Link to github & twitter-profil!

Task description

* Generative model for text (via softmax over sliding-window-language model)
* Discriminator for text (simple gradient descent)
* Recognizer for text (in contrast to random)
* …All of which using word2vec
* And using our own dataset, crawled and preprocessed via twitter
* This is general, one could provide any set of twitter profiles to learn on
* Always-on-shitposting-bot

Related work and similar approaches

* Text generaton:
  + SequenceGAN <https://arxiv.org/pdf/1609.05473.pdf> (code updated, ask me) https://github.com/dennybritz/deeplearning-papernotes/blob/master/notes/seq-gan.md
  + Uses the logic of the TF-Languagemodel <https://www.tensorflow.org/tutorials/recurrent>, but not on PTB, with a whole new iterator, of course the softmax-generator is new, decreasing learningrate, incorporation of word2vec, …
  + Other text-generators?
* Word2vec: taken from tf tutorial, only change lies in the batch-generator, because we don’t have one long text, but (blick mal durch den Zufall :o)
* Discriminator:
  + The original network behind the discriminator was implemented for the movie-reviews dataset (hence the occasionally “reviews” instead of “strings”) (only because getting the trump-data happened simultaneously and I needed a dataset)
  + Wir kommen an 90% bei dem dataset? Any papers was andere so Machen?

Theoretical basis and used procedures

* Dass pre-trained w2v das angeblich besser macht <https://arxiv.org/abs/1408.5882>

Network structure and design choices

* Automatisches kürzen und anpassen der string-lens
* Weight decay vs adam optimizer
* Saver mit #iterationen selbst, because no clue.

Performance evaluation and comparison

* Dass “4” und “for” nah sind bei twitter
* @realdonaldtrump #MAGA undso^^
* Hier den screenshot vom word2vec

Plots etc

* Of course, plotten tun wir!