Assignment 2:

Learning Distributional Semantics with the SGNS Model

**Submission​ ​deadline:​ ​23:59,​ May 21 (Sundayy, just ​before​ ​midnight)**

In this assignment you will implement the SkipGram model with Negative Sampling (SGNS).

**Objectives:**

1. Better understanding the training process based on word corpus.
2. Better understand the logistic regression and the use of Loss functions, Gradient descent and backpropagation.
3. Better understand distributional semantics, uses and limitations.

The purpose of this assignment is to give you hands-on experience with probabilistic models of language by implementing the full algorithmic pipeline.

**Specifics:**

* You should implement a class called SkipGram (API below)
* The SkipGram embeddings should be learned with negative sampling.
* You have to implement the API specified below - this is the API we will use.

As usual, you should add other methods as you see fit but you cannot assume these are called by an external driver.

API

Your​ ​code​ ​should​ ​work​ ​seamlessly​ ​with​ ​the​ ​following​ ​API:

Static functions:

who\_am\_i()

normalize\_text(fn)

load\_model(fn)

class SkipGram:

def \_\_init\_\_(self, sentences, d=100, neg\_samples=4, context=4,

word\_count\_threshold=5)

def compute\_similarity(self, w1, w2)

def get\_closest\_words(self, w, n)

def learn\_embeddings(self, step\_size=0.001, epochs=50,

early\_stopping=3, model\_path=None)

def combine\_vectors(self, T, C, combo=0, model\_path=None)

def find\_analogy(self, w1,w2,w3)

def test\_analogy(self, w1, w2, w3, w4, n=1)

See​ ​detailed​ ​specifications​ ​and​ ​documentation​ [here](https://www.dropbox.com/s/pqpxs6qp8vjtugt/ex2_api.py?dl=0).

## Corpora

Here​ ​are​ ​some​ ​corpora​ ​to​ ​use:

* [Dr. Seuss writings](https://www.dropbox.com/s/l8fluga2q32xk2h/drSeuss.txt?dl=0) (30KB) - **you should start with this small file!**
* [Harry Potter and the Sorcerer's Stone](https://www.dropbox.com/s/vk0anp2u0xzygne/harryPotter1.txt?dl=0) (450KB) (some funky encoding!)
* Norvig’s​ [​​big.txt](https://norvig.com/big.txt)​​ ​file (make sure to look at the file and its format)

​Feel​ ​free​ ​to​ experiment with ​other​ ​resources (e.g. nltk.corpus) or the corpora from Ex1 (given you have the time and hardware)

**Think** (no need to provide answers in the submission):

1. Extract the number of types and tokens in the different datasets. What are the differences between the datasets?
2. What is the effect of the embedding dimension?
3. How useful are embeddings learnt over a small dataset (better semantic representation over the dataset or that the representation is not very good because the dataset is small)?
4. Experiment with different types of T&C combinations. Do you see a difference?

## Efficiency

While this course is not about software engineering and code design, your code is expected to be reasonably efficient.

Even efficient implementation of the SGNS model can take over 12 hours on a decent size of a corpus.

Submission​ ​Guidelines

1. You should use the course Moodle to submit a **single** **gz** file containing a single python file. (and only gz!)
2. Your code file should be named **ex2.py**, and must contain all necessary methods/functions/classes that support the specified API.
3. Your code **shouldn’t print anything** to standard output! (if you print for testing, remember to set the relevant flag appropriately)
4. You should use **python 3.9**.
5. You can **only** import the following modules: pickle, os,time, re, sys, random, math, collections, nltk, numpy, pandas
6. You should **document your code** using either [Google Style](http://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html) or [Pyhton PEP 257](https://www.python.org/dev/peps/pep-0257/).

Sandbox and Tips

Training can take a long time (>12 hours):

1. It would be wise to save intermediate models

2. It is recommended to use the time module and track the training process (epoch? 5K sentences?)

3. It is recommended to track the (hopefully shrinking) loss.

You could also evaluate on a fixed list of Gold-Labeled pairs (think how)

4. It is recommended to start by training on a relatively small file with a limited vocabulary (and a repetitive structure)

Integrity​ ​and​ ​Cooperation

You​ ​should​ ​work​ ​on​ ​your​ ​assignments​ ​alone​ ​and​ ​refrain​ ​from​ ​sharing​ ​code​ ​snippets.​ ​However, you​ ​are​ ​encouraged​ ​to​ ​discuss​ ​various​ ​aspects​ ​of​ ​the​ ​assignment​ ​in​ ​the​ ​dedicated​ ​assignment forum​ ​and​ ​you​ ​are​ ​welcome​ ​to​ ​share​ ​testers​ ​and​ ​additional​ ​corpora.  
**After you submit your solutions may sample a small number of students for a short interview to discuss their implementation decisions.**