
Deep-Learning-NLP Documentation

Release 1.0

Amirsina Torfi

Mar 10, 2020

1	Introduction	1
1.1	Motivation	1
2	Papers	3
2.1	Data Representation	3
2.2	Applications	5
3	Courses	11
4	Books	13
5	Blogs	15
6	Tutorials	17
7	Datasets	19
7.1	General	19
7.2	Text classification	19
7.3	Question Answering	19
7.4	Sentiment Analysis	20
7.5	Machine Translation	20
7.6	Summarization	20
8	Contributing	21
8.1	Pull Request Process	21
8.2	Final Note	21
9	Contributor Covenant Code of Conduct	23
9.1	Our Pledge	23
9.2	Our Standards	23
9.3	Our Responsibilities	24
9.4	Scope	24
9.5	Enforcement	24
9.6	Attribution	24
10	LICENSE	25

The purpose of this project is to introduce a shortcut to developers and researcher for finding useful resources about Deep Learning for Natural Language Processing.

1.1 Motivation

There are different motivations for this open source project.

1.1.1 What's the point of this open source project?

There other similar repositories similar to this repository and are very comprehensive and useful and to be honest they made me ponder if there is a necessity for this repository!

The point of this repository is that the resources are being targeted. The organization of the resources is such that the user can easily find the things he/she is looking for. We divided the resources to a large number of categories that in the beginning one may have a headache!!! However, if someone knows what is being located, it is very easy to find the most related resources. Even if someone doesn't know what to look for, in the beginning, the general resources have been provided.



This chapter is associated with the papers published in NLP using deep learning.

2.1 Data Representation

2.1.1 One-hot representation

- **Character-level convolutional networks for text classification** : Promising results by the use of one-hot encoding possibly due to their character-level information. [[Paper link](#) , [Torch implementation](#) , [TensorFlow implementation](#) , [Pytorch implementation](#)]



- **Effective Use of Word Order for Text Categorization with Convolutional Neural Networks** : Exploiting the 1D structure (namely, word order) of text data for prediction. [[Paper link](#) , [Code implementation](#)]



- **Neural Responding Machine for Short-Text Conversation** : Neural Responding Machine has been proposed to generate content-wise appropriate responses to input text. [[Paper link](#) , [Paper summary](#)]



2.1.2 Continuous Bag of Words (CBOW)

- **Distributed Representations of Words and Phrases and their Compositionality** : Not necessarily about CBOWs but the techniques represented in this paper can be used for training the continuous bag-of-words model. [[Paper link](#) , [Code implementation 1](#) , [Code implementation 2](#)]



2.1.3 Word-Level Embedding

- **Efficient Estimation of Word Representations in Vector Space** : Two novel model architectures for computing continuous vector representations of words. [[Paper link](#) , [Official code implementation](#)]



- **GloVe: Global Vectors for Word Representation** : Combines the advantages of the two major models of global matrix factorization and local context window methods and efficiently leverages the statistical information of the content. [[Paper link](#) , [Official code implementation](#)]



- **Skip-Thought Vectors** : Skip-thought model applies word2vec at the sentence-level. [[Paper](#) , [Code implementation](#) , [TensorFlow implementation](#)]



2.1.4 Character-Level Embedding

- **Learning Character-level Representations for Part-of-Speech Tagging** : CNNs have successfully been utilized for learning character-level embedding. [[Paper link](#)]



- **Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts** : A new deep convolutional neural network has been proposed for exploiting the character- to sentence-level information for sentiment analysis application on short texts. [[Paper link](#)]



- **Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation** : The usage of two LSTMs operate over the characters for generating the word embedding [[Paper link](#)]



- **Improved Transition-Based Parsing by Modeling Characters instead of Words with LSTMs** : The effectiveness of modeling characters for dependency parsing. [[Paper link](#)]



2.2 Applications

2.2.1 Part-Of-Speech Tagging

- **Learning Character-level Representations for Part-of-Speech Tagging** : A deep neural network (DNN) architecture that joins word-level and character-level representations to perform POS tagging [[Paper](#)]



- **Bidirectional LSTM-CRF Models for Sequence Tagging** : A variety of neural network based models have been proposed for sequence tagging task. [[Paper](#), [Code Implementation 1](#), [Code Implementation 2](#)]



- **Globally Normalized Transition-Based Neural Networks** : Transition-based neural network model for part-of-speech tagging. [[Paper](#)]



2.2.2 Parsing

- **A fast and accurate dependency parser using neural networks** : A novel way of learning a neural network classifier for use in a greedy, transition-based dependency parser. [[Paper](#), [Code Implementation 1](#)]



- **Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations** : A simple and effective scheme for dependency parsing which is based on bidirectional-LSTMs. [[Paper](#)]



- **Transition-Based Dependency Parsing with Stack Long Short-Term Memory** : A technique for learning representations of parser states in transition-based dependency parsers. [[Paper](#)]



- **Deep Biaffine Attention for Neural Dependency Parsing** : Using neural attention in a simple graph-based dependency parser. [[Paper](#)]



- **Joint RNN-Based Greedy Parsing and Word Composition** : A greedy parser based on neural networks, which leverages a new compositional sub-tree representation. [[Paper](#)]



2.2.3 Named Entity Recognition

- **Neural Architectures for Named Entity Recognition** : Bidirectional LSTMs and conditional random fields for NER. [Paper]



- **Boosting named entity recognition with neural character embeddings** : A language-independent NER system that uses automatically learned features. [Paper]



- **Named Entity Recognition with Bidirectional LSTM-CNNs** : A novel neural network architecture that automatically detects word- and character-level features. [Paper]



2.2.4 Semantic Role Labeling

- **End-to-end learning of semantic role labeling using recurrent neural networks** : The use of deep bi-directional recurrent network as an end-to-end system for SRL. [Paper]



2.2.5 Text classification

- **Convolutional Neural Networks for Sentence Classification** : By training the model on top of the pretrained word-vectors through finetuning, considerable improvement has been reported for learning task-specific vectors. [Paper link , Code implementation 1, Code implementation 2, Code implementation 3, Code implementation 4]



- **A Convolutional Neural Network for Modelling Sentences** : Dynamic Convolutional Neural Network (DCNN) architecture, which technically is the CNN with a dynamic k-max pooling method, has been proposed for capturing the semantic modeling of the sentences. [Paper link , Code implementation]



- **Very Deep Convolutional Networks for Text Classification** : The Very Deep Convolutional Neural Networks (VDCNNs) has been presented and employed at character-level with the demonstration of the effectiveness of the network depth on classification tasks [Paper link]



- **Character-level convolutional networks for text classification** : The character-level representation using CNNs investigated which argues the power of CNNs as well as character-level representation for language-agnostic text classification. [Paper link , Torch implementation , TensorFlow implementation , Pytorch implementation]



- **Multichannel Variable-Size Convolution for Sentence Classification** : Multichannel Variable Size Convolutional Neural Network (MV-CNN) architecture Combines different version of word-embeddings in addition to employing variable-size convolutional filters and is proposed in this paper for sentence classification. [[Paper link](#)]



- **A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification** : A practical sensitivity analysis of CNNs for exploring the effect of architecture on the performance, has been investigated in this paper. [[Paper link](#)]



- **Generative and Discriminative Text Classification with Recurrent Neural Networks** : RNN-based discriminative and generative models have been investigated for text classification and their robustness to the data distribution shifts has been claimed as well. [[Paper link](#)]



- **Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval** : An LSTM-RNN architecture has been utilized for sentence embedding with special superiority in a defined web search task. [[Paper link](#)]



- **Hierarchical attention networks for document classification** : Hierarchical Attention Network (HAN) has been presented and utilized to capture the hierarchical structure of the text by two word- level and sentence-level attention mechanism. [[Paper link](#) , [Code implementation 1](#) , [Code implementation 2](#) , [Code implementation 3](#), [Summary 1](#), [Summary 2](#)]



- **Recurrent Convolutional Neural Networks for Text Classification** : The combination of both RNNs and CNNs is used for text classification which technically is a recurrent architecture in addition to max-pooling with an effective word representation method and demonstrates superiority compared to simple windows-based neural network approaches. [[Paper link](#) , [Code implementation 1](#) , [Code implementation 2](#) , [Summary](#)]



- **A C-LSTM Neural Network for Text Classification** : A unified architecture proposed for sentence and document modeling for classification. [[Paper link](#)]



2.2.6 Sentiment Analysis

- **Domain adaptation for large-scale sentiment classification: A deep learning approach** : A deep learning approach which learns to extract a meaningful representation for each online review. [[Paper link](#)]



- **Sentiment analysis: Capturing favorability using natural language processing** : A sentiment analysis approach to extract sentiments associated with polarities of positive or negative for specific subjects from a document. [Paper link]



- **Document-level sentiment classification: An empirical comparison between SVM and ANN** : A comparison study. [Paper link]



- **Learning semantic representations of users and products for document level sentiment classification** : Incorporating of user- and product- level information into a neural network approach for document level sentiment classification. [Paper]



- **Document modeling with gated recurrent neural network for sentiment classification** : A neural network model has been proposed to learn vector-based document representation. [Paper, Implementation]



- **Semi-supervised recursive autoencoders for predicting sentiment distributions** : A novel machine learning framework based on recursive autoencoders for sentence-level prediction. [Paper]



- **A convolutional neural network for modelling sentences** : A convolutional architecture adopted for the semantic modelling of sentences. [Paper]



- **Recursive deep models for semantic compositionality over a sentiment treebank** : Recursive Neural Tensor Network for sentiment analysis. [Paper]



- **Adaptive recursive neural network for target-dependent twitter sentiment classification** : AdaRNN adaptively propagates the sentiments of words to target depending on the context and syntactic relationships. [Paper]



- **Aspect extraction for opinion mining with a deep convolutional neural network** : A deep learning approach to aspect extraction in opinion mining. [Paper]



2.2.7 Machine Translation

- **Learning phrase representations using RNN encoder-decoder for statistical machine translation** : The proposed RNN Encoder–Decoder with a novel hidden unit has been empirically evaluated on the task of machine translation. [Paper, Code, Blog post]

- **Sequence to Sequence Learning with Neural Networks** : A showcase of NMT system is comparable to the traditional pipeline by Google. [Paper, Code]

- **Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation** : This work presents the design and implementation of GNMT, a production NMT system at Google. [Paper, Code]

- **Neural Machine Translation by Jointly Learning to Align and Translate** : An extension to the encoder–decoder model which learns to align and translate jointly by attention mechanism. [Paper]

- **Effective Approaches to Attention-based Neural Machine Translation** : Improvement of attention mechanism for NMT. [Paper, Code]

- **On the Properties of Neural Machine Translation: Encoder-Decoder Approaches** : Analyzing the properties of the neural machine translation using two models; RNN Encoder–Decoder and a newly proposed gated recursive convolutional neural network. [Paper]

- **On Using Very Large Target Vocabulary for Neural Machine Translation** : A method that allows to use a very large target vocabulary without increasing training complexity. [Paper]

- **Convolutional sequence to sequence learning** : An architecture based entirely on convolutional neural networks. [Paper, Code[Torch], Code[Pytorch], Post]

- **Attention Is All You Need** : The Transformer: a novel neural network architecture based on a self-attention mechanism. [Paper, Code, Accelerating Deep Learning Research with the Tensor2Tensor Library, Transformer: A Novel Neural Network Architecture for Language Understanding]

2.2.8 Summarization

- **A Neural Attention Model for Abstractive Sentence Summarization** : A fully data-driven approach to abstractive sentence summarization based on a local attention model. [Paper, Code, A Read on “A Neural Attention Model for Abstractive Sentence Summarization”, Blog Post, Paper notes,]

- **Get To The Point: Summarization with Pointer-Generator Networks** : A novel architecture that augments the standard sequence-to-sequence attentional model by using a hybrid pointer-generator network that may copy words from the source text via pointing and using coverage to keep track of what has been summarized. [Paper, Code, Video, Blog Post]



- **Abstractive Sentence Summarization with Attentive Recurrent Neural Networks** : A conditional recurrent neural network (RNN) based on convolutional attention-based encoder which generates a summary of an input sentence. [Paper]



- **Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond** : Abstractive text summarization using Attentional Encoder-Decoder Recurrent Neural Networks [Paper]



- **A Deep Reinforced Model for Abstractive Summarization** : A neural network model with a novel intra-attention that attends over the input and continuously generated output separately, and a new training method that combines standard supervised word prediction and reinforcement learning (RL). [Paper]



2.2.9 Question Answering

- **Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks** : An argue for the usefulness of a set of proxy tasks that evaluate reading comprehension via question answering. [Paper]



- **Teaching Machines to Read and Comprehend** : addressing the lack of real natural language training data by introducing a novel approach to building a supervised reading comprehension data set. [Paper]



- **Ask Me Anything Dynamic Memory Networks for Natural Language Processing** : Introducing the dynamic memory network (DMN), a neural network architecture which processes input sequences and questions, forms episodic memories, and generates relevant answers [Paper]



CHAPTER 3

Courses



- **Natural Language Processing with Deep Learning** by Stanford : [[Link](#)]
- **Deep Natural Language Processing** by the University of Oxford: [[Link](#)]
- **Natural Language Processing with Deep Learning in Python** by Udemy: [[Link](#)]
- **Natural Language Processing with Deep Learning** by Coursera: [[Link](#)]

CHAPTER 4

Books



- **Speech and Language Processing** by Dan Jurafsky and James H. Martin at stanford: [[Link](#)]
- **Neural Network Methods for Natural Language Processing** by Yoav Goldberg: [[Link](#)]
- **Deep Learning with Text: Natural Language Processing (Almost) from Scratch with Python and spaCy** by Patrick Harrison, Matthew Honnibal: [[Link](#)]
- **Natural Language Processing with Python** by Steven Bird, Ewan Klein, and Edward Loper: [[Link](#)]



- **Understanding Convolutional Neural Networks for NLP** by Denny Britz: [\[Link\]](#)
- **Deep Learning, NLP, and Representations** by Matthew Honnibal: [\[Link\]](#)
- **Embed, encode, attend, predict: The new deep learning formula for state-of-the-art NLP models** by Sebastian Ruder: [\[Link\]](#)
- **Embed, encode, attend, predict: The new deep learning formula for state-of-the-art NLP models** by Sebastian Ruder: [\[Link\]](#)
- **Natural Language Processing** by Sebastian Ruder: [\[Link\]](#)
- **Probably Approximately a Scientific Blog** by Vered Schwartz: [\[Link\]](#)
- **NLP news** by Sebastian Ruder: [\[Link\]](#)
- **Deep Learning for Natural Language Processing (NLP): Advancements & Trends:** [\[Link\]](#)
- **Neural Language Modeling From Scratch:** [\[Link\]](#)



- **Understanding Natural Language with Deep Neural Networks Using Torch** by NVIDIA: [\[Link\]](#)
- **Deep Learning for NLP with Pytorch** by Pytorch: [\[Link\]](#)
- **Deep Learning for Natural Language Processing: Tutorials with Jupyter Notebooks** by Jon Krohn: [\[Link\]](#)

7.1 General

- **1 Billion Word Language Model Benchmark:** The purpose of the project is to make available a standard training and test setup for language modeling experiments: [\[Link\]](#)
- **Common Crawl:** The Common Crawl corpus contains petabytes of data collected over the last 7 years. It contains raw web page data, extracted metadata and text extractions: [\[Link\]](#)
- **Yelp Open Dataset:** A subset of Yelp’s businesses, reviews, and user data for use in personal, educational, and academic purposes: [\[Link\]](#)

7.2 Text classification

- **20 newsgroups** The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups: [\[Link\]](#)
- **Broadcast News** The 1996 Broadcast News Speech Corpus contains a total of 104 hours of broadcasts from ABC, CNN and CSPAN television networks and NPR and PRI radio networks with corresponding transcripts: [\[Link\]](#)
- **The wikitext long term dependency language modeling dataset:** A collection of over 100 million tokens extracted from the set of verified Good and Featured articles on Wikipedia. : [\[Link\]](#)

7.3 Question Answering

- **Question Answering Corpus** by Deep Mind and Oxford which is two new corpora of roughly a million news stories with associated queries from the CNN and Daily Mail websites. [\[Link\]](#)
- **Stanford Question Answering Dataset (SQuAD)** consisting of questions posed by crowdworkers on a set of Wikipedia articles: [\[Link\]](#)

- **Amazon question/answer data** contains Question and Answer data from Amazon, totaling around 1.4 million answered questions: [\[Link\]](#)

7.4 Sentiment Analysis

- **Multi-Domain Sentiment Dataset** The Multi-Domain Sentiment Dataset contains product reviews taken from Amazon.com from many product types (domains): [\[Link\]](#)
- **Stanford Sentiment Treebank Dataset** The Stanford Sentiment Treebank is the first corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language: [\[Link\]](#)
- **Large Movie Review Dataset:** This is a dataset for binary sentiment classification: [\[Link\]](#)

7.5 Machine Translation

- **Aligned Hansards of the 36th Parliament of Canada** dataset contains 1.3 million pairs of aligned text chunks: [\[Link\]](#)
- **Europarl: A Parallel Corpus for Statistical Machine Translation** dataset extracted from the proceedings of the European Parliament: [\[Link\]](#)

7.6 Summarization

- **Legal Case Reports Data Set** as a textual corpus of 4000 legal cases for automatic summarization and citation analysis.: [\[Link\]](#)

For typos, please do not create a pull request. Instead, declare them in issues or email the repository owner. Please note we have a code of conduct, please follow it in all your interactions with the project.

8.1 Pull Request Process

Please consider the following criterions in order to help us in a better way:

1. The pull request is mainly expected to be a link suggestion.
2. Please make sure your suggested resources are not obsolete or broken.
3. Ensure any install or build dependencies are removed before the end of the layer when doing a build and creating a pull request.
4. Add comments with details of changes to the interface, this includes new environment variables, exposed ports, useful file locations and container parameters.
5. You may merge the Pull Request in once you have the sign-off of at least one other developer, or if you do not have permission to do that, you may request the owner to merge it for you if you believe all checks are passed.

8.2 Final Note

We are looking forward to your kind feedback. Please help us to improve this open source project and make our work better. For contribution, please create a pull request and we will investigate it promptly. Once again, we appreciate your kind feedback and elaborate code inspections.

Contributor Covenant Code of Conduct

9.1 Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, gender identity and expression, level of experience, nationality, personal appearance, race, religion, or sexual identity and orientation.

9.2 Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

9.3 Our Responsibilities

Project maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

9.4 Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

9.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the project team at amirsina.torfi@gmail.com. The project team will review and investigate all complaints, and will respond in a way that it deems appropriate to the circumstances. The project team is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Project maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

9.6 Attribution

This Code of Conduct is adapted from the [Contributor Covenant][homepage], version 1.4, available at [<http://contributor-covenant.org/version/1/4/>][version]

[homepage]: <http://contributor-covenant.org> [version]: <http://contributor-covenant.org/version/1/4/>

CHAPTER 10

LICENSE

MIT License

Copyright (c) 2017 Amirsina Torfi

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.