CS 6320: Natural Language Processing

Project 1: Sentiment Classification with Deep Learning

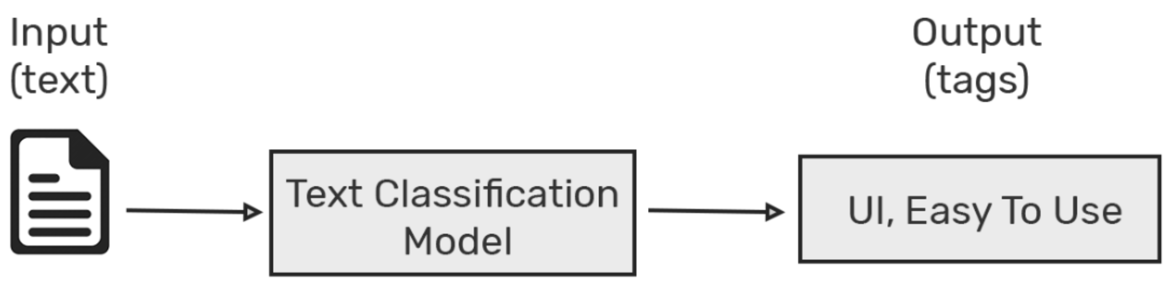
Shuang Wang sxw171930

Introduction:

Text Classification:

Text classification is the process of assigning tags or categories to text according to its content. It's one of the fundamental tasks in Natural Language Processing (NLP) with broad applications such as sentiment analysis, topic labeling, spam detection, and intent detection.

Text classification (a.k.a. text categorization or text tagging) is the task of assigning a set of predefined categories to free-text. Text classifiers can be used to organize, structure, and categorize pretty much anything. For example, new articles can be organized by topics, support tickets can be organized by urgency, chat conversations can be organized by language, brand mentions can be organized by sentiment, and so on.



Deep Learning:

Deep learning is a set of algorithms and techniques inspired by how the human brain works. Text classification has benefited from the recent resurgence of deep learning architectures due to their potential to reach high accuracy with less need of engineered features. The two main deep learning architectures used in text classification are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

On the one hand, deep learning algorithms require much more training data than traditional machine learning algorithms, i.e. at least millions of tagged examples. On the other hand, traditional machine learning algorithms such as SVM and NB reach a certain threshold where adding more training data doesn't improve their accuracy. In contrast, deep learning classifiers continue to get better the more data you feed them with.

Recurrent Neural Network for Specific-Task Text Classification:

There are several kinds of models to model text, such as Neural Bag-of-Words (NBOW) model, recurrent neural network (RNN) [Chung et al., 2014], recursive neural network (RecNN) [Socher et al., 2012; Socher et al., 2013] and convolutional neural network (CNN) [Collobert et al., 2011; Kalchbrenner et al., 2014]. These models take as input the embeddings of words in the text sequence, and summarize its meaning with a fixed length vectorial representation. Among them, recurrent neural networks (RNN) are one of the most popular architectures used in NLP problems because their recurrent structure is very suitable to process the variable-length text.

Method:

Dataset:

• IMDB The IMDB dataset consists of 100,000 movie reviews with binary classes [Maas et al., 2011]. One key aspect of this dataset is that each movie review has several sentences.

• Packages:

Numpy, Pandas, NLTK, Word2Vec, Keras, Sk-learn.

Steps:

1. Load the data:
2. Build the Embedding dictionary:
3. Create a TensorDataset and DataLoader
   (above output files are in 'output' folder)
4. Define the baseline model
5. Add an RNN layer to the baseline model
6. Replace the RNN by self-attention
7. Train and test the model
8. CNN test

Result:

Vanilla RNN vs GRU vs LSTM:

All RNNs have feedback loops in the recurrent layer. This lets them maintain information in 'memory' over time. However, it can be difficult to train standard RNNs to solve problems that require learning long-term temporal dependencies. This is because the gradient of the loss function decays exponentially with time (called the vanishing gradient problem). LSTM networks are a type of RNN that uses special units in addition to standard units. LSTM units include a 'memory cell' that can maintain information in memory for long periods of time. A set of gates is used to control when information enters the memory, when it's output, and when it's forgotten. This architecture lets them learn longer-term dependencies. GRUs are similar to LSTM, but it uses a simplified structure. They also use a set of gates to control the flow of information, but they don't use separate memory cells, and they use fewer gates.

Adding more layers not always result in more accuracy in neural networks. Adding more layers will help you to extract more features. But we can do that up to a certain extent. There is a limit. After that, instead of extracting features, it will tend to 'overfit' the data. Overfitting can lead to errors in some or the other form like false positives.

According to the information, Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems.

A neural network armed with an attention mechanism can actually understand what "it" is referring to. That is, it knows how to disregard the noise and focus on what's relevant, how to connect two related words that in themselves do not carry markers pointing to the other.

In terms of computational complexity, self-attention layers are faster than recurrent layers when the sequence length n is smaller than the representation dimensionality d, which is most often the case with sentence representations used by state-of-the-art models in machine translations

Reference:

[Chung et al., 2014] Junyoung Chung, Caglar Gulcehre,, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[Socher et al., 2012] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of EMNLP, pages 1201–1211, 2012.

[Socher et al., 2013] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over r a sentiment treebank. In Proceedings of EMNLP, 2013.

[Collobert et al., 2011] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and ´ Pavel Kuksa. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 12:2493–2537, 2011.

[Kalchbrenner et al., 2014] Nal Kalchbrenner, Edward network for modelling sentences. In Proceedings of ACL, 2014.

[Ashish Vaswani, Noam Shazeer et al., 2017] Illia Polosukhin, Attention Is All You Need, arXiv.org > cs > arXiv:1706.03762, 2017

External links:

https://blog.csdn.net/omnispace/article/details/100660324

https://keras.io/zh/layers/wrappers/

https://keras.io/zh/layers/recurrent/#rnn

https://keras.io/zh/activations/

https://www.cnblogs.com/xiaosongshine/p/10929934.html