Guesso

- name: Guesso
- solves: 5
- points: 498
- categories
  - medium
  - AI13n
- attachments
  - challenge.py

> My friend is consistently engrossed in reading Google News, and lately he has been incessantly discussing the concept of dimensions—specifically, a staggering 300 of them. He fervently insists that words can be represented as vectors, a notion that seems perplexing at best. In the midst of his enthusiasm, he has even developed a game centered around guessing words. Can you solve his game?
>
> `nc dyn.ctf.pearlctf.in 30021`
>
> - challenge.py

solution

Use the `gensim.downloader` to download the `word2vec-google-news-300` word vector model.

Select 5 random words in the corpus and precompute the cosine similarities between the 5 words and all other words. Multiply the cosine similarities by 100 and round it off to 3 significant figures. Then, for each selected random word, create a reverse lookup table to lookup the possible other words from the cosine similarity.

Connect to the challenge server and send the 5 selected words in order. Lookup the possible other words from the returned cosine similarity. Intersect the possible other words with the other 4 selected words' possible other words. After 2 to 4 tries, you should have only 1 word as the possible candidate. Send that 1 word to get the flag. If it fails to get the flag, repeat the hack.

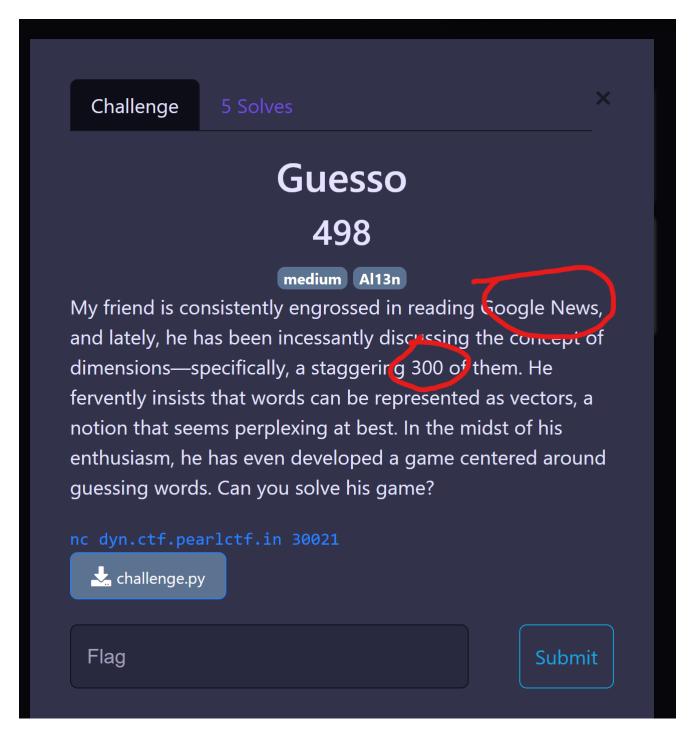Finally, the flag is `pearl{d4yumm_y0u_kn0w_y0ur_v3ct0rs}`.

See `hack.py` for a reference implementation.

process

From the provided source code `challenge.py`, it looks like we need to find the word using the returned cosine similarities only, as cosine similarities is the only useful information.

This would be easy if we have the model, but `MODEL_NAME` is blank, so we do not know the model...

Or do we? This is the most difficult part of this challenge. If you read the description carefully, hints can be found...

And if you have a bit of background knowledge about word-to-vector embedding, you could guess what you need to find. Searching `word2vec google news 300` online would yield the `word2vec-google-news-300` model. Since their source code is using `gensim`, we will also use the `gensim` method of getting the model. In particularly, I found out about the model on https://radimrehurek.com/gensim/models/word2vec.html#pretrained-models.

Now that we know the model name, we just need to code up the thing described in , and we are good to go! Well, except that you would need to spend quite some time coding up the thing, downloading the model (it's somewhat large), and debugging your code. Good luck!

One more thing, we know it is feasible to obtain the word using at most 4 cosine similarities only. Consider that there are only about ~150000 words in the vector model, and there are about ~2000 possible cosine similarities, we expect 2 to 3 cosine similarities suffice to single out 1 word from the model by combinational explosion.

Finally, the flag is `pearl{d4yumm_y0u_kn0w_y0ur_v3ct0rs}`.