Wiki Racer (Jaroslaw Hirniak and Michael Lotkowski). Wiki Race is a game where you are given two articles on the Wikipedia and you have to get from the first to the other in as few steps as possible [http://cs.mcgill.ca/~rwest/wikispeedia/]. It seems on average to be possible to win a game in 5-6 steps, e.g. you can traverse "Black Hole > Roger Penrose > King's College London > London > Iron Maiden" or "Monarch Butterfly > International Space Station > United Kingdom > Edinburgh > University of Edinburgh > Philip Wadler > Monad".

Apparently, even such seemingly unrelated terms as 'monarch butterfly' and 'monad' or 'black hole' and 'Iron Maiden' only have few separating links. We want to verify if Milgram's famous small-world observation, where between any two americans there were only 6 handshakes apart, applies to knowledge on the web and the Wikipedia particular [https://en.wikipedia.org/wiki/Small-world_experiment].

Whether the notion of the small-world in the Wikipedia is observed or not, there are a few ideas listed below we would like to investigate further. However, due to time constraints we will only be able to do maximally 2-3 from the listed below.

Optimal Wiki Racer. Could a general strategy for traversing Wikipedia be created that could be implemented in a computer? For example, if the strongest connectors are places then we could always take place, then concentrate on field specific articles. Is it simple enough to implement it on a computer without giving it any unfair advantage (e.g. the knowledge of links further ahead)? *This task could result with an implementation of bot playing against human or a description of strategy for a human supported by experimental results.*

Strongest associators. Do categories of articles exist functioning as strongly connected hubs? For example, locations seem a good pathway to navigate to a seemingly unrelated topic. This could provide further observations on what connects people or article subjects, e.g. what do Black Holes and Iron Maiden have in common? They both have people from London associated with them. *This could result with a program stating what two things have in common, or report description of findings on ran experiments.*

Superhub. Related to the above, but requiring enough additional time to be classified as a separate class. Do there exist superhub(s), from which we can get to any other article on the Wikipedia? Do there exist articles that are completely separated from others? If so, why does it happen?

Domains of knowledge. Do there exist strongly-connected clusters identifying areas and fields of knowledge/study? If yes, what are they, does separation correspond to real life differences between fields? Can we observe the influence some fields have on others (e.g. biology and informatics on bioinformatics)? Are there any salient points we can find about domains, and how they evolve and are shaped.

Structure of the Wikipedia. In lecture 7, the 'bow tie' structure of the web was presented. What can we say about the structure of the Wikipedia or the web knowledge systems in general? Do they form any particular structures? If yes, what influences such and not the

other topologies? Does certain domains have certain “shapes” of nodes (in/out edges, to what they connect, etc.)?

Similarity search. All articles (for this experiment about 1M subset of the articles) can be hashed to a binary number (e.g. to int) and stored in a map. Now, we can request similar documents by computing hash for the original document, and find related documents by performing a search in a map by flipping one bit for each search (Hamming distance = 1). It will require using deep autoencoder as a hash-function which has desired properties of producing similar hashes for similar documents.

This has many interesting aspects as we provide a unique description of an article using hash which should have correspondence to an article position in the graph, but only takes one integer to describe, hence giving very efficient encoding (given if it works at a desired accuracy). Using deep autoencoder to find approximate match for similar documents by recovering the matching documents by flipping bits in the hash function. Also, it enables us to return similar articles in constant time (O(1) to compute hash and O(1) to find 32 similar articles, including misses). *Deep autoencoders are artificial neural networks used for learning efficient codings. This is likely very involved and time consuming project, but if enough code is found online and reused then it should be doable. [Geoffrey Hinton’s description https://www.youtube.com/v/AyzOUbkUf3M&start=1898 and code for MNIST (handwritten digit recognition) autoencoder http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html]*

How does the knowledge on the WWW is different from one in the brain? Neuroscience and computation is still very young field, but we already have some knowledge about how acquired information is stored in our brains. It would be interesting to compare that with results found by analysing the Wikipedia graph (mainly the associativity).

Fundamental knowledge identification. Can we identify fundamental knowledge, e.g. mathematical axioms from which proves, lemmas are derived from which whole new fields emerge? What counts as fundamental knowledge in certain fields e.g. mathematics, physics, biology, pop culture, history, etc. ? What are the influences from different fields, e.g. Boltzmann distribution and Boltzmann machines in Neural Networks?

Influencers and the evolution of idea.
Can people or ideas that influenced most given domains be identified from the graph structure? Most likely, they would be in origins and centers. If we can identify influencers, can we identify the beneficiaries as well by flipping the description, e.g. the nodes with most incoming edges, or most edges from influencers, etc.?

How does one idea influence other ideas? Can we study the number of in/out links and the graph properties and identify how one idea influence studies of another idea, and lead to development of a field or how some fields come into forming others (e.g. biology and informatics to form bioinformatics, etc.).

Improving the Wikipedia by links backpropagation.

It is not uncommon that some pages have some phrases as links while others do not, as they all require human annotation. In the result, we have situations where there is strong direction in a flow (i.e. often we can go one way, but not back). Could we improve the Wikipedia by back-propagating the links, i.e. if the ‘International Space Station’ links to the ‘United Kingdom’, should the ‘United Kingdom’ also link back to the ‘International Space Station? Also, could backpropagation generate indexes, would all articles linked to the ‘United Kingdom’ by category even be useful?

Cut-search.
Usually, Wikipedia queries are in the form of an article name or a keyword, relying on the existence of an index, and can only lead us to an indexed article. However, if a user would enter ‘Boltzmann distribution’ and ‘autoencoder’; could we return the document most relevant to both by finding that point in the middle of the path between the two? Could finding such a point be feasible? Would it be meaningful to the search or would the category be too general and rendered irrelevant (e.g. a general term like Austria)?

Book generation.
Can books be simply generated? Writing a scientific book requires expertise in a certain domain, but let's imagine we want to write a book about spectral graphs. To begin, we would first look which aspects are fundamental for spectral graphs (such as linear algebra, Laplacian, etc.) and include it in the first chapters. Next, we would identify the core knowledge in the area by finding the strongest connected articles related to spectral graphs, which we would include in the middle of the book. Finally, we would look for articles that have flow from the core area, that is very likely form expert knowledge in spectral graphs, and include it at the end of the book. We can clearly trace the flow of influence here, linear algebra, mathematical operators, graphs, etc. to make spectral graphs analysis possible, thus establishing the basis of spectral graphs. The core knowledge should come from the strongly-connected cluster about spectral graphs. Finally, expert knowledge is cutting-edge, so it should not have too many outgoing edges, but some incoming from spectral graphs topic (the ratio between other incoming edges can be measured to identify how much it is related to spectral graphs).

Having generated such an index, the rest is a matter of choosing appropriate sections of a text from the articles. We can even put references from articles at the end of the book, let’s be scientific!

Would such an idea be possible? Could we construct an index for a hypothetical book? How would the information be related? If books are too complex to be generated can we at least generate guidelines to study given matter? You say to give me learning material on “spectral graphs” and the index is generated which should guide you through the process with links to the relevant articles.

Reliability of the Wikipedia articles.
It often arises as a subject of discussion on how much one can rely on the Wikipedia articles, if anyone can edit them. Notably, there have been cases of article modifications by individuals or institutions seeking personal or financial gains, e.g. companies wishing to sell

supplements by adding information to the Wikipedia on certain conditions and well-being related sections [https://en.wikipedia.org/wiki/Wikipedia:Conflict_of_interest]. Could such conflict of interest be detected by classifying individuals pattern of edits (e.g. many editions on a specific page type) or individual connections to a product added to the pages.

Datasets:

- Wikiseedia (navigation paths on the Wikipedia hyperlink network): https://snap.stanford.edu/data/wikispeedia.html for experiment if the Wikipedia is small-world network/
- Wiki-RfA: https://snap.stanford.edu/data/wiki-RfA.html and wiki-meta: https://snap.stanford.edu/data/wiki-meta.html for experiments regarding reliability of the Wikipedia articles/.
- The whole wikipedia: https://en.wikipedia.org/wiki/Wikipedia:Database_download preprocessed using Apache Spark.