Revisiting Neural Language Modelling with Syllables

Arturo Oncevay†‡ Kervy Rivas Rojas‡
† School of Informatics, University of Edinburgh, Scotland
‡ Artificial Intelligence Research Group, Pontificia Universidad Católica del Perú, Peru
a.oncevay@ed.ac.uk, k.rivas@pucp.edu.pe

Abstract
Language modelling is regularly analysed at
word, subword or character units, but syllables
are seldom used. Syllables provide shorter
sequences than characters, they can be ex-
tracted with rules, and their segmentation typi-
cally requires less specialised effort than iden-
tifying morphemes. We reconsider syllables
for an open-vocabulary generation task in 20
languages. We use rule-based syllabification
methods for five languages and address the rest
with a hyphenation tool, which behaviour as
syllable proxy is validated. With a compara-
tible perplexity, we show that syllables outper-
form characters, annotated morphemes and un-
supervised subwords. Finally, we also study
the overlapping of syllables concerning other
subword pieces and discuss some limitations
and opportunities.

1 Introduction

In language modelling (LM), we learn distributions
over sequences of words, subwords or characters,
where the latter two can allow an open-vocabulary
generation (Sutskever et al., 2011). We rely on
subword segmentation as a widespread approach to
generate rare subword units (Sennrich et al., 2016).
However, the lack of a representative corpus, in
terms of the word vocabulary, constrains the unsu-
ervised segmentation (e.g. with scarce monolin-
gual texts (Joshi et al., 2020)). As an alternative,
we could use character-level modelling, since it
also has access to subword information (Kim et al.,
2016), but we face long-term dependency issues
and require longer training time to converge.

In this context, we focus on syllables, which
are based on speech units: “A syl-la-ble con-
tains a sin-gle vow-el u-nit”. These are more linguistically-
based units than characters, and behave as a map-
ing function to reduce the length of the sequence
with a larger “alphabet” or syllabary. Their extrac-
tion can be rule-based and corpus-independent, but
data-driven methods or hyphenation using diction-
aries can approximate them as well (see §4).

Previous work on syllable-aware neural
LM failed to beat characters in a closed-vocabulary
generation at word-level (Assylbekov et al.,
2017); however, we propose to assess syllables
under three new settings. First, we analysed an
open-vocabulary scenario with syllables by disre-
garding additional functions in the input layer (e.g.
convolutional filters to hierarchically compose
the representations (Botha and Blunsom, 2014)).
Second, we extended the scope from 6 to 20
languages to cover different levels of orthographic
depth, which is the degree of grapheme-phoneme
correspondence (Borgwaldt et al., 2005) and a fac-
tor that can increase complexity to syllabification.
English is a language with deep orthography (weak
correspondence) whereas Finnish is transparent
(Ziegler et al., 2010). Third, we distinguished
rule-based syllabification with hyphenation tools,
but also validated their proximity for LM.

Therefore, we revisit LM for open-vocabulary
generation with syllables using pure recurrent neu-
ral networks (Merity et al., 2018) for a more diverse
set of languages, and compare their performance
against characters and other subword units.

2 Open-vocabulary language modelling
with a comparable perplexity

Language modelling Given an input of generic
sequence units (such as words, subwords or char-
acters), denoted as $s = s_1, s_2, \ldots, s_n$, a language
model computes the probability of $s$ as:

$$p(s) = \prod_{i=1}^{n} p(s_i | s_1, s_2, \ldots, s_{i-1})$$

(1)

We then can use a recurrent neural network or
RNN (e.g. a LSTM variation (Merity et al., 2018)),
trained at each time step $t$, for calculating the prob-
ability of the input $s_{t+1}$, given $s_t$:
\[ p(s_{t+1}|s_{\leq t}) = g(\text{RNN}(w_t, h_{t-1})) \] (2)

where $w_t$ is the learned embedding of the sequence unit $s_t$, $h_{t-1}$ is the hidden state of the RNN for the previous time step, and $g$ is a softmax function for the vocabulary space of the segmentation.

We thereafter compute the loss function $\mathcal{L}_{\text{LM}}(s)$ for the neural LM as the cross entropy of the model for the sequence $s$ in a time step $t$:
\[ \mathcal{L}_{\text{LM}}(s) = - \sum_{j=1}^{|V|} b_{t,j} \times \log (p(s_{t,j})) \] (3)

where $|V|$ is the size of vocabulary, $p(s_{t,j})$ is the probability distribution over the vocabulary at each time-step $t$ and $b_{t,j}$ is an indicator if $j$ is the true token for the sequence $s$.

**Character-level perplexity** For a fair comparison across all granularities, we evaluate all results with character-level perplexity:
\[ \text{ppl}^c = \exp \left( \mathcal{L}_{\text{LM}}(s) \cdot \frac{|s^\text{seg}| + 1}{|s^\text{c}| + 1} \right) \] (4)

where $\mathcal{L}_{\text{LM}}(s)$ is the cross entropy of a string $s$ computed by the neural LM, and $|s^\text{seg}|$ and $|s^\text{c}|$ refer to the length of $s$ in the chosen segmentation and character-level units, respectively (Mielke, 2019). The extra unit considers the end of the sequence.

**Open-vocabulary output** We generate the same input unit (e.g. characters or syllables) as an open-vocabulary LM task, where there is no prediction for the length of the sequence unit.

### 3 Experimental setup

**Languages and datasets** Corpora are listed in Table 1. We do not use the English Penn Treebank (Marcus et al., 1993) as in Assylbekov et al. (2017), given that it is not suitable for open-vocabulary assessment. We then chose WikiText-2-raw (en$\text{wt}_2$; Merity et al., 2016), which contains around two million word-level tokens extracted from Wikipedia articles in English. Furthermore, we employ 20 Universal Dependencies (UD; Nivre et al., 2020) treebanks, similarly to Blevins and Zettlemoyer (2019).1

Table 1: Total number of tokens and token types in the training set for words, syllables and characters (other segmentation data is in Appendix A). We highlight (*) the languages with extracted rule-based syllables, and we report the statistics with hyphenation for the rest.

**Syllable segmentation** For splitting syllables in different languages, we used rule-based syllabification tools for English, Spanish, Russian, Finnish and Turkish, and a dictionary-based hyphenation tool for all of them except for Finnish and Turkish. All the tools are listed in Appendix B.

**Segmentation baselines** Besides the annotated morphemes in the UD treebanks, we consider Polygloss (polyglot-nlp.com), which includes models for unsupervised morpheme segmentation trained with Morfessor (Virpioja et al., 2013). Moreover, we employ an unsupervised subword segmentation baseline of Byte Pair Encoding (BPE; Sennrich et al., 2016) with different vocabulary sizes from 2,500 to 10,000 tokens, with 2,500 steps. We also fix the parameter to the syllabary size. Appendix B includes details about the segmentation format.

**Model and training** Following other open-vocabulary LM studies (Mielke and Eisner, 2019; Mielke et al., 2019), we use a low-compute version of an LSTM neural network, named Average SGD Weight-Dropped (Merity et al., 2018), developed in PyTorch.3 Appendix C includes more details.

---

1The languages are chosen given the availability of an open-source syllabification or hyphenation tool. We prefer to use the UD treebanks, instead of other well-known datasets for language modelling (e.g. Multilingual Wikipedia Corpus (Kawakami et al., 2017)), because they provide morphological annotation used for the study.

2We use: https://github.com/salesforce/awd-lstm-lm

3https://github.com/awslabs/open-lm
4 Segmentation analysis

Syllabification versus Hyphenation For English, Spanish and Russian, we confirmed that hyphenation is a reasonable proxy for syllabification in LM. Table 2 shows that the overlapping of hyphenation pieces concerning rule-based syllables is more than 70%. Furthermore, we note that the perplexity gap is not significant. The small sample is still representative because English has a deep orthography in contrast with Spanish or Russian, and Russian use a different script (Cyrillic instead of Latin). In terms of morphology, all of them are fusional, but highly agglutinative languages like Turkish or Finnish are not feasible candidates for developing a Hunspell-like dictionary, which is the source of most hyphenation tools.

|        | $\frac{(H \cap S)}{S}$ | $\frac{(S \cap H)}{H}$ | $\Delta \text{ppl}^c_{S,H}$ |
|--------|------------------------|------------------------|--------------------------|
| ca     | 70.07%                 | 73.84%                 | -0.0001                  |
| de     | 78.13%                 | 74.68%                 | -0.0186                  |
| es     | 76.37%                 | 61.95%                 | 0.0373                   |
| en     | 73.58%                 | 56.13%                 | 0.0901                   |

Table 2: Token types overlapping of syllabification (S) versus hyphenation (H) and their perplexity gap.

Vocabulary growth In Figure 1, we show the vocabulary growth rate of unsupervised morphemes (with Morfessor)\(^4\) in contrast with syllables extracted by rules (S) or hyphenation (H). We do not observe a significant difference between syllabification and hyphenation, which reinforce their functional proximity for our study. We also observe that in all the UD datasets we do not have more than 10k Morfessor-based pieces, which is the reason for the upper boundary on our BPE baselines. However, for Czech, German, English and French, we observe that the syllabary size significantly surpass the number of Morfessor-based pieces. Potential explanations are the orthographic depth and the degree of syllabic complexity for those languages (Borleffs et al., 2017): a low letter-sound correspondence and the difficulty to determine the syllable boundaries can induce a larger syllabary.

Overlapping of syllables with subwords In Figure 2a, we show the stacked area plot of the overlapping ratio of syllables, Morfessor-based pieces and different BPE settings concerning the annotated morphemes. We note that the proportion of syllable overlapping is relatively low in contrast with unsupervised morphemes or BPE with larger vocabulary size. The outcome is expected, as the annotated morpheme contains lemmas or full words with long sequences of characters, which are adopted by BPE with more merge-operations.

Analogously, Figure 2b shows the overlapping of syllables and BPE concerning unsupervised morphemes. We observe that the intersection ratio of syllables has increased, and approximate the values of BPE with larger vocabulary size.

In both scenarios, it is worthy to note that the BPE setting with the largest overlapping with annotated or unsupervised morphemes has a vocabulary size that equates the syllabary. Future work can assess whether the number of unique syllables can support the tuning of the BPE vocabulary size.

Finally, in Figure 2c, we observe that syllables overlap the most with BPE pieces that uses a small vocabulary size of 2,500. From 5,000 to 10,000, the overlapping ratio shows a downtrend for most of the datasets. However, in 9 out of 20 datasets, the largest BPE setting (fixed with the syllabary size) shows an increase of the overlapping.

5 Open-vocabulary LM results

Table 3 shows the ppl\(^c\) values for the different levels of segmentation we considered in the study,
where we did not tune the neural LM for an specific setting. We observe that syllables always result in better perplexities that other granularities, even for deep orthography languages such as English or French. The results obtained by the BPE baselines are relatively poor as well, and they could not beat characters in any dataset. We could search for an optimal parameter for the BPE algorithm; however, the advantage of the syllables is that we do not need to tune an hyper-parameter to extract a different set of subword pieces.

As a significant outcome, we note that syllables did not fail to beat characters, at least in an open-vocabulary LM task, which extends the results provided by Assylibekov et al. (2017). Moreover, in Figure 4b in the Appendix, we can observe a strong linear relationship of the syllable type/token ratio with the gain of ppl_c obtained by syllables concerning characters. In other words, if our dataset possesses a rich syllabary, we are fairly approximating the amount of word-level tokens, which reduces the ppl_c gain.

Beating characters implies a gain in time processing as well, given the shorter sequences of the syllable pieces. Figure 3 shows details about how many epochs each segmentation requires to converge during training, as well as how much time requires all the training. Syllable-level LM trains faster than characters, and even when they are slower (seconds per epoch) than using BPE-pieces or unsupervised morphemes, they obtain a better ppl_c score.
6 Related work

In subword-aware LM, Vania and Lopez (2017) investigated if we can capture morphology using characters, character n-grams, BPE pieces or morphemes with a closed-vocabulary. Whereas for open-vocabulary generation, Blevins and Zettlemoyer (2019) incorporates morphological supervision with a multi-task objective, and Kawakami et al. (2017); Mielke and Eisner (2019) have focused in improving the neural architecture to jointly use the representation of characters and words in a hybrid open-vocabulary setting.

The closest study to ours is from Mikolov et al. (2012), where they performed subword-grained prediction with different settings, and used syllables as a proxy to split words with low frequency, reduce the vocabulary and compress the model size. However, they only focused on English. Besides, syllable-aware LM was addressed by Assylbekov et al. (2017) for English, German, French, Czech, Spanish and Russian, and by Yu et al. (2017) for Korean. However, in both cases, the syllable units have been composed with convolutional filters into word-level representations to assess a closed-vocabulary setting. As far as we know, we propose the first syllable-level open-vocabulary LM study that analyses up to 20 languages and compare their results against morphemes.

7 Limitations and opportunities

Syllables only cannot offer a universal solution to the subword segmentation problem for all the languages, as the syllabification tools are language-dependent. Besides, the analysis should be extended to different scripts and morphological typology. Furthermore, we do not encode any semantics in the syllable-vector space, with a few exceptions like in Korean (Choi et al., 2017).

Nevertheless, our results confirm that syllables are reliable for LM, and building a syllable splitter might require less effort than annotating morphemes to train a robust supervised tool. Moreover, as we discussed at the end of §4, we could further assess whether the syllables can support the hyper-parameter tuning or impact the internal procedure of unsupervised segmentation methods like BPE or Morfessor, which are corpus-dependent.

8 Conclusion

We proved that syllables are valuable for an open-vocabulary LM task, where they behave positively even for languages with deep orthography, and they overcome character-level models that require longer time to train. Syllables do not have an embedded meaning in most of the languages; however, the required effort for their segmentation could be advantageous against morphological-aware or unsupervised-driven methods. Given our analysis, we could consider working on syllable-driven subword segmentation, neural machine translation and hybrid-LM (with characters and words).

Acknowledgments

The first author is supported by funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements No 825299 (GoURMET) and the EPSRC fellowship grant EP/S001271/1 (MT-Stretch), whereas the second author is supported by a research grant of the “Consejo Nacional de Ciencia, Tecnolog ía e Innovaci ón Tecnol ógica” (CONCYTEC, Peru) under the contract 183-2018-FONDECYT-BM-IADT-MU.

We express our thanks to Shay B. Cohen, Clara Vania and Adam López, whose provided us feedback in a very early stage of the study, and to Alexandra Birch, Barry Haddow, Fernando Alva-Manchego, John E. Miller and Marco Sobrevilla-Cabezudo, whose reviewed different draft versions. Finally, we acknowledge the support of NVIDIA Corporation with the donation of a Titan Xp GPU used for the study.

References

Zhenisbek Assylbekov, Rustem Takhanov, Bagdat Myrzakmetov, and Jonathan N. Washington. 2017. Syllable-aware neural language models: A failure to beat character-aware ones. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1866–1872, Copenhagen, Denmark. Association for Computational Linguistics.
Terra Blevins and Luke Zettlemoyer. 2019. Better character language modeling through morphology. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, pages 1606–1613, Florence, Italy. Association for Computational Linguistics.

Susanne R Borgwaldt, Frauke M Hellwig, and Annette MB De Groot. 2005. Onset entropy matters—letter-to-phoneme mappings in seven languages. Reading and Writing, 18(3):211–229.

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

Jan A. Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML’14, pages II–1899–II–1907. JMLR.org.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. 2017. Learning to create and reuse words in open-vocabulary neural language modeling. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 6282–6293, Online. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI’16, pages 2741–2749. AAAI Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics, 19(2):313–330.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing LSTM language models. In International Conference on Learning Representations.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843.

Sabrina J. Mielke. 2019. Can you compare perplexity across different segments? Available in: http://sjmielke.com/comparing-perplexities.htm.

Sebastian J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. What kind of language is hard to language-model? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4975–4989, Florence, Italy. Association for Computational Linguistics.

Sebastian J. Mielke and Jason Eisner. 2019. Spell once, summon anywhere: A two-level open-vocabulary language model. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 6843–6850.

Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. 2012. Subword language modeling with neural networks. preprint (http://www.fit.vutbr.cz/~tmikolov/rnnlm/char.pdf), 8:67.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 4034–4043, Marseille, France. European Language Resources Association.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, James Martens, and Geoffrey Hinton. 2011. Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML’11, pages 1017–1024, USA. Omnipress.
A Datasets

Table 4 shows the size of the training, validation and test splits for all the datasets used in the study.

|        | Train   | Valid   | Test    |
|--------|---------|---------|---------|
| Word   | Syl     | Char    | Word    | Syl     | Char    | Word    | Syl     | Char    |
| bg     | 125     | 386     | 710     | 16      | 50      | 92      | 16      | 49      | 90      |
| ca     | 436     | 1,123   | 2,341   | 59      | 152     | 317     | 61      | 157     | 327     |
| cs     | 1,158   | 3,546   | 6,868   | 157     | 482     | 933     | 172     | 524     | 1,012   |
| da     | 81      | 215     | 442     | 10      | 28      | 57      | 10      | 27      | 56      |
| de     | 260     | 735     | 1,637   | 12      | 34      | 75      | 16      | 45      | 102     |
| es     | 210     | 488     | 1,061   | 26      | 61      | 133     | 26      | 61      | 132     |
| en     | 2,089   | 4,894   | 10,902  | 218     | 505     | 1,157   | 246     | 568     | 1,304   |
| fr     | 154     | 484     | 930     | 20      | 62      | 119     | 23      | 75      | 145     |
| it     | 263     | 762     | 1,504   | 11      | 32      | 64      | 10      | 28      | 57      |
| lv     | 113     | 349     | 690     | 19      | 58      | 115     | 20      | 59      | 116     |
| nl     | 187     | 488     | 1,074   | 12      | 30      | 66      | 11      | 31      | 68      |
| pl     | 102     | 293     | 589     | 13      | 37      | 73      | 13      | 37      | 74      |
| pt     | 192     | 551     | 1,040   | 10      | 29      | 54      | 9       | 27      | 51      |
| ro     | 183     | 549     | 1,056   | 17      | 51      | 98      | 16      | 48      | 94      |
| ru     | 867     | 2,707   | 5,411   | 118     | 364     | 722     | 117     | 360     | 717     |
| sk     | 80      | 232     | 437     | 12      | 39      | 76      | 13      | 41      | 80      |
| sl     | 38      | 126     | 242     | 10      | 33      | 63      | 10      | 33      | 64      |
| uk     | 88      | 289     | 501     | 12      | 41      | 71      | 16      | 56      | 99      |

Table 4: Total number of tokens (in thousands) at word, syllable and character-level for all the splits.

B Segmentation

Tools

- English syllabification: Extracted from https://www.howmanysyllables.com/
- Spanish syllabification: https://pypi.org/project/pylabeador/
- Russian syllabification: https://github.com/Koziev/rusyllab
- Finnish syllabification: https://github.com/tsnaomi/finnsyll
- Turkish syllabification: https://github.com/MeteHanC/turkishnlp
- Hyphenation: PyPhen (https://pyphen.org/), which is based on Hunspell dictionaries.

Format

For syllables, we adopt the segmentation format used by SentencePiece (Kudo and Richardson, 2018) to separate subwords: “A @ syl la ble @ con tains @ a @ sin gle @ vow el @ u nit”, where “@” is a special token that indicates the word boundary. We also evaluated syllables with a segmentation format like in Sennrich et al. (2016): “A syl@ la@ ble con@ tains a ...”, but we obtained lower performance in general.
Figure 4: (a) $V_{\text{syl}}/N_{\text{syl}}$ vs. $V_{\text{word}}/N_{\text{word}}$. (b) $V_{\text{syl}}/N_{\text{syl}}$ vs. $\Delta \text{ppl}^c_{\text{char-syl}}$

C Model and Training

In contrast with the default settings, we use a smaller embedding size of 500 units for faster training. Additionally, we have 3 layers of depth, 1152 of hidden layer size and a dropout of 0.15. We train for 25 epochs with a batch size of 64, a learning rate of 0.002 and Adam optimiser (Kingma and Ba, 2015) with default parameters. We fit the model using the one cycle policy and an early stopping of 4. We run our experiments in a NVIDIA Titan Xp.

D Validation results

Figures 5 and 6 shows the validation perplexity (ppl$^c$) and the training time until convergence, respectively, in all the Universal Dependency tree-banks.

E Complementary discussion

Type/token ratio of syllables In Figure 4a, we show a scatter plot of the token/type growth rate of syllables versus words for all languages and corpora. In other words, the ratio of syllable-types (syllabary or $V_{\text{syl}}$) per total number of syllable-tokens ($N_{\text{syl}}$) versus the type/token ratio of words ($V_{\text{word}}/N_{\text{word}}$) in the train set. The figure suggests at least a weak relationship, which agrees with the notion that a low word-vocabulary richness only requires a low syllabary richness for expressivity. Also, a richer vocabulary can use a richer syllabary or just longer words, so the distribution of the vocabulary richness could be larger.

We expected that the syllabary growth rate ($V_{\text{syl}}/N_{\text{syl}}$) for a low phonemic language like English would be relatively high, but wikitext-2 (en-wt2) is located in the bottom-left corner of the plot, probably caused by its large amount of word-tokens. However, we observe a large $V_{\text{syl}}/N_{\text{syl}}$ for the English (en-UD) and French (fr) treebanks, despite their low $V_{\text{word}}/N_{\text{word}}$ ratio, which is an expected pattern for languages with deep orthographies.

We also observe that languages with a more transparent orthography, like Czech (cs) or Finnish (fi), are located in the left side of the figure, whereas Turkish (tr) is around the middle section. Nevertheless, our study does not aim to analyse the relationship between the level of phonemic orthography with the $V_{\text{syl}}/N_{\text{syl}}$ ratio. For that purpose, we might need an instrument to measure how deep or shallow a language orthography is (Borgwaldt et al., 2005; Borleffs et al., 2017), and a multi-parallel corpus for a more fair comparison.
Figure 5: Validation perplexity for all the UD treebanks.
Figure 6: Training time (in seconds) until convergence for the UD treebanks