Probing Linguistic Knowledge In Non Latin Script Neural Language Models

Abed Qaddoumi
New York University
amq259@nyu.edu

Anusha Patil
New York University
arp624@nyu.edu

Francesca Guiso
New York University
fg746@nyu.edu

Priyanka Shishodia
New York University
ps4118@nyu.edu

1 Description

The project will explore the linguistic knowledge implicitly learned by character-based recurrent neural networks (RNN) and its variants including long short-term memory (LSTM) and gated recurrent units (GRU). It will compare these character-based architectures with their word-based level counterparts. The project will focus on understanding morphology and syntactic dependencies in the following languages: English, Italian, Arabic, and Hindi by probing the networks using a correlation analysis technique called "diagnostic classifier".

The hypothesis is that the character-based model for non-Latin synthetic languages like Arabic and poly-synthetic language like Hindi will capture morphological and syntactic linguistic knowledge. The project will explore character-based models' understanding of linguistic features on smaller datasets of English and Italian when compared to the larger models explored in (Hahn and Baroni, 2019). There have been other developments in using character-based neural networks for machine translation in (Ling et al., 2015) and (Chung et al., 2016), where both papers used different types of RNNs on unsegmented words and gave input as characters to the RNNs for neural translations.

We will use one of the probing techniques that has been used to explore the knowledge in diagnostic classifiers developed by (Hupkes et al., 2018). The idea is to use the output of a hidden layer as an input for a shallow logistic classifier that will be used to try and predict a morphological or syntactic hypothesis. If the classifier is able to predict the correct feature to a high accuracy, then we can conclude that the network is learning that feature. We will use this method to further our understanding of how language models capture morphological and syntactic features for new languages.

2 Plan

The plan for the project is the following:

1. Gather data from Wikipedia for the four languages. The size of documents for each language should be in the range of 100K paragraphs to 1M paragraph depending on the availability of computational power to train the language models.

2. Train character-based RNN, GRU and LSTM and word-based LSTM as language models for the four different languages.

3. Decide on two morphological features for the languages like word classes (verbs and nouns) and numbers.

4. Generate a small dataset for the morphological features for each language by using a part of speech tagger to generate 1000 verbs and 1000 nouns, while maintaining conditions that will eliminate orthographic cues from the words. For example not use verbs that ends with *-ing* as they will help the network learn verb meaning just from that.

5. Develop a diagnostic classifier, train it on a small set from the dataset and test it against a larger set. For example train it on 100 verbs and 100 nouns and test it against 900 verbs and nouns.

6. Compare the results of accuracy between character-level RNN, GRU, and LSTM, the word-level LSTM, and baseline models.

7. Generate a small dataset for syntactic dependency for each of the languages in the magnitude of 1000. An example of syntactic dependency for Italian is article-noun gender agreement.

8. Develop a diagnostic classifier for the syntactic dependency, train it and test it for all different languages model.

9. Explore the effect of longer sentence on agreement across different networks.

10. Collect and visualize the results.

11. Write and document the previous steps in a report.

Each teammate will be responsible for developing a similar project for each language. Abed will be responsible for Arabic, Francesca Italian, Priyanka Hindi, and Anusha English.

3 Data and Tools

- The dataset for training the language models will be extracted from wikipedia using Wikiextractor (Attardi, 2012).

- We can use Stanza (Qi et al., 2020) for POS for all English, Italian, Hindi and Arabic. In case there was a problem with the toolkit we will use the following options as an alternative:

  - For English and Italian we will use spacy (Honnibal and Montani, 2017) or TreeTagger (Schmid, 1994) for part of speech tagging.
  - For Hindi we will use Natural Language Processing Toolkit (NLTK) (Loper and Bird, 2002) for part of speech tagging
  - For Arabic we will use CAMeL tools (Obeid et al., 2020).

- For the diagnostic classifier we will use something similar to the code developed by (Hupkes et al., 2018).

- For developing the RNNs we will use either Tensorflow (Abadi et al., 2016) or Pytorch (Paszke et al., 2019).

- For the dataset for training and testing the diagnostic classifier we will use the universal dependency treebank (Nivre et al., 2016).

- The rest of the coding will be done using Python and its related libraries like Numpy, Pandas and Matplotlib.

4 Literature Review

With the recent improvements in the performance of RNNs there has been an increasing interest in understanding what representations these networks learn about linguistic knowledge. The research uses character-level based neural networks because they do not require tokenization and due to the contested nature of deciding what constitutes a word.

In (Haspelmath, 2011) the authors show that there is no good criteria for defining the concept of ”word” cross-linguistically. This motivated researchers to explore neural networks that do not use tokenization as an input for training their models. One of the most prominent uses was the use of characters as an input to a Convolutional Neural Network (CNN) and a highway network, whose output is then fed to an LSTM language model (Kim et al., 2015).

The success of these character-based models combined with the fact that there is no good definition for what is a ”word” motivated research into looking what and how character-based models understand linguistic information. There are multiple probing techniques used to understand the previous character-based language models, for example (Hahn and Baroni, 2019) the researchers did use the ’diagnostic classifier’ to probe linguistic knowledge. The paper showed that character-based RNNs are able to solve morphological, syntactic and semantic tasks. In addition to that, the networks were able to learn word boundaries even when the networks were trained without white-spaces.

Another paper that probed neural language models showed that these models learn part of speech (POS) first and later learn more global information (Saphra and Lopez, 2018). In (Vig and Belinkov, 2019) the researchers demonstrated how attention in transformer language models interacts with syntax, and the results suggest that attention heads ”specialize” in POS tags. In (Arras et al., 2019) the authors compare various revelance-based explainability methods and test the usefulness of such relevances as representations. The authors find that gradient-based methods and layer-wise relevance propagation (LRP) result in good explanations of input relevance for a simple arithmetic task, and that linear combinations of word-embeddings weighted by their relevance (as measured by LRP) result in useful sentence-level representations.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. 2019. Evaluating recurrent neural network explanations.

Giuseppe Attardi. 2012. Wikiextractor.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*.

Michael Hahn and Marco Baroni. 2019. Tabula nearly rasa: Probing the linguistic knowledge of character-level neural language models trained on unsegmented text. *Transactions of the Association for Computational Linguistics*, 7:467–484.

Martin Haspelmath. 2011. The indeterminacy of word segmentation and the nature of morphology and syntax. *Folia linguistica*, 45(1):31–80.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615*.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Edward Loper and Steven Bird. 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. Camel tools: An open source python toolkit for arabic natural language processing. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7022–7032.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Naomi Saphra and Adam Lopez. 2018. Language models learn pos first. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 328–330.

Helmut Schmid. 1994. Treetagger-a language independent part-of-speech tagger. *http://www. ims. unistuttgart. de/projekte/corplex/TreeTagger/*.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*.