Faculty of Computer science
Higher School of Economics

Development of guess a word game with NLP and optimal strategies search

Anwar Ibrahim, Fares Ghazzawi, Valentin Kopylov, Grigoriy Kryukov

Supervisor: Dmitry I. Ignatov

Contexto

- Contexto is word-based game
- players have to guess a word within an unlimited number of guesses
- With each guess, the game outputs a number which indicates the contextual relevance between the guess and the secret word.
- In this project, we produced a front-end Contexto game for three different languages using Bert models and NLP.

Arabic Language Task

Consists of two phases:

1. Find most common words: the words should be on their base form, in Modern Standard Arabic , without any Arabic diacritics and without any prefixes and suffixes.

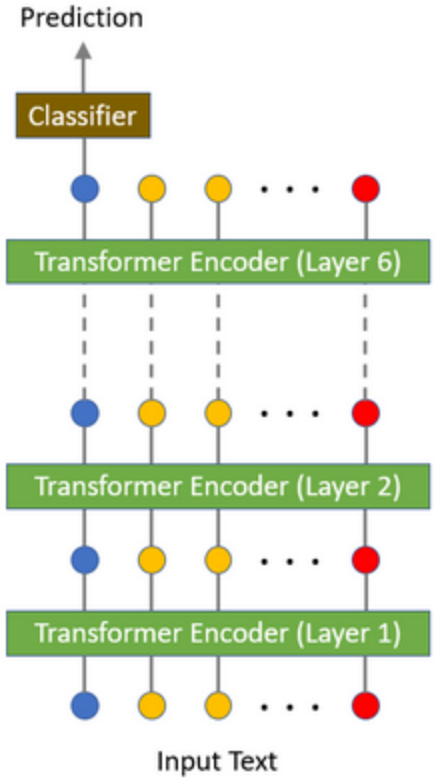
مرض	المرضية	المَرَضيْة
Without suffixes and prefixes (base form)	Without Arabic diacritics	With Arabi diacritics + suffixes and prefixes

2. Deploying a model: The model should be small in size , since large models might not be so useful in the this task and would worsen the performance of the front-end product, and it should be trained on texts from Modern Standard Arabic, models that are trained on tweets texts would contain in results a lot of dialectal Arabic.

Embedding vector calculation

For each language we aspired to acquire vector representation of the word

1. Baseline solution: we used pretrained base models, such as Word2Vec and Glove, for initial glimpse on the possible strategies of the game. For example, for English language, we used GloVe, trained on different datasets
2. Another approach: using transformer-based models in order to calculate vector representation.
   We used BERT and it's variants in English, Russian and Arabic languages
   Specifically, during inference stage, we acquired representations before classification step and averaged it across hidden dimension.



Proximity score calculation

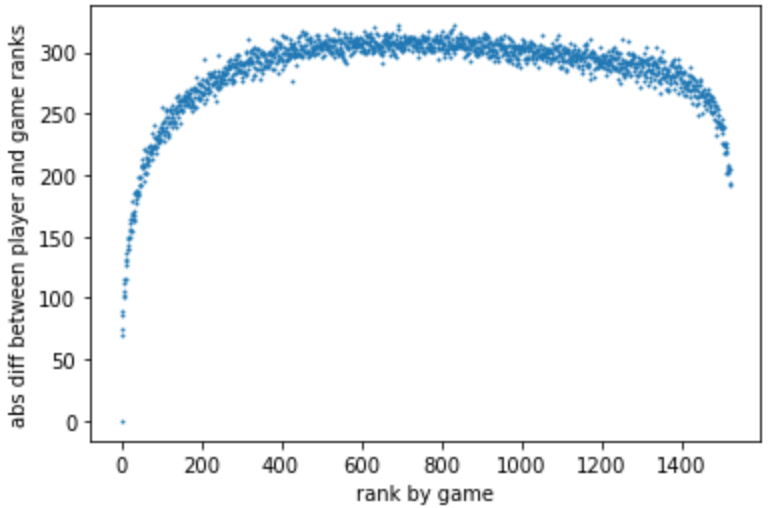
After gaining embedding vectors for the nouns:

1. We calculated their proximity using the cosine distance
2. Assessed difference between predictions of different versions of pretrained BERTs within the same language + additionally compared different subtasks' models

Player simulation

Let's simulate the game according to this plan:

1. Limit the number of words to the most popular (*N = 1522*, prepared by Anwar Ibrahim)
2. Train the game using the *glove-wiki-gigaword-100* model from gensim
3. The game will return to the player the rank where the player's word is close to the word game (*1* - as close as possible, *N-1* - as far as possible)
4. Create a player with your own ideas about the proximity of words (trained on the *glove-twitter-100* model from gensim)
5. Teach the player to play according to a certain strategy.



Random word guessing (RWG)

Let *X* be the number of attempts to guess the word, *N* is the number of words in the game.

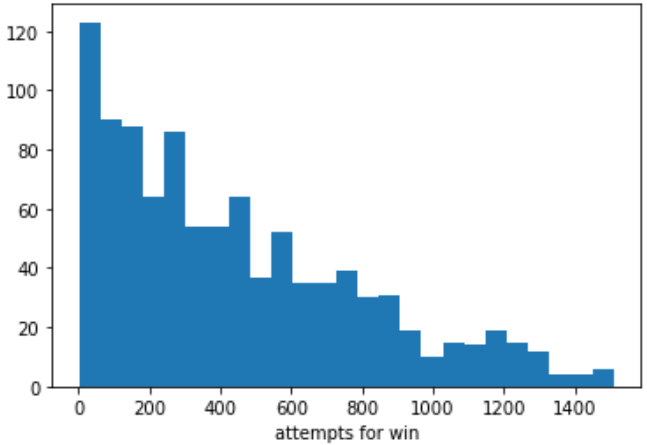
$$E[X] = = \frac{1 + 2 + \cdots + N}{N} = \frac{N(N+1)}{2N} = \frac{N+1}{2}$$

$$Var(X) = E[X^2] - (E[X])^2 = \frac{1^2 + 2^2 + \cdots + N^2}{N} - \frac{(N+1)^2}{4} =$$
$$= \frac{N(N+1)(2N+1)}{6N} - \frac{(N+1)^2}{4} = \frac{N^2 - 1}{12}$$

It is too long! On average, the player must use half of the words to win. Let's speed it up!

Player Rank Using (PRU)

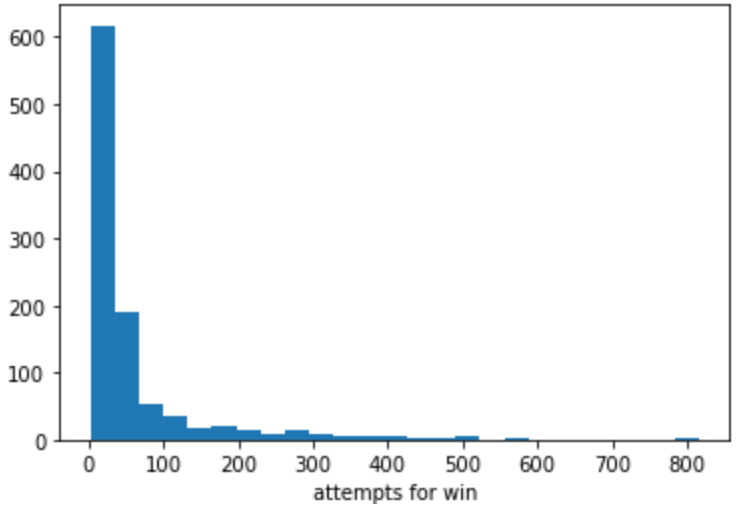
- On the first move, the player uses a random word
- After the player's $(i-1)$-th move, the game returns closeness rank k
- In the i-th move, the player names the unused word that has the rank closest to k relative to the $(i-1)$-th word



number of attempts to win	RWG	PRU
mean	761.5	442.01
std	439.36	356.09
median	761.5	353

Maximum likelihood using(MLU)

- On the first move, the player uses a random word
- Each i-th move, the player uses an unused word that has the minimum value of $\sum_{j=1}^{i-1} |k_j - w_j|$
  k_j is the rank of the game on the j-th move
  w_j is the rank of the player according to word w on the j-th move



number of attempts to win	RWG	PRU	MLU
mean	761.5	442.01	57.48
std	439.36	356.09	86.80
median	761.5	353	27

Q&A session