Automatic Syllabification of English Orthographic Forms

Annika Shankwitz

Indiana University

Correspondence: ashankwi@iu.edu

Abstract

Automatic syllabification refers to the task of adding syllable boundaries to orthographic or phonetic forms. While there are many ways to approach this task, Dinu et al. 2024 liken it to a sequence labeling task. To do so, they label graphemes as either beginning, or not beginning, a syllable, implement a bidirectional gated recurrent unit model to perform classification, and achieve an impressive overall accuracy of 99.74% and F1-score of .9969 when syllabifying Italian orthographic forms. Italian, however, has a shallow orthography and simple syllable structure. English, by contrast, has a deep orthography and complex syllable structure (Seymour et al., 2003). The goal of the current project, then, was to determine how Dinu et al. 2024's approach performs on English. Ultimately, it was found that Dinu et al. 2024's approach performs well on English, achieving a comparable overall accuracy of 96.51% and F1-score of .9612.

1 Introduction

Automatic syllabification, as the name suggests, is the task of adding syllable boundaries to forms (either orthographic or phonetic) automatically. This project will focus specifically on the automatic syllabification of English orthographic forms, that is, on adding syllable boundaries to written English words. For example, given the word 'about', the desired syllabification is 'a-bout', where '-' indicates a syllable boundary.

Syllabified corpora can be helpful for a number of natural language processing tasks. One such task is low-resource language modeling, where syllable-based tokenization results in lower perplexity than character or sub-word tokenization (Oncevay et al., 2022). Another application is in the domain of automatic speech recognition, where syllable-based models have been found to outperform phoneme-based models for Mandarin Chinese (Zhou et al., 2018) and Indian languages (Anoop and Ramakrishnan, 2023). It is thus desirable to be able to syllabify corpora automatically.

There are generally two approaches to automatic syllabification: those which are rule-based and those which are data-driven (Marchand et al., 2009). In rule-based approaches, some theoretical position on the syllable specifies syllabification. For example, Hammond 1997 proposes and implements an optimality-theory-based syllable parser for English orthographic forms.

In data-driven approaches, the notion of a syllable is learned from syllabified training data. While a number of different data-driven methods have been applied to automatic syllabification (Adsett and Marchand, 2009), more recent work focuses specifically on the use of neural models. For example, Krantz et al. 2019 successfully syllabify English, Dutch, Italian, French, Manipuri, and Basque phonetic forms using a model consisting of "bidirectional long short-term memory cells, a convolutional component, and a conditional random field output layer" (p.1). Syllabification of phonetic forms, however, is an easier task than syllabifying orthographic forms (Marchand et al., 2009).

Despite this, Dinu et al. 2024 achieve an impressive overall accuracy of 99.74% and F1-score of .9969 when syllabifying Italian orthographic forms using a bidirectional gated recurrent unit (GRU) model. In order to use said model, Dinu et al. 2024 frame automatic syllabification as a sequence labeling task. More explicitly, graphemes within an inputted word are labeled with either 0, indicating that the grapheme doesn't begin a syllable, or 1, indicating that the grapheme does begin a syllable. To illustrate, 'about' should be labeled as 11000. To the best of my knowledge, Dinu et al. 2024's approach has yet to be applied to other languages. Herein lies the goal of the present project.

English and Italian differ from one another in both orthography and syllable structure. The com-

plexity of orthographies is described in terms of depth: shallow orthographies are simpler, and consist of mainly one-to-one mappings between graphemes and phonemes, while deep orthographies are more complex, and consist of mainly many-to-one or one-to-many mappings between graphemes and phonemes. Italian's orthography is shallow; English's orthography is deep (Seymour et al., 2003).

Similarly, Italian has a simple syllable structure, a syllable structure consisting of mainly open syllables with simple onsets (e.g., CV), while English has a complex syllable structure, a syllable structure consisting of open and closed syllables, with simple and complex, onsets and codas (e.g., CV, CVC, CCVCC, etc.) (Seymour et al., 2003).

Considering these differences, it is possible that syllabifying Italian orthographic forms is an easier task than syllabifying English orthographic forms. As such, this project seeks to answer the follow question: *How does Dinu et al. 2024's approach perform on English*?

2 Methods

This section presents the project's methodology[1]. The dataset used is described in section 2.1, and the models implemented in section 2.2.

2.1 Data

A corpus of 24,411 syllabified English words acts as this project's dataset[2] (Solheim, 2015). To illustrate this dataset, the first three items are as follows: *a;back*, *a;ban;don*, *a;ban;don;ment*, where semicolons indicate syllable boundaries. The break down of the dataset by syllable number can be seen in Table 1[3].

In terms of data preprocessing, the numeric equivalent of each word was found with regular expressions, and was paired with its orthographic counterpart. The resulting pairs were then split randomly into the training set (80%, or 19,527 pairs), the development set (10%, or 2,442 pairs), and the testing set (10%, or 2,441 pairs).

[1]The GitHub repository housing this project can be found here: https://github.com/its-Annika/SyllabifyingEnglish

[2]The dataset can be accessed via the following link: https://github.com/gautesolheim/25000-syllabified-words-list

[3]The word 'nan' was removed from the dataset, as its similarity to NaN (not a number), posed difficulty for the code.

Syllable #	Total Words
1	4,431
2	9,130
3	5,987
4	3,266
5	1,269
6	283
7	40
8	4

Table 1: Breakdown of Syllabified Dataset by Syllable Number

2.2 Models

Two models were implemented via PyTorch (Paszke et al., 2019). The first is a simple Elman recurrent neural network model (Elman Model), which consists of:

- a character embedding layer, producing 96-dimension vectors for each inputted character,
- an Elman RNN cell with an 192-dimension hidden state,
- and a final linear layer which produces tag scores for each character.

The second model (GRU Model) is a near replication of Dinu et al. 2024's model (Dinu Model)[4]. It consists of:

- a character embedding layer, producing 96-dimension vectors for each inputted character,
- a stacked bidirectional GRU with 3 layers, an 192-dimension hidden state, and .2-rate dropout between GRU layers,
- .5-rate dropout applied to the GRU output,
- layer normalization applied to the GRU output,
- a time-distributed, fully-connected linear layer with ReLU activation, which projects each time step onto the tag set,
- and a final linear layer which produces tag scores for each character.

Both models were trained for 15 epochs with a learning rate of .001, using Cross Entropy Loss with the Adam optimizer (Kingma and Ba, 2017).

[4]The GRU model differs from the Dinu model in a few regards. The Dinu model 1) employed a learning rate scheduler which halved the learning rate every 5 epochs, 2) computed both cross-entropy loss and root mean squared error, and 3) utilized the following parameters: embedding dimension=64, GRU hidden dimension=128, learning rate=0.0003, number of training epochs=10-15.

Additionally, both model's hyper-parameters were found via a manual grid search loop targeting:

- Embedding dimension = [32, 64, 96]
- Hidden dimension = [96, 128, 192][5]
- Learning rate = [.01, .001, .0001]
- Number of training epochs = [5, 10, 15]

3 Results

Ultimately, the Elman model achieved 89.49% overall accuracy, and the GRU model achieved 96.51% overall accuracy. Additional performance metrics can be seen in Table 2. Finally, the performance of the Elman model by syllable number is presented in Table 3, and the performance of the GRU model by syllable number appears in Table 4. It should be noted that all metrics were performed at the grapheme-level, that is on the labels of individual graphemes. Additionally, it is important to specify that automatic syllabification is an imbalanced classification problem; in the current testing set, roughly 66% of graphemes (n=12,027) did not begin syllables, compared to the 33% of graphemes (n=6,167) which did begin syllables.

	Elman Model			
	Prec	Rec	F1	Support
Beginning	.8697	.8116	.8396	6167
Not Beginning	.9066	.9376	.9219	12027
Macro-Avg	.8881	.8746	.8807	18194
	GRU Model			
	Prec	Rec	F1	Support
Beginning	.9437	.9539	.9488	6167
Not Beginning	.9763	.9708	.9735	12027
Macro-Avg	.9600	.9624	.9612	18194

Table 2: Model Performance (Beginning = 1, Not Beginning = 0)

Overall, the GRU model solidly outperforms the Elman model, as evidenced by the GRU model's macro F1-score of .9612, as compared to the Elman model's macro F1-score of .8807. Additionally, both models perform better on the majority class, not beginning, over the minority class, beginning. This performance difference is more pronounced in the Elman model than in the GRU model.

Turning now to the results presented in Tables 3 and 4, both models perform best on extreme cases (1 and 8 syllable words), but largely differ in performance otherwise. By macro F1-scores, the performance ranking of the Elman model by syllable is 8 > 1 > 2 > 3 > 6 > 7 > 5 > 4, and the performance ranking of the GRU model by syllable is 7, 8 > 1 > 5 > 6 > 4 > 3 > 2.

Syllable #	Total Words	Prec	Rec	F1
1	478	.8746	.9495	.9017
2	874	.8969	.8905	.8936
3	610	.8950	.8716	.8813
4	327	.8713	.8474	.8558
5	117	.8729	.8496	.8570
6	28	.8888	.8673	.8745
7	6	.8807	.8690	.8733
8	1	.9615	.9375	.9467

Table 3: Elman Model Performance by Syllable Number (Macro-Averages)

Syllable #	Total Words	Prec	Rec	F1
1	478	.9858	.9956	.9906
2	874	.9508	.9542	.9525
3	610	.9550	.9573	.9561
4	327	.9579	.9586	.9582
5	117	.9732	.9741	.9737
6	28	.9672	.9664	.9668
7	6	1.00	1.00	1.00
8	1	1.00	1.00	1.00

Table 4: GRU Model Performance by Syllable Number (Macro-Averages)

4 Discussion

This section first discusses the performance of the current models on English as compared to the performance of Dinu et al. 2024's model on Italian, then considers the errors made by the Elman and GRU models, and finally outlines possible limitations in the current limitation.

4.1 Comparison to Italian Results

The performance of the Dinu Model, the Elman model, and the GRU model can be seen in Table 5. While the Elman model does not approach the performance level of the Dinu model, the GRU model performs comparably, as indicated by the relatively small difference between their performance metrics[6]. Also, even without comparison, the GRU

[5]This parameter is multiplied by two in the GRU model. More explicitly, hidden dimension=96 'won' the GRU grid search, but was doubled (96 dimensions for the forward pass, and 96 dimensions for the backward pass) in the model's actual implementation.

[6]The GRU model and Dinu model's accuracy are separated by 3.23% and the macro-F1 scores are separated by .0357.

model's overall accuracy of 96.51% and F1-score of .9612 are noteworthy. As such, it can be said that Dinu et al. 2024's approach performs well on English.

Additionally, it appears that differences in orthographic depth and syllable structure complexity don't greatly impact model performance; Italian and English are nearly opposite in these respects, yet the same approach performs comparably on both. On one hand, this suggests that syllabifying English is not a much more difficult task than syllabifying Italian. On the other hand, it's possible that syllabifying English *is* more difficult than syllabifying Italian, but that the power of the implemented neural model renders this difference irrelevant.

	Dinu Model	Elman Model	GRU Model
Overall Accuracy	99.74	89.49	96.51
F1-Score	.9969	.8807	.9612

Table 5: Model Comparison

4.2 Error Analysis

For the purpose of error analysis, four types of errors were defined. They are as follows:

- Early: A syllable beginning[7] was marked on the grapheme preceding the true syllable beginning (e.g., gold: 010, pred: 100).
- Late: A syllable beginning was marked on the grapheme following the true syllable beginning (e.g., gold: 010, pred: 001).
- Missed: A syllable beginning was not marked on the grapheme preceding, at, or following the true syllable beginning (e.g., gold: 010, pred: 000).
- Added: An extra syllable beginning was predicted (e.g., gold: 010, pred: 110).

The errors made by both models can be seen in Table 6. Clearly, the GRU model made far fewer errors than the Elman model. Additionally, the two models exhibit different error patterns; the GRU model has difficulty 'deciding' exactly where to mark syllable beginnings (Early/Late), while the Elman model has difficulty 'deciding' whether or not to mark syllable beginnings all together (Missed/Added).

[7]Meaning the tag '1', which indicates that a grapheme begins a syllable.

	Early	Late	Missed	Added	Total
Elman	232	184	746	397	1559
GRU	131	118	35	97	381

Table 6: Model Errors by Type

4.3 Limitations

Given that the current project presents a data-driven approach to automatic syllabification, possible limitations of the used data should be seriously considered. During the course of this project, one such limitation became apparent in the corpus of syllabified English words (Solheim, 2015). Explicitly, the corpus contains incorrect syllabifications.

Consider the forms presented in Table 7. In each form, codas are prioritized over onsets which is 1) uncommon cross-linguistically (Hayes, 2009) and 2) incorrect for English. So, the models trained on this data have been trained to make linguistically incorrect predictions, regardless of whether they generate the 'correct' syllabifications. Additionally, not all forms in the corpus are incorrectly syllabified (e.g. *bone;less*, *chee;tah*). This inconsistency potentially turns what should be a straightforward situation (mark inter-vocalic consonants as syllable beginnings) into an ambiguous situation (sometimes mark inter-vocalic consonants as syllable beginnings).

Word	Corpus Syllabification	Correct Syllabification
abated	a;bat;ed	a;ba;ted
beeper	beep;er	bee;per
cheater	cheat;er	chea;ter

Table 7: Incorrect Syllabifications

A second limitation concerns using random selection to split the syllabified corpus into training, development, and testing sets. Random selection allowed related word forms to appear across datasets. For example, *mo;ti;vate* appears in the training set and *mo;ti;vat;ed* appears in the testing set. Such overlap may make the task easier, and as such, artificially increase model performance.

5 Conclusion

Dinu et al. 2024 present an approach which successfully performs automatic syllabification of Italian orthographic forms. The current project sought to investigate the performance of said approach on English.

More specifically, Dinu et al. 2024 frame automatic syllabification as a sequence labeling task, where graphemes of an imputed word are labeled with either 1 (indicating that they begin a syllable) or 0 (indicating that they do not begin a syllable). They then perform classification with a bidirectional GRU model, and achieve an overall accuracy of 99.74% and an F1-score of .9969.

English and Italian, however, differ from each other in both orthographic complexity (where Italian's orthography is shallow, and English's orthography is deep) and syllable structure (where Italian's syllable structure is simple, and English's syllable structure is complex) (Seymour et al., 2003). Despite these differences, Dinu et al. 2024's approach achieved a comparable overall accuracy of 96.51% and an F1-score of .9612 when syllabifying English orthographic forms. So, Dinu et al. 2024's approach performs well on English.

Turning now to future work, while syllabified corpora have applications in low-resource language modeling (Oncevay et al., 2022), the current implementation utilized a ∼25,000 word corpus, making it difficult to directly apply in low-resource settings. As such, the performance of Dinu et al. 2024's approach on smaller datasets should be assessed. Alternatively, attempts should be made to adapt Dinu et al. 2024's approach to run specifically on smaller datasets.

Another direction for future work lies in the opposite direction, namely, to attempt implementation of larger neural models. Given the power and current popularity of transformer models, the application of said models to automatic syllabification should be considered.

References

Connie R. Adsett and Yannick Marchand. 2009. A comparison of data-driven automatic syllabification methods. In *String Processing and Information Retrieval*, pages 174–181, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chandran Savithri Anoop and Angarai Ganesan Ramakrishnan. 2023. Suitability of syllable-based modeling units for end-to-end speech recognition in sanskrit and other indian languages. *Expert Systems with Applications*, 220:119722.

Liviu Dinu, Ioan-Bogdan Iordache, Simona Georgescu, Alina Maria Cristea, and Bianca Guita. 2024. ItGraSyll: A computational analysis of graphical syllabification and stress assignment in Italian. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 316–324, Pisa, Italy. CEUR Workshop Proceedings.

Michael Hammond. 1997. Parsing syllables: modeling ot computationally. *Preprint*, arXiv:cmp-lg/9710004.

Bruce Hayes. 2009. *Introductory Phonology*. Wiley Blackwell.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Jacob Krantz, Maxwell Dulin, and Paul De Palma. 2019. Language-agnostic syllabification with neural sequence labeling. *Preprint*, arXiv:1909.13362.

Yannick Marchand, Connie R. Adsett, and Robert I. Damper. 2009. Automatic syllabification in english: A comparison of different algorithms. *Language and Speech*, 52(1):1–27. PMID: 19334414.

Arturo Oncevay, Kervy Dante Rivas Rojas, Liz Karen Chavez Sanchez, and Roberto Zariquiey. 2022. Revisiting syllables in language modelling and their application on low-resource machine translation. *Preprint*, arXiv:2210.02509.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Philip Seymour, Mikko Aro, and Jane Erskine. 2003. Foundation literacy acquisition in european orthographies [electronic version]. *British Journal of Psychology*, 94:143–174.

Gaute Solheim. 2015. 25,000 syllabified word list.

Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu. 2018. Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese. *Preprint*, arXiv:1804.10752.