Exercises
Natural Language Understanding
Spring 2019

Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Prof. Dr. Thomas Hofmann
Dr. Massimiliano Ciaramita
Web http://www.da.inf.ethz.ch/teaching/2019/NLU/

Project Description (Project 2)

Please post any questions you may have on piazza.

1 Predicting the End of a Story (60 points)

The Turing test has ever since highlighted the ability to communicate using natural language as a key signifier of artificial intelligence. With a surge of new research [1] on chat bots and conversational agents progress on this challenge is more sought-after than ever. However, evaluating such systems automatically and realistically is an unsolved problem [2].

As a consequence, the NLP community has focused on designing new tasks that can serve as a proxy to an actual communication task [3, 6]. Solving these tasks will require applying logical reasoning, connecting distant facts and including common sense; But evaluation can be done by a metric computed automatically, such as a classification error.

1.1 The Story Cloze Task

We will focus on the recently proposed Story Cloze Task [4]. In a nutshell, you are given a four-sentence short story and two alternatives for the last, fifth sentence that ends the story. Your system needs to decide which of the two is correct. The training data consists of five-sentence short stories that include the correct ending *but there is no incorrect ending given*. The validation set of the original task ([5]), which contains positive and negative endings, is a lot smaller than the training set, therefore naively training a binary classifier on this data will give bad generalization performance. The training dataset has the following interesting properties:

1. All stories were generated manually.
2. All stories were manually verified to:
   (a) Have a realistic content.
   (b) Have an ending that is semantically the actual end of the story plot.
   (c) Have no redundant sentences.
3. The dataset is significantly larger than other datasets with a comparable quality.

The beauty of the task lies in that you can follow different methodologies to tackle it including generative systems (what ending is more likely to be generated?), some form of anomaly detection (ending A is further away from the story than ending B) or data augmentation (can I bootstrap a training set for a classifier?).

1.2 Task Description

1. Read the original paper about the Story Cloze task [6].
2. Implement a system that can solve the Story Cloze task (input/output format see below).
3. Train it using the training set [7].
4. Evaluate and report the accuracy of your method on the validation set [5].
5. Write a two to four pages report using this LaTeX template [8].
6. Process a test file, which will be distributed shortly before the submission deadline.

Guidelines for Model Design You are allowed to build on any existing idea as long as you cite it. If there is some key idea that you advertise as novel, you are required to have skimmed the relevant literature for it. You can make simplifying assumption at any point as long as you state them and ideally argue why they are reasonable. This might include computational reasons (we limit the vocabulary to . . .), assumptions about the nature of the data (we assume the last sentence . . .) or the expressiveness of the model (we use $K = 10$ clusters . . .).

Guidelines for Implementation You are allowed to use any Tensorflow code (again, cite where you found it) or technology that builds on top of Tensorflow.

1.3 Submission and grading

Model (70%) We will grade your model based on the creativity, appropriateness and effectiveness of your idea. In particular, we will measure you to some degree with respect to the hardness of the methodology (see above) your model falls into.

Report (30%) Furthermore, we expect your report to be written in clear, scientific English as you would expect it in a Master thesis. The purpose of the report is to

1. Motivate your idea.
2. Relate your idea to existing work and – possibly – highlight novelties.
3. Support the effectiveness of your model experimentally.

Make sure that you present your model (and ideally also the evaluation) in an incremental fashion. What is the plain-vanilla version of your model? What did you add on top of it? Note that the evaluation can contain much more detailed results than what we require as submission. What can your model do better? How is it related to your original motivation?

Formalities

- Deadline: Sunday, June 9th 2019, 23:59:59.
- Hand in
  - Your runnable python code (including a pip requirements file or a conda spec list) with a small README. (more details on this later)
  - Your report.
  - Output for a test file (to be distributed) in the following format:
    CSV file, no header, one entry per line and story which is either *1* or *2* and indicates which of the two provided candidate endings is labelled as correct.
  - You have to submit at `https://cmt3.research.microsoft.com/ETHZNLU2019`

1.4 Hints

Possible Directions You are in general free to follow any direction to designing a system solving the Story Cloze task. To give you some inspiration from our side, we provide some potential directions (we do not claim that this list is complete by any means). Many current approaches have already been summarized in a bit more detail in [9]. Note that they are neither equally complicated nor mutually exclusive:

- Extract linguistic features and feed them into a classifier
- Use techniques from sentiment analysis
- Use a generative model, similar to a Language Model.
- Focus only on parts of the stories, such as events, entities etc. (maybe some form of attention?).

- Find an advanced suitable embedding space for stories and endings, minimize some similarity metric.
- Data augmentation e.g. to obtain negative endings for the training set [10], [11].
- Train a Conditional Generative Adversarial Network (CGAN), then use its discriminator for validation and testing [12].
- Including external commonsense knowledge base yields better accuracy [13].
- There are some other tasks quite similar to the Story Cloze task, such as the Argument Reasoning Comprehension task [14]. Doing a quick literature search might give you some cool ideas.

Pitfalls

- We do not recommend to solely train on the Story Cloze validation set. Use the much bigger training set of stories we provide [7]. The more you rely on the validation set, the more prone you are to overfitting. If you anyway choose to go this route, make it very clear in the report.
- Work in incremental fashion. Start with some simple approach before you design a state-of-the-art algorithm.

Infrastructure

Same as for Project 1. You must use Tensorflow and we strongly recommend using python3. You have access to two compute resources: Unlimited CPU usage on Euler [15] and GPU usage on Leonhard [16]. Note that the difference in speed is typically a factor between 10 and 100.
Remember: For using Leonhard, no two students from the same team are allowed to run jobs at the same time! If you violate this rule we may revoke your access.

References

[1] http://aclweb.org/anthology/I17-5003

[2] https://web.stanford.edu/~jurafsky/slp3/29.pdf, subsection 29.4

[3] http://arxiv.org/abs/1508.05326

[4] http://cs.rochester.edu/nlp/rocstories/

[5] https://polybox.ethz.ch/index.php/s/02IVLdBAgVcsJAx/download

[6] https://arxiv.org/abs/1604.01696

[7] https://polybox.ethz.ch/index.php/s/l2wM4RIyI3pD7Tl/download

[8] https://polybox.ethz.ch/index.php/s/p7Hi8yjWawmJkc9/download

[9] http://www.aclweb.org/anthology/W17-0906

[10] http://www.aclweb.org/anthology/W17-0911

[11] https://pdfs.semanticscholar.org/cbb5/c7363e11c3302df7032cffdcf216d52bb36d.pdf

[12] https://www.ijcai.org/proceedings/2017/0576.pdf

[13] https://arxiv.org/pdf/1811.00625.pdf

[14] https://arxiv.org/abs/1708.01425

[15] http://brutuswiki.ethz.ch/brutus/EULER_for_beta_users#Useful_bsub_options

[16] https://scicomp.ethz.ch/wiki/Leonhard_beta_testing