



**GDAŃSK UNIVERSITY  
OF TECHNOLOGY**

# **Assessing Word Difficulty for Quiz-like Game**

**12.09.2017r.**

**Jakub Jagoda  
Tomasz Maria Boiński  
*tobo@eti.pg.edu.pl***



## The Origin

1. The Colabmap project - we created a set of mappings between English Wikipedia articles and WordNet synsets.
2. Each mapping consists of a WordNet synset, definition of the synset and the title of Wikipedia article with special characters encoded using RFC 3986.
3. Such mappings, when proved to be correct, will allow formalization of Wikipedia structure.
4. The obtained set of mappings contained algorithmically created 54475 connections.



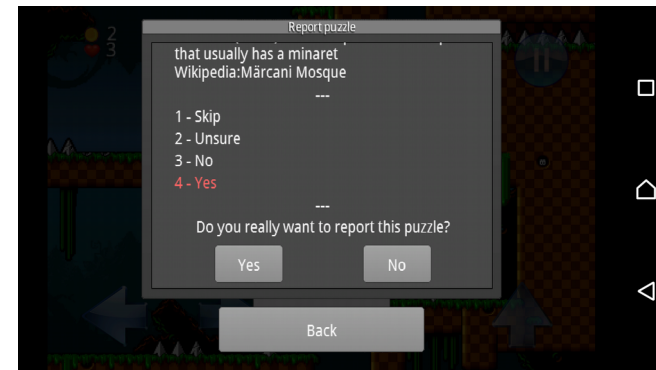
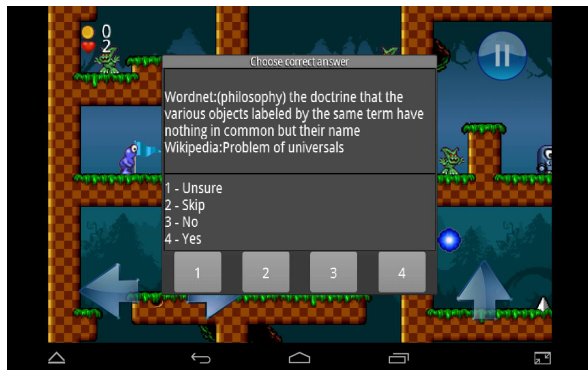
## Were the results good?

1. The accuracy of the mappings was of 73%
2. Which of them are actually correct and which are not?
3. We aim at creating 100% correct mappings corpora so the set required manual verification.



## TGame

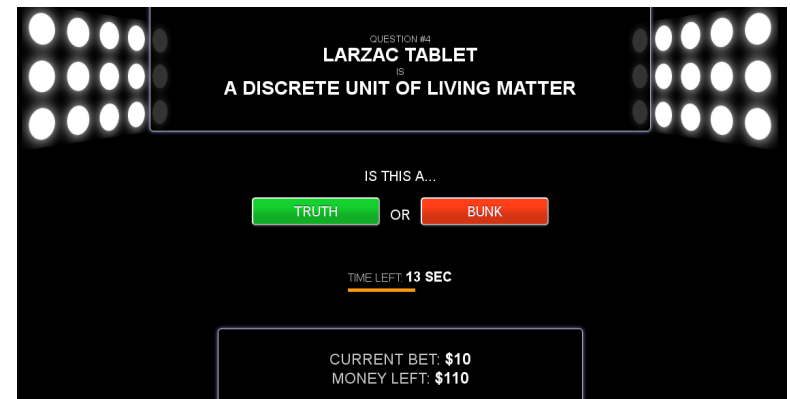
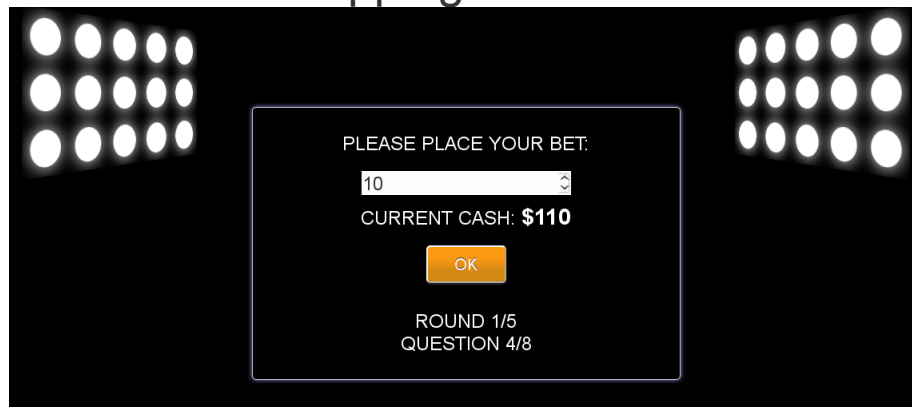
1. <https://kask.eti.pg.gda.pl/tgame>
2. A game with a purpose used to verify the mappings using the power of the crowd
3. A simple platform game for Android platform, available on Google Play
4. Checkpoint activation requires the player to solve a puzzle
5. Semi-easy error report possibility
6. Currently converted into a framework used for general verification of the results of heuristics algorithms





## New client

1. Wikipedia – WordNet mappings did not work very well with a platform game
2. We are developing a new game – Truth or Bunk (beta at <https://kask.eti.pg.gda.pl/tgame5/tob/>)
3. A quiz-like game better suiting the type of questions
4. The player is faced with 5 rounds of question that increase in difficulty
5. We assume that the mapping in our database is the correct one, if a lot of players fail the question we know that something might be wrong with the mapping





## What do we need?

1. To scale the word/definition pairs somehow
2. We assume the word that we ask for determine the difficulty
3. We have 10 difficulty levels in the game so we group the words in categories
4. Each category should contain similarly difficult words



## How do we do it?

1. The question actually is what do we mean by the difficulty of the word?
2. Is autotomy a difficult word? burnsides?  
pneumonoultramicroscopicsilicovolcanoconiosis? titin?  
Methionylthreonylthreonylglutaminylarginyl...isoleucine'?
3. It is all domain dependent but in general if we see or hear a word often than it can be considered easy for us – research show that frequency of a word might be correlated to its difficulty
4. We define difficulty of words by their frequency of occurrence
5. As a knowledge base we use Wikipedia (4 838 000 pages) as we try to assess the difficulty of the names of Wikipedia articles



## How do we do it? (2)

1. In general we count the number of occurrences of given word in Wikipedia – the more often it occurs the easiest it is
2. Words get additional penalty for length – the longer the word the more probable it is that it can be considered difficult
3. The above also makes the score values different for more words
4. We also take into account the overall difficulty of analyzed text – without that short, common words like “moo” can be treated as difficult
5. For that we use Flesch-Kincaid readability ease test that for each text applies the formula

$$fkScore = 206.835 - 1.015 * \left( \frac{totalwords}{totalsentences} \right) - 84.6 * \left( \frac{totalsyllables}{totalwords} \right)$$

6. The Flesch-Kincaid is then normalized to range <0;100>
7. The words are sorted by the score value from the easiest to the most difficult ones





## Evaluation

1. To evaluate the algorithm we generated 20 words using Random Word Generator online tool
2. The words were scored using the algorithm and given to a group of 90 people for assessing the difficulty
3. The human results were merged as follows:
  1. For each person participated, get a list of their result
  2. For every list, attach scores with every word, so that the first word (the easiest, according to the participant) gets 1 point and the last (the most difficult) gets 20 points
  3. For every word then calculate average score from all the lists
  4. Sort words by calculated average score ascending
4. For the test purpose we then divided the words in 4 difficulty categories based on the score



## Evaluation (2)

1. The proposed algorithm assigned 11 words, 8 words were placed in the neighboring category
2. 1 word (wobble), considered by humans as one of the most difficult was placed by the algorithm in the second category, meaning it should be rather easy
3. The question is are those results satisfactory? - For our purposes yes



## Any problems?

1. Time to score words – with large database it takes a lot of time to calculate scores.
2. Especially the words starting with “w” took few hours per word to assess
3. Scoring all 54 475 mappings against 4 838 000 pages takes few months on an average server with a dedicated VM for the database
4. The database requires about 100GB for the task
5. The score will never be perfect



## Conclusions

1. We proposed a tool for heuristically assignment of difficulty to given words
2. We believe it will work well for quiz like games as difficulty is always a subject of personal opinion
3. It should lower the chance of getting very difficult words at the beginning of the game
4. The order of the words can be tweaked by the players while playing the game – in this case many wrong answers can mean that either the mapping is wrong or the word is displayed to early in the game



GDAŃSK UNIVERSITY  
OF TECHNOLOGY

Thank you!  
Any questions?