Syll2Vec: A Phonetic Approach in Reading Comprehension

Xiao Lu[*]
CS224N Default Final Project
xiaolu@stanford.edu

Abstract

This project explores a different sub-word level embedding approach using syllables. Whereas other sub-word level approaches (e.g. n-gram) operates on language morphology, syllable embedding operates on phonemes. Because the number of possible phonemes is much lower than the number of character combination, syllable embedding are expected to shed new light on correlation between words that sound similar, but are spelled differently. In this project, a syllable embedding is trained using the same text corpus from assignment 2 that trained character embedding. The embedding was further trained on the SQuAD 2.0 dataset. The results show no significant advantage of syllable embedding over character embedding, and the two embedding do not show synergistic effect when working together.

1 Introduction

Traditionally, morphemes have been studied as the smallest semantic unit (e.g. breaking down words into stem and root). Sub-word level units are generally used to resolve the challenge of out-of-vocabulary words. Successful attempts at such endeavor includes n-grams embedding and character level embedding[11][10]. The main challenges for model below the word level involve the need to handle large, open vocabulary, rich morphology, transliteration and informal spelling.

This project explores the efficacy of using phonemes as basic unit of embedding. Whereas character embedding is purely based on morphology (how letters are arranged), phoneme embedding are phonetically based (referred to as 'syllable' embedding for the rest of the report). The basic assumption for character embedding is that words that are spelled similarly should have similar meaning. The assumption for syllable embedding is that words that are made of similar sound should have similar meaning.

The hypothesis of this project is that syllabus embedding can be equally helpful in English language as in Korean, and more helpful than character embedding. The line of reasoning is that although character embedding have small vocabulary size (upper case, lowercase letters, numbers, special characters). The possible permutation is huge. The character decomposition of a word of length n results in 52^n possible permutation (considering upper and lower case letters only).

In contrast, a word can be conveniently separated into syllables, and many out-of-vocabulary words are combination of reusable syllables. A long word, such as "stationary" can be broken into three-syllable "sta-tion-ary" instead of nine-letter combination. The exponent drops from 9 to 3, and the possible permutations are greatly reduced from $|V_c|^9$ to $|V_s|^3$ (five magnitude lower). A neural net with good learned embedding for the three syllables should learn that the vocabulary is an adjective that relates to a state (as "ary" from binary, planetary, military, and "tion" in action, condition, motion).

[*] Stanford University CS224N teaching staff for all the resources and support

2 Related Works

Syllable embedding have been applied to highly agglutinative language such as Korean, which exhibits complex morphology of words that renders word embedding less effective than for less agglutinative languages[5]. The results shows its robustness to out-of-vocabulary phrases in Korean language. In particular, an implementation of simple RNN with syllable and morphene embedding outperformed the character level embedding counterpart by 16.87 perplexity with 9.5 million parameters, on a prediction task in Korean language[12].

While Korean language has wide variety of possible agglutinations possible, English language is more regular morphology, with cleanly separated word and character. In English-related NLP works, syllables were studied in speech translation task, but not in text comprehension task[13].

3 Approach

Initially, the baseline model is intended to be the R-net model, using word embedding and character embedding [1]. Another baseline is the bidirectional attention flow model, also using word embedding and character embedding. These baselines are compared to the performance of syllable embedding and word embedding. The F1 and EM metrics are used to evaluate performance, as required by the leaderboard.

The syllables are extracted using a syllabify module which depends on the CMU Pronouncing Dictionary of North American English word pronunciations [8]. The module is simplified to ignore stress on vowel, and original implementation of symbolizing syllable as short 3 4 character string is integrated into the module.

The syllabify module is applied to the text sentences corpus dataset from assignment two. The dataset contains 11855 sentences with sentiment scores. The same methodology of word2vec is applied to learn syllable embedding with negative sampling. The overall process can be seen as breaking every word into smaller, more modular words that represent its sound, instead of its spelling. Each English word maps to at least one syllable, just as each English word maps to at least one character.

After extracting syllables, syllables are indexed and have embedding randomly initialized. The word2vec model is applied to train syllable embedding. Afterward, the syllable embedding is incorporated into the SQuAD starter code, and loaded along with word embedding. The embedding is further fine-tuned during the SQuAD training process.

> Source sentence 1: Steers turns in a snappy screenplay that curls at the edges ; it 's so clever you want to hate it .
>
> Syllabified sentence 1: stihrz ternz ihn ah snae piy skriyn pley dhaet kerlz aet dhah eh jhahz ; iht ehs sow kleh ver yuw waant tuw hheyt iht .
>
> Source sentence 2: What really surprises about Wisegirls is its low-key quality and genuine tenderness .
>
> Syllabified sentence 2: waht rih liy ser pray zihz ah bawt wisegirls ihz ihts low-key kwaa lah tiy ahnd jheh nyah wahn tehn der nahs .
>
> Source sentence 3: The film provides some great insight into the neurotic mindset of all comics – even those who have reached the absolute top of the game .
>
> Syllabified sentence 3: dhah fihlm prah vaydz sahm greyt ihn sayt ihn tuw dhah nuh raa tihk maynd seht ahv aol kaa mihks – iy vihn dhowz hhuw hhaev riycht dhah aeb sah luwt taap ahv dhah geym .

The syllable embedding is taken as input by a bidirectional LSTM unit, and the hidden states are taken as the syllable-based word embedding, and are concatenated with the pre-trained word embedding. The concatenated embedding is passed through a projection layer, a high-way layer, and encoded by a RNN encoder to acquire the preliminary representation of the original paragraph and question. Then depending on the model characteristic, different branches are used.

BiDAF model For the BiDAF model, the context and question representation are taken as input by the BiDAF attention layer, and finally predictions are generated by the output layer. The attention layer computes direction in both direction: context-aware question representation and question-aware context representation. The output layer applies a linear transformation, which is followed by a softmax layer to compute the starting pointer, and a bidirectional LSTM to compute the ending pointer.

Since the BiDAF model is given as baseline, it requires minimal modification except for incorporating the syllable embedding when instantiating the model, coding the syllable embedding layer, and mapping word to syllable during training and testing.

R-net For the R-net applies the Gated Attention-Based Recurrent Networks to modify the representation of each passage word to become aware of the question ($\{v_t^P\}_{t=1}^n$). For passage word at step t, attention c_t is aggregated over the entire question u^Q. A signmoid input gate is added to attenuate the cell state c_t, in order to capture the relation between current context word u_t^P and the entire question.

$$v_t^P = \text{RNN}(v_{t-1}^P, \text{sigmoid}(W_g[u_t^P, c_t]) \odot [u_t^P, c_t]) \tag{1}$$

$$c_t = \text{att}(u^Q, [u_t^Q, v_{t-1}^P]) = u^Q \cdot \text{softmax}(v^t \text{tanh}(W_u^Q u_j^Q + W_u^P u_t^P + W_v^P v_{t-1}^P)) \tag{2}$$

The R-net then matches question-aware representation of passage words to other parts of the passage, to increase each word's awareness of the entire context. As RNN processes each passage word, it collects related evidence from everywhere else in the passage. A sigmoid gate is added to attenuate the input to the RNN.

$$h_t^P = \text{BiRNN}(h_{t-1}^P, \text{sigmoid}(W_g'[v_t^P, c_t]) \odot [v_t^P, c_t]) \tag{3}$$

$$c_t = \text{att}(u^Q, [u_t^Q, v_{t-1}^P]) = v^P \cdot \text{softmax}(v^t \text{tanh}(W_v^P v_j^P + W_v^{\tilde{P}} u_t^P)) \tag{4}$$

Finally, the R-net applies is a pointer network[4] that selects the starting word and ending word from the passage[3]. The model was trained to minimize the negative log likelihood loss for the starting and ending words.

4 Experiment

Out of the original sentences dataset, a total number of 21701 distinct words were found. After transforming into syllables, a total of 5524 syllables were found, resulting in 74.54% compression rate. Furthermore, only 0.24% of the vocabulary failed to translate into syllables. Upon inspection, those words are not named entity or out-of-vocabulary words. The issue lies within the syllabification process. A few example are includes: Engrossing, branched, engrossing and self-congratulation. The syllable embedding took 4 hours to train, and converged in 40,000 iteration. The resulting embedding is used by the SQuAD starter code, loaded in the same way as word embedding and character embedding.

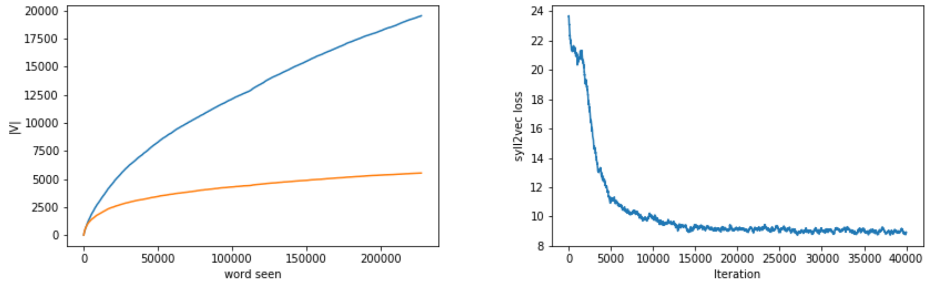

Figure 1: Growth rate of syllable size versus vocabulary size; visualization of top 20 most frequent syllables.

Table 1: BiDAF model with different flavor of embedding

Variant	Dev NLL	F1	EM	AvNA
Baseline	03.29	59.9	56.63	67.48
Baseline char	02.94	63.79	60.46	70.36
Baseline syll	03.17	63.61	60.33	70.83
Baseline char + syll	03.32	63.25	59.55	70.43

The syllable embedding is incorporated through the following process. The syllabify module maps every English word in the word embedding to its syllable constituents. The syllable embedding are looked up by index. Those without embedding are replaced with –OOV– token. The dimension of the loaded syllable embedding for each batch is (batch size, sentence length, max word length measured in syllable, syllable embedding size). After being looked up, the question embedding and context embedding are concatenated with the word embedding, just as character embedding are concatenated with word embedding.

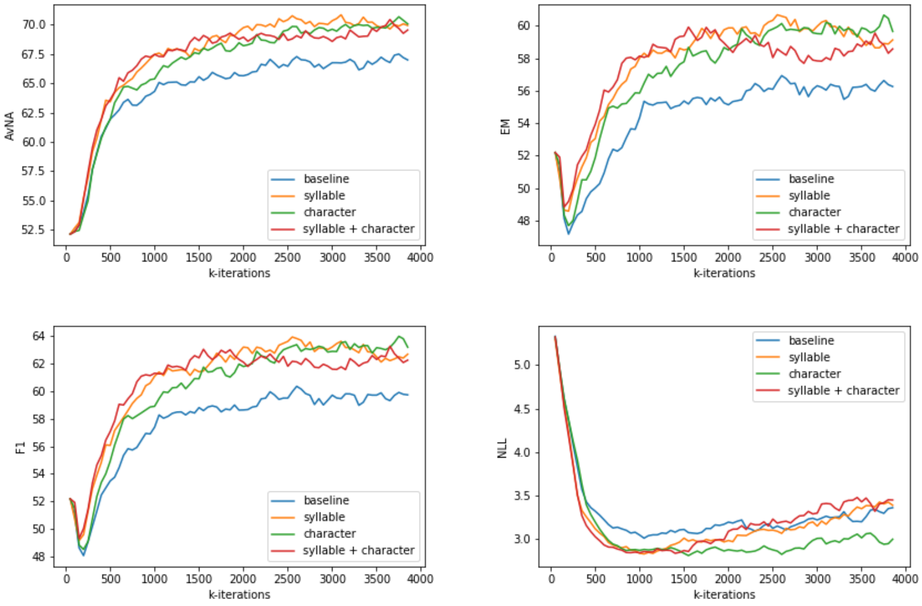

Figure 2: Growth rate of syllable size versus vocabulary size; visualization of top 20 most frequent syllables.

The R-net implementation is ultimately unsuccessful. While the model could train without crash, the negative log loss remains mysteriously negative all the time. R-net implementation from GitHub is used as reference, which also turns out to be broken and poorly implemented. In the end, the project proceeds with the BiDAF model only, and acquires the following result[9].

All three variants — word embedding with (1) character embedding, (2) syllable embedding, (3) both character and syllable embedding beat the baseline definitively. Among the three variants, character embedding achieves the best score in Dev NLL, F1 and EM. The other two variants with syllable embedding oscillate at similar level, but began to overfit after 2.5 million iteration, which is evident in the rising NLL curve. When combining syllable and character together with word embedding, the model clearly overfit the data, which is evident from the rise of dev NLL curve above ever other model, and the corresponding decline in EM and F1 scores. The syllable embedding model achieves the 23rd place on the leaderboard.

The reason that syllable model under performs compared to character embedding can be traced back to two major source. First, the syllable vocabulary (cardinality of the syllable embedding) is too small. The syllable space is extracted from the Stanford tree sentiment database, which is much smaller than the SQuAD database. Among the vocabulary of the SQuAD dataset, 40.4% of the words fail to translate to syllables, meaning that nearly half of the words are mapped to –OOV– token in the syllable embedding. To fix this problem, one must inspect deeper into the syllabification module to increase its success rate in syllabifying words.

The second possible reason for its sub-optimal performance is that among those words that are successfully syllabified, over half of the syllables do not have embedding. Specifically, 6,986 (out of 12,073) syllables found in the SQuAD dataset do not have matches in the embedding trained from the Stanford tree sentiment dataset. A simple fix to this problem is to randomly initialize the embedding for those syllables without existing embedding.

5 Conclusion

This project explores the efficacy of syllable embedding as a complement to character embedding in specifics, and to sub-level morphology-based embedding in general. The result shows that syllable embedding significantly improves upon the baseline, just as character embedding does. This comparison validates the effectiveness of syllable embedding. However, syllable embedding fails to surpass character level embedding. The reason for its failure cannot be attributed to the syllable embedding methodology itself. Ablative analysis shows that half of the vocabulary in SQuAD dataset fail to translate to syllable constituents. This problem can be mediated with more sophisticated syllabification module. Also, about 58% of the syllables extracted from the SQuAD vocabulary do not have any match from the embedding trained from Stanford tree sentiment dataset. This issue can be simply fixed by randomly initializing embedding at the beginning of the SQuAD training process. Unfortunately, due to limited time, this simple fix is not implemented.

In conclusion, the topic of syllable embedding is severely lacking attention in the English language NLP literature, because English is widely regarded as a non-agglutinated language the way Korean and Arabic are. However, English language has evolved to incorporates variety of foreign words (from French, German, Latin etc.), many of which retained their native spelling but adopted English phonemes. Syllable embedding has the potential to capture the phonetic similarity between those words with peculiar spelling, in a way that morphology based embedding cannot.

Acknowledgments

I'd like to thank all Stanford University CS224N teaching staff for all the resources and support that make this project possible. I'm taking this course P/NP, but enjoyed the learning process very much, and ended up learning more than I could expect.

References

[1] 2017. R-Net: Machine Reading Comprehension with Self-Matching Networks Natural Language Computing Group, Microsoft Research Asia

[2] Liang, Frank M. 1983. *Word hy-phen-a-tion by computer*. Ph.D Thesis, Stanford University.

[3] Shuohang Wang and Jing Jiang. Learning natural language inference with LSTM. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 2016a.

[4] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2692–2700, 2015.

[5] Seunghak Yu, Nilesh Kulkarni, Haejun Lee, Jihie Kim Syllable-level Neural Language Model for Agglutinative Language *arXiv:1708.0551*, 2017.

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016.

[7] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. CoRR, abs/1611.09268, 2016.

[8] Syllabify GitHub repository: https://github.com/cainesap/syllabify

[9] R-net GitHub repository: https://github.com/matthew-z/R-net

[10] Huang, Po-Sen and and Gao, Jianfeng Learning Deep Structured Semantic Models for Web Search using Clickthrough Data In *ACM International Conference on Information and Knowledge Management (CIKM)*, 2013.

[11] Chung, Junyoung, Kyunghyun Cho, and Yoshua Bengio. "A character-level decoder without explicit segmentation for neural machine translation." arXiv preprint arXiv:1603.06147 (2016).

[12] Yu, Seunghak, et al. "Syllable-level neural language model for agglutinative language." arXiv preprint arXiv:1708.05515 (2017).

[13] Salverda, Anne Pier, Delphine Dahan, and James M. McQueen. "The role of prosodic boundaries in the resolution of lexical embedding in speech comprehension." Cognition 90.1 (2003): 51-89.