## 1. Introduction

The main idea of this project is to create a word-guessing game inspired by Semantle. Just as Semantle is a word-guessing game that is designed to predict words based on their similarity in meaning, this project requires words to be guessed based on their semantic relation. It is a fun way to do some vocabulary practice.

Every time the user types a word they will receive a similarity score. This score is on a scale from -100 to 100, where 100 is the highest. If you type the correct word, the game will let the user know that the words are identical and congratulate them on winning. While playing the game, if typed *HINT* in capital letters, it retrieves the definition of the target word. If typed any digit e.g *5*, the 5th most similar word is retrieved from the data set, the same applies for any other number up to 500 typed by the player. Additionally, there is a *GIVE UP* command added to show the correct word and end the game, or start again if desired.

## 2. Material and methods

The main data comes from a file of pre-trained word vectors using fastText, taken from the following source: https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.vec.gz. Several options were considered until reaching this choice, first we tried to use Spacy's default English model on its medium sized version (en_core_web_md), but as the code progressed it organically made sense to use Word2Vec to process the semantic model mentioned above. We made sure no words shorter than three characters would be chosen for the correct word, as well as words that aren't available in fastText, since there would then be no hint-option for the word. We limited the distributional semantics model to 50000 words in order to work comfortably.

The main idea for this project was to replicate the word-guessing game Semantle, but our game has some new features that Semantle doesn't.

While Semantle has one hint-option, which gives the user a random word that is more similar to the correct one than their best guess, we have two hint-options. The first one lets the user choose to see any of the 500 most similar words by writing the number they are interested in in the guess-space. For example, if the user writes "50" they will see the 50th most similar word. The second hint-option lets the user see the definition of the word if they write HINT in capital letters in the guess-space. The definition is taken from the first meaning in the correct word's synset.

The original Semantle also has a "hot and cold" function that tells you if the word you guessed is cold, tepid or hot. We chose to go in a different direction and tell you how close each word is to the correct word with the similarity score, as well as the ranking if the word is one of the 500 closest ones.

The hints make the game easier and more enjoyable for the player, and the give-up option and the fact that you can play the game as many times as you want to frees the user from stress and dissatisfaction if the word feels too complicated.

## 3. Results

Our overall expectations were met. We succeeded in replicating Semantle and, even though we did not work on a visual interface for it (following teacher's instructions), we ended up having a well-functioning game. Luckily, we had some experience working with Word2Vec from our Computational Semantics class in the first semester. This made the whole process significantly easier. Otherwise, it might have taken longer to figure out the inner configuration of a vector model like this.

Nevertheless, one limitation of the *HINT* (and the synsets where it takes the hints from) is that the program takes the first definition from the synsets, but sometimes this might not have the same meaning as the one that the word vectors go for. In a hypothetical case where the word is *spit* and all the closest words are *spew*, *vomit*, *swallow* and etc., the definition from the synset is *a narrow point of land projecting into the sea* since apparently *spit* is a word that has several meanings. This could be fixed by comparing the list of closest neighbours (based on the word vectors) to the synonyms in the synset, and choosing another synset if they don't match. However, we felt like this was out of scope for this project. If we worked further on this project, we would like to adapt the same kind of code on a word bank of several other languages.

There were other many features that we would have loved to include, like the possibility of adapting it to different languages or excluding inappropriate words out of the dataset; however, we all feel like this serves as a proper approach to the replication of Semantle. If you enjoyed Wordle, we are sure that you will also enjoy playing our version of Semantle.

**Link to download the pre-trained word vectors:** https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.en.300.vec.gz (this file was too big to upload to github, it needs to be downloaded manually).

## References

Turner, D. (2022, 27 febrero). *Semantle*. https://semantle.com/.