongorre" (instead of "bonjour"), showing that it did not learn that /ʒ/, when followed by /u/, should be written as <j>.

Finally, Figure 4 also allows categorizing the studied orthographies with respect to their degree of transparency:

- **Finnish, Korean and Turkish**: Their scores above 70% both in writing and reading confirmed that their orthography is highly transparent as indicated in [4], [5] and [7].

- **Arabic, Breton, Italian, Serbo-Croatian and Spanish**: With all their scores above 50% their orthography was also measured as fairly transparent. Arabic and Spanish performed slightly less well in the phoneme-to-grapheme direction whereas Italian and Serbo-Croatian performed slightly less well in the other direction. For Spanish, the detailed results showed that the most common failure during writing occurs with accents: the ANN had great difficulty predicting whether a vowel should contain an accent or not. For Italian, typical errors observed in the results were the prediction of /ɛ/ instead of a /e/ and /ɔ/ instead of a /o/, which were minor mistakes. Future work may revise the scoring formula to reduce the cost of such errors in the performance calculation.

- **Dutch, Portuguese and Russian**: With average scores close to 50%, the results tended to suggest that Dutch would be moderately transparent in writing and reading; that Portuguese would offer some fair level transparency in writing but little transparency in reading; and the opposite for Russian, which would confirm Kerek and Niemi [15] who have declared that Russian orthography is a *"combination of complexity and regularity"*.

- **German**: With all scores below 50%, the results are surprisingly low since German orthography is generally considered consistent. The detailed results for German showed that the ANN generally failed to predict correct spelling during the writing task. The most common error was not to guess if a /t/ should be written as a single <t> or as a double <t>, which is a understandable since many German words use a double consonant. Future work is needed to explain whether silent letters alone can explain these low scores or if there are other reasons. In particular, Table 1 shows that German words generally have more characters than words of other orthographies, which might be a disadvantage since the seq2seq has a longer sequence of characters to predict, which increases the probability of failure.

- **French**: With the lowest writing score (5%), the results showed that the chances of correctly writing a French word on the sole basis of its pronunciation were rare, as anticipated given the high number of phoneme-to-grapheme possibilities. Without being able to access a broader context than the word itself, the ANN was not able to reliably predict how to write a French word. With a higher reading score (58%), the ANN obtained better reading results but remained far from a transparent orthography. As a comparison, the 'fro' orthography, which uses 23% less graphemes than 'fr', obtained excellent scores of 92% and 83% (the differences in score between writing and reading are again due to the fact that the 'fro' orthography is not bijective. For instance, the <o> letter can be translated into /o/ or /ɔ/).

- **English**: With a low writing score (16%) and the lowest reading score (18%), the results showed that English orthography is also highly opaque, which is consistent with most studies. As a reminder, a phonemic reading of an English word often does not work because of its high number of grapheme-to-phoneme possibilities. For instance the grapheme <u> can either correspond to /ʌ/ (as in "hug"), to /juː/ (as in "huge"), to /ɜːr/ (as in "cur") or /jʊəː/ as in "cure".

4 Discussion and Conclusion

Our OTEANN method showed that a seq2seq ANN can convincingly estimate a level of phonemic transparency for multiple orthographies both for the phoneme-to-grapheme and grapheme-to-phoneme directions. The results for Dutch, English and French orthographies reasonably extended those of van den Bosch et al.[8] while the other results reflected the perception of several other studies.

This method should be easily applicable to other orthographies beyond those tested in this study. A drawback is that, by its nature, our ANN requires $10,000$ to $20,000$ words for a decent training, which is a lot. Ideally, the new studies will follow with a more efficient mechanism requiring only the first top $1,000$ words of a language or even less. In the meantime, for a better ANN training, and thus more accurate scores, some datasets would require more samples. It would therefore be desirable that each Wiktionary page systematically contain the IPA description of its word even if the pronunciation is considered obvious. In addition, since the superfluous IPA symbols slightly influence the score results, future work should closely examine and discuss the phonemes to use depending on the orthography to be tested.

As OTEANN also points out some possible grapheme or phoneme errors when writing or reading phonemically, it could also be used to detect errors in the Wiktionary pages of transparent orthographies or also to evaluate improvement proposals for opaque orthographies.

Another interpretation of our results is that our ANN and its only 5,000 artificial neural units somehow imitate the way a beginner learns to write and read a language, suggesting that a transparent orthography would be easier to learn.

5 Acknowledgements

We thank all Wiktionary contributors for their tremendous lexicographic work, the LazyProgrammer for its deep learning courses and its inspiring slogan "all data is the same", the Keras web site for its blog and documentation, as well as Google Colab for its Jupyter and GPU support.

References

[1] Philip HK Seymour, Mikko Aro, Jane M Erskine, and Collaboration with COST Action A8 Network. Foundation literacy acquisition in european orthographies. *British Journal of psychology*, 94(2):143–174, 2003.

[2] Elisabeth Borleffs, Ben AM Maassen, Heikki Lyytinen, and Frans Zwarts. Measuring orthographic transparency and morphological-syllabic complexity in alphabetic orthographies: a narrative review. *Reading and writing*, 30(8):1617–1638, 2017.

[3] Sylvia Defior, Francisco Martos, and Luz Cary. Differences in reading acquisition development in two shallow orthographies: Portuguese and spanish. *Applied Psycholinguistics*, 23(1):135–148, 2002.

[4] Mikko Aro. *Learning to read: The effect of orthography*. Number 237. Jyväskylän yliopisto, 2004.

[5] Yu-Chun Wang and Richard Tzong-Han Tsai. Rule-based korean grapheme to phoneme conversion using sound patterns. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2*, pages 843–850, 2009.

[6] MT Turvey, Laurie B Feldman, and G Lukatela++. The serbo-croatian orthography constrains the reader to a phonologically analytic strategy. *Status Report on Speech Research: A Report on the*, page 17, 1984.

[7] Banu Öney and Aydin Yücesan Durgunoğlu. Beginning to read in turkish: A phonologically transparent orthography. *Applied psycholinguistics*, 18(1):1–15, 1997.

[8] Antal van den Bosch, Alain Content, Walter Daelemans, and Beatrice de Gelder. Analysing orthographic depth of different languages using data-oriented algorithms.

[9] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning in natural language processing. *arXiv preprint arXiv:1807.10854*, 2018.

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[12] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI, 2018.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[15] Eugenia Kerek and Pekka Niemi. Russian orthography and learning to read. *Reading in a Foreign Language*, 21(1):1–21, 2009.