On the right track! Analysing and Predicting Navigation Success in Wikipedia

Tobias Koopmann
DMIR Group
University of Würzburg
Germany
koopmann@informatik.uni-wuerzburg.de

Alexander Dallmann
DMIR Group
University of Würzburg
Germany
dallmann@informatik.uni-wuerzburg.de

Lena Hettinger
DMIR Group
University of Würzburg
Germany
hettinger@informatik.uni-wuerzburg.de

Thomas Niebler
DMIR Group
University of Würzburg
Germany
niebler@informatik.uni-wuerzburg.de

Andreas Hotho
DMIR Group
University of Würzburg
Germany
hotho@informatik.uni-wuerzburg.de

ABSTRACT

Understanding and modeling user navigation behaviour in the web is of interest for different applications. For example, e-commerce portals can be adjusted to strengthen customer engagement or information sites can be optimized to improve the availability of relevant content to the user. In web navigation, the users goal and whether she reached it, is typically unknown. This makes navigation games particularly interesting to researchers, since they capture human navigation towards a known goal and allow building labelled datasets suitable for supervised machine learning models.

In this work, we show that a recurrent neural network model can predict game success from a partial click trail without knowledge of the users navigation goal. We evaluate our approach on data from WikiSpeedia and WikiGame, two well known navigation games and achieve an AUC of 86% and 90%, respectively. Furthermore, we show that our model outperforms a baseline that leverages the navigation goal on the WikiSpeedia dataset.

A detailed analysis of both datasets with regards to structural and content related properties reveals significant differences in navigation behaviour, which confirms the applicability of our approach to different settings.

ACM Reference Format:
Tobias Koopmann, Alexander Dallmann, Lena Hettinger, Thomas Niebler, and Andreas Hotho. 2019. On the right track! Analysing and Predicting Navigation Success in Wikipedia. In *30th ACM Conference on Hypertext and Social Media (HT '19), September 17–20, 2019, Hof, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3342220.3343650

HT '19, September 17–20, 2019, Hof, Germany

ACM ISBN 978-1-4503-6885-8/19/09…$15.00
https://doi.org/10.1145/3342220.3343650

1 INTRODUCTION

Understanding human navigation in the Web is an important part of user behaviour analysis, and can for example be used to improve website navigation [1]. One aspect of web navigation is finding the shortest path towards a specific page, be it an item on Amazon or a specific article on Wikipedia [2]. While search engines can often retrieve the desired information, they may not always do so [3]. Hence, the user needs to locate the desired information in the depth of the web by navigating along hyperlinked pages [4, 5, 6].

However, due to the complex hyperlink structure of the web, users may still not find the desired information even after navigating along several links. Since in a real-world setting we do not know a user's navigation target, it is almost impossible to distinguish whether a user ceased the navigation task because she was successful in reaching her desired information or because she gave up searching [7]. To prevent the user from getting frustrated and leave a web page unsatisfied, it is helpful to know as early as possible whether a user is on the right track to reach his intended target.

A scenario, where the target is known are navigational games, for example on Wikipedia. In those games, users start on a randomly selected start page. Their mission is to find a path using Wikipedia's link structure to reach a predefined, randomly selected target page (see left part of Figure 1). The collected click trail data is especially interesting for research purposes, since it allows investigation in human navigation behaviour towards a known goal. Although, in this setting the goal is artificially chosen, the user still follows his intuition to find a shortest path by navigating along the hyperlink structure, similar to "normal" human navigation in the web. Thus we consider this game setting a useful substitute for other types of navigation. We will study how models can exploit the latent knowledge about the information graph expressed in click-trails to predict task completion using data from navigational games.

In previous work, Scaria et al. [8] analysed and predicted task abandonment in the context of the navigation game WikiSpeedia. However, their approach utilizes hand-engineered features and is

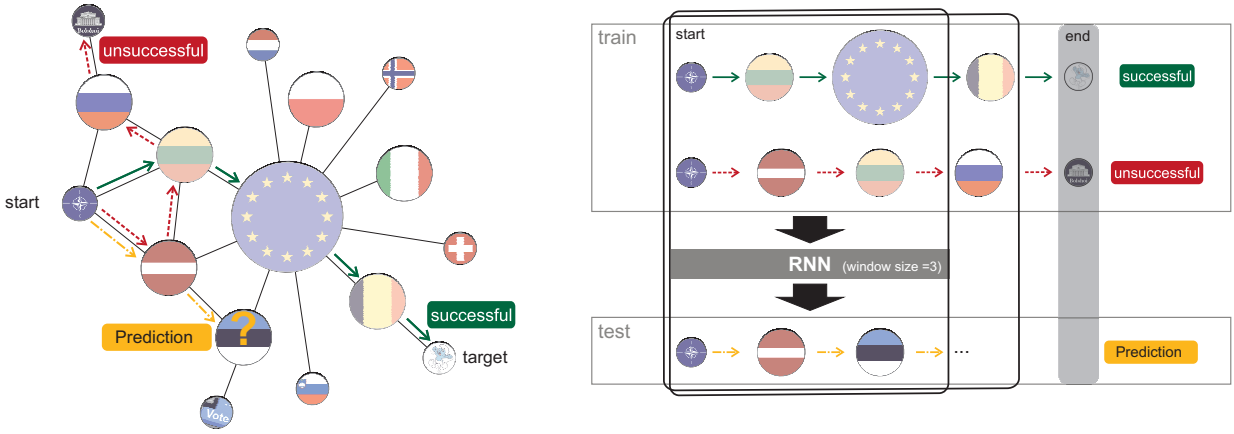


Figure 1: Illustration of the success prediction problem on Wikipedia. Section 1 shows an exemplary information graph. e.g. from Wikipedia with three played navigation games. In those games, the user has to find the shortest possible path from a start to a target page. One session is finished successfully (green, solid), one unsuccessfully (red, dotted) and one is still in progress (yellow, dashed). We analyse these click trails from two different navigation games on different properties and build a model (Section 1) to predict the outcome of the session based on a partial trail (in this case 3 nodes). The main goal is to predict "live", e.g. for the yellow path, if a game will finish successfully or not.

heavily dependent on prior knowledge of the target page, which is unknown in most settings.

We propose a RNN based model that predicts successful navigation in the setting of navigational games on Wikipedia, by only considering partial click-trails and explicitly excluding knowledge of the target page. Figure 1) shows an illustration of this approach. Building upon previous insights by the work from Scaria et al., our model aims for a next step towards a live prediction, which would be applicable to a range of applications. To ensure the validity of our approach, we evaluate the model on two different navigation datasets, originating from the two well-known games WikiSpeedia[1] and WikiGame[2]. Our results show that although no target node knowledge is included, our model can outperform all given baselines and is able to predict success on both datasets.

An analysis of the datasets in terms of structural and semantic features with regard to successive clicks shows differences in user behaviour between both games, which is most likely caused by different game settings. This indicates that our approach is not limited to a single type of human navigation behaviour. Additionally, we study the impact of different representations for the model input and consider representations based on purely structural properties and ones based on semantic properties, e.g. document embeddings.

Structure. This work is structured as follows. Section 2 covers related work. In Section 3 we provide a description of the used datasets from WikiSpeedia and WikiGame. Next we analyse the user behaviour in Section 4 with respect to structural and semantic features. We describe our experimental setup in Section 5. In Section 6 we present and discuss our results. Finally, we summarise our contributions in Section 7.

[1]https://www.wikispeedia.net
[2]https://thewikigame.com

2 RELATED WORK

Related work can be divided in different sections: user behaviour modelling, click trail analysis, query abandonment, and graph node embedding algorithms.

User Behaviour modelling in the web analysis clicks and understands the reasons of human transitions between pages. Previous work by Trattner et al. [9] analysed and modeled human navigation on Wikipedia using decentralized search. Their findings show that the best decentralized search approach is using background information based on the link graph of the underlying dataset.

Helic et al. [10] extended this work and compared navigation in information networks to navigation in social networks., Based on this insight, they created different generative models based on decentralized search for both networks types.

Click trail analysis is a specific part of user behaviour modelling and examines clicks made by users on web pages and finds reasons for clicks or click patterns in order to explain user behaviour. As we focus on game navigation in Wikipedia, we only consider click trail analysis based on WikiSpeedia [8, 11, 12, 13, 14] and WikiGame [14, 15, 16, 17].

In particular, West and Leskovec [12] examined how users find their way from a given start node to a target node in an information graph, such as Wikipedia. Their investigation shows, that users are likely to get an overview first. This is justified by their need to orientate first, which is done by clicking hyperlinks to nodes higher in the graph hierarchy. After obtaining an overview, they navigate to the given target node. This is usually done by clicking links to nodes, whose content is more similar to the target node. This process is called zooming-in. If the user is not able to zoom in properly, he clicks nodes with a similar or higher shortest possible distance (click-wise) to the target node. Because the distance to the target node remains the same, this is called orbiting. We are able to find this behaviour in our analysis as well. Furthermore

West and Leskovec [13] examined the difference between automatic and human navigation. For that they used the WikiSpeedia dataset as baseline and created different models with individual strategies in order to outperform human search. One question was, if humans with background knowledge are able to find the target faster than machine agents, whose only knowledge is their direct neighbourhood. Surprisingly their investigation shows, that the agents were better in finding short paths and hence they argue that global knowledge about the graph is not necessary. Nogueira and Cho [18] are also using machine agents to navigate on Wikipedia-based networks. Their investigation shows that their agents are able to outperform search engines, which shows that click-based navigation is more likely to finish in satisfactory results than search engines.

Query abandonment studies, why users click or ignore the returned links of a search query. The information need behind a search query is comparable to an information need when browsing and navigating in the web. The abandonment of such processes could be caused by similar reasons. Li, Huffman, and Tokuda [7] introduced the idea of *good* and *bad search abandonment*. They defined a general Web search abandonment as not clicking any link offered. However, although no link was clicked, the user might have found the sought information by reading the snippets of links or information boxes provided by modern search engines, which happens in 50% of the cases. If so, it is called good abandonment and thus equals a successful navigation in our setting. Further work by Diriye et al. continued this idea and examined reasons for good or bad abandonment [19]. Based on these insights, they predicted abandonment with features extracted from the inserted query, the returned results from the search engine and mouse movements from the user after seeing the results. Another approach using bypass rates was proposed by Sarma, Gollapudi, and Ieong [20]. Bypass rates denote for each search result, how likely it is that a lower-listed result is clicked instead. This means the higher-ranked result is bypassed. The idea is to reduce bypass rates using click logs and similarity of the query results. Finally, Scaria et al. [8] analysed the WikiSpeedia dataset and presented reasons, why users are likely to abandon a task or finish it successfully. They extracted features regarding the graph and content structure of the underlying dataset, as well as human interaction in terms of back clicks from human click trails.

Finally Han et al. analysed task abandonment in a crowdsourcing setting [21]. They set up different case studies and analyse factors why users are abandoning their task in this setting. Their main finding is, that this abandonment mostly happens in the early stages of a task, because the user understands the intrinsic hardness of the task. Abandonment in later stages of a crowdsourcing task is rather rare.

Embeddings are used for representing large scale input in a lower dimensional vector to be used as input for different models. As the graph of Wikipedia is very large, we will represent the nodes in this graph by means of embeddings respecting different properties of the node. word2vec [22] can be seen as the origin of embeddings, which embeds single words in the context they appear. To embed whole documents, Dai, Olah, and Le [23] developed the paragraph vectors approach, which is based on word2vec.

Furthermore Wikipedia contains an underlying link structure. Based on this structure graph embeddings [24, 25, 26] can be created. They embed embed a node into an n-dimensional vector space by looking at the context a node appears in. The context in graph embeddings can be created by performing random walks on the graph structure.

3 DATASET ANALYSIS

As previously mentioned, we make use of two datasets based on Wikipedia in order to analyse user navigation. These datasets are WikiSpeedia[3] and WikiGame[4]. WikiSpeedia (d_{WS}) has already been analysed several times [8, 12, 13, 14]. We utilise a second dataset based on WikiGame (d_{WG}). It contains more click trails, is based on a larger link graph and has a different setting by including a time limitation for played games.

In this section we will analyse both datasets and explain our preprocessing steps.

3.1 Setting

In general, each dataset consists of a graph G and trails T. G consists of a set of nodes N, corresponding links L between nodes and contents C associated with each node. In our case, the content of a node is represented by the text of each page. The set of click trails T contains the logs of played game sessions and includes all transitions generated by players.

At the start of each session, a user is asked to navigate from a randomly selected *start* to a *target* article/page by exclusively following hyperlinks. For both, WikiSpeedia and WikiGame, the users are allowed to backtrack. The goal is to reach the target page in as few clicks and/or little time as possible.

In case of *successful* navigation, the end node of the click trail is equal to the *target page* of the session. A navigation trail is called *unsuccessful*, if it is abandoned before the target page is reached. In the case of d_{WS}, the reason of abandonment is logged. There exist two possible reasons, either the user was not active for a long period (*expired*) or he chose to end the session and start a new one (*restart*). In d_{WG} exists a maximum time limit of two minutes in which a user has to finish the session. Hence, a session is either finished successfully by a user within the given time span (successful) or not (unsuccessful).

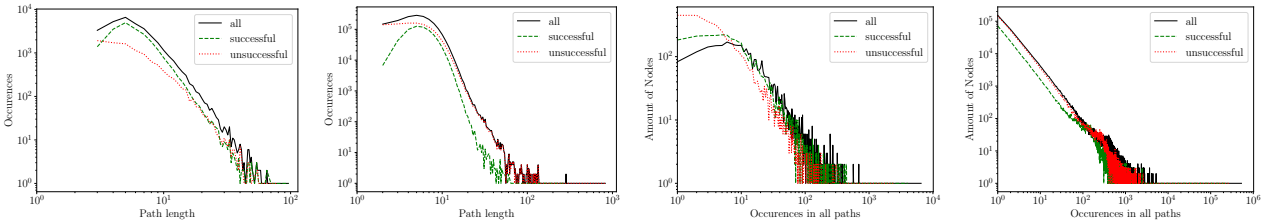
In the following section, we compare both datasets by means of their components, the graph G, the content C and the click trails T. Aside from this detailed description, we summarise information about the datasets in Table 1.

3.2 Information Graph Preprocessing

WikiSpeedia is built upon the Wikipedia for Schools corpus, a small selection of articles from Wikipedia to be distributed in schools. It contains ~4600 nodes N_{WS} and ~120 000 links L_{WS} connecting them. The average shortest path distance between two nodes is 3.18. The content of Wikipedia articles ranges from 120 to 22 390 words, with a mean of ~4220 words per page.

The information graph of WikiGame is structurally similar to WikiSpeedia, but much larger with respect to nodes and links. It

[3] https://www.cs.mcgill.ca/~rwest/wikispeedia/
[4] https://www.thewikigame.com/



(a) Click trail lengths d_{WS} (b) Click trail lengths d_{WG} (c) Node usage d_{WS} (d) Node usage d_{WG}

Figure 2: Overview of different distributions for the click trails of WikiSpeedia (d_{WS}) and WikiGame (d_{WG}). The two left hand graphs depict the distribution of click trail lengths, whereas the right-hand figures show the occurrence of nodes in all click trails, e.g. d_{WS} has more than 100 nodes occurring in only a single game.

Table 1: Comparison of the WikiSpeedia and WikiGame Dataset. This table shows the properties after preprocessing the datasets (Section 3.2). (# = amount)

	WikiSpeedia	WikiGame
# pages	4603	151 948
# links	119 882	12 305 081
Average links per page	26	71
Average usage of a node	49	30
# successful click trails	21 997	613 621
# unfinished click trails	10 410	986 314
# all click trails	32 407	1 599 935
Ratio of successful trails	0.68	0.38
Average trail length	6.91	5.98

is constructed from all Wikipedia articles occurring at least once in a game. This results in a total number of ~360 000 nodes and ~25 000 000 links. Due to the information sparsity of nodes on the border of the graph(can be seen in the power law tail distribution in Figure 2d), we employ an induced subgraph by removing all nodes occurring only once or twice in total. This results in 151 948 nodes and 12 305 081 links. By deleting these nodes, we also remove the click trails containing these nodes. This reduces the amount of overall click trails from ~1.8 million to ~1.6 million click trails.

3.3 Click Trail Preprocessing

To get a better understanding of user navigation behaviour in Wikipedia games, we further investigate the click trails of both datasets. A visualisation of the statistics on the number and frequency of nodes in click trails are shown in Figure 2.

Originally, d_{WS} contains more than 75 000 human click trails, gathered over a period of about three years through WikiSpeedia. Unfortunately, successful and unsuccessful click trails were collected during different periods of time. To align the time span, only trails in the overlapping interval are considered, which is between 02/27/2011 and 01/15/2014. In addition, all expired unsuccessful trails were omitted from the dataset, as the user might have been distracted or disinterested. In contrast, a restart indicates, that the user was unable to find an optimal way to the target and hence he started a new session. Additionally, we consider only trails containing at least 3 nodes. We adapted these preprocessing steps from Scaria et al. [8] to ensure comparability between our approaches.

Figure 2a shows the distribution of trail lengths in the preprocessed WikiSpeedia (d_{WS}) using this preprocessed dataset. The average navigation trail is 6.91 nodes long. Since we only consider click trails with at least 3 clicks, the left-most part of the plot is blank. Most of T_{WS} have a length of 5 to 6 clicks, whereas the total span reaches from 2 to 96 clicks. The most successful trails occur in the range of 5 to 8 clicks, whereas we can observe more unsuccessful trails at a length of 3. We assume that users tend to either abandon early (at 3 clicks), or keep playing until 8 or more clicked nodes.

Based on the length distribution of the click trails, we decided to use an additional preprocessing step. We assume very long games to be noise, due to the fact that it is very unlikely to seriously play the game. Hence we remove very long click trails (31+ nodes) from the dataset without loosing many samples (less than 1% of all click trails). After the preprocessing the number of trails in T_{WS} is reduced to 32 407. From these, 21 997 are successfully finished and 10 410 are unsuccessful.

On average, a single node occurs in 55 click trails (c.f. Figure 2c). The most used nodes are 'United States' (6523), 'Europe' (2833), 'United Kingdom' (2723), 'Earth' (2429) and 'England' (2264). 83 nodes from the Wikipedia for Schools dataset are only mentioned once and 518 nodes are not used at all.

T_{WS} contains only a small amount of very long click trails, e.g. 95% of all click trails have less than 21 clicks.

In contrast, the WikiGame dataset d_{WG} contains more click trails with a shorter average length of ~5.98. This is likely caused by the time limit in WikiGame that forces users to find the target node within two minutes or restart the session. After applying the previously explained graph preprocessing (see Section 3.2), the click trails contain 613 621 successful and 986 314 unsuccessful trails captured between 02/17/2009 and 09/12/2011. Comparable to the preprocessing for the WikiSpeedia dataset, we remove trails with more than 31 clicks, which equals 1% of the longest click trails.

The distribution of nodes over trails for d_{WG} are displayed in Figures 2b and 2d. Similar to WikiSpeedia the most used nodes

are 'United States'(527 662), 'United Kindom' (81 260), 'England' (52 012) and 'Europe' (50 990).

In contrast to d_{WS}, which is biased towards successful outcomes, d_{WG} contains a surplus of unsuccessful click trails. This is attributable to the time limit, which yields an unsuccessful trail, if it is exceeded. In total, d_{WG} has 55 times as many trails, 78 times as many nodes and 214 times as many links as d_{WS}.

4 USER BEHAVIOUR ANALYSIS

Our goal is to predict game success from partial information of click trails in a web navigation setting. In this section we study user behaviour relating to successive nodes in click trails. By focusing on these properties, instead of e.g. node relation to the target node [8], we obtain insights into user behaviour with only local knowledge, which can be observed by the model in an online setting.

Faced with the decision which link to choose, a user likely decides based on different aspects of the current node (and the potential neighbours) in order to choose a node closer to the target. With respect to this observation, we study nodes from two different perspectives:

- We consider a structural perspective, in which the node is measured with respect to its absolute position in the graph G.
- We study the node from a semantic perspective by analysing its similarity in terms of node content C (TF-IDF) to successive nodes.

We normalize sessions according to their length, which allows us to make general statements and detect different stages of sessions. Figure 3 shows textual and structural properties of nodes. The positions are displayed relative to the total path length.

4.1 Semantic Analysis

At first, we analyse how similar consecutive nodes in click trails are with respect to their content. As representation we will use Term Frequency - Inverted Document Frequency (TF-IDF) vectors. Figures 3a and 3b show the cosine similarity of these representations between successive nodes throughout the click trail.

For d_{WS}, Scaria et al. observe an increasing similarity between the current and the target node as a users game progresses. Our analysis shows the same phenomena, but instead of observing the similarity between current node and the target, we observe similarity of successive nodes. The differences of similarity are different for successful and unsuccessful click trails. Complementary, our results (Figure 3a) indicate the same behaviour regarding the similarity of consecutive nodes throughout the click trail. For unsuccessful games the similarity tends to stagnate towards the end, implying that the user is unable to find a way toward the target by means of a semantically more similar node.

For d_{WG}, semantic similarity behaves rather differently. Firstly, it is overall lower in comparison to d_{WS}, which is most likely the result of the larger amount of distinct words in the dataset. Unsuccessful click trails have overall a higher similarity of successive nodes and remain on the same level.

In contrast, successful games have an overall low similarity and even decrease towards the end. This behaviour differs from d_{WS} and is rather unexpected. We assume that due to the time limit, users are forced to use different tactics and rely less on semantic similarity for finding a path to the target node.

4.2 Structural Analysis

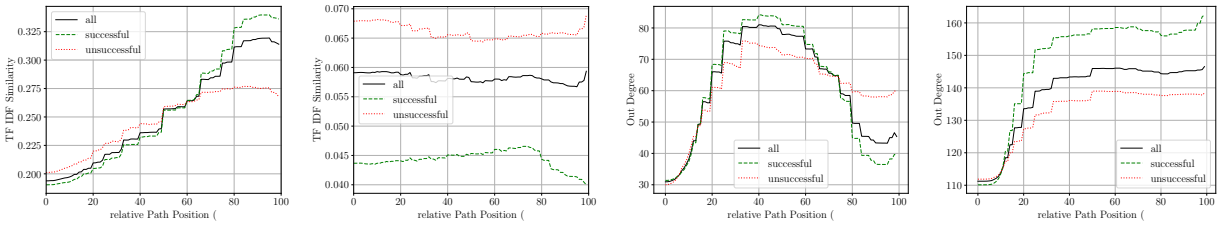
Apart from semantic content, nodes can be described by their structural characteristics. Thus, we analyse human navigation based on different graph properties like out degree, page rank or degree ratios. As we observe equal trends for all properties, we only describe out degree as an example for structural navigation.

Figure 3c shows the structural analysis of WikiSpeedia. A typical session starts at any node in the graph with an average out degree of ~30. This represents an arbitrary selected node in the graph, based on an average out degree of 29 for all nodes. First the user utilises nodes with higher out degree to move away from the source and find a hub node, from which he can zoom in to the target node. According to the plot, the zoom-out process takes up to 30% of the click trail until he finds hub nodes with ~75 to ~85 out links, which is 2.9 times higher than the start page. Afterwards the user remains on nodes with similar high out degree until he starts to zoom in onto the target.

On average, successful click trails end on a node with an out degree of ~40, whereas the average target node of all sessions has an out degree of ~30. This indicates, that sessions with better connected targets nodes are more likely to be finished successfully. In contrast, the out degree of target nodes from unsuccessful sessions is at ~25, which is below the overall average out degree and indicates a more difficult session. This is more likely to result in user not able to find the target, because it is "hidden" in the graph structure. The average out degree of the last node in unsuccessful sessions is at ~60, which shows an attempt to zoom in, but the user is not able to finish the zooming process and find the target.

The behaviour for WikiGame is similar to the first part of the WikiSpeedia analysis, Figure 3d shows how the out-degree changes throughout the session. Due to the larger size of the WikiGame graph, out degrees are overall higher (~71.4) and on average users start at nodes with an out degree of ~110. This means that in contrast to WikiSpeedia sessions tend to start at hierarchically higher than average nodes. Comparable to WikiSpeedia, users start to zoom out towards a hub node. For successful sessions this behaviour is more prominent and users reach a node with an out-degree of ~155 on average. Unsuccessful sessions only reach a hub node with a mean out-degree of ~140. This is an increase of factor 1.4 for successful and 1.3 for unsuccessful nodes in comparison to the out degree of the start node, which is significantly lower compared to WikiSpeedia (2.9). We assume that users are stimulated to make faster decisions due to the time limit of WikiGame. This results in users not willing to spend much time on zooming out, which applies to unsuccessful click trails especially. The average out degree of the target node in successful sessions is ~160. This shows that successful sessions typically end on nodes with higher out degree, hence they are easier to solve. Unsuccessful games end on nodes with an average out degree of ~140, and are thus inherently harder to complete.

Overall, WikiSpeedia and WikiGame exhibit quiet different properties especially concerning the zoom-in phase identified for WikiSpeedia in [12]. The analysis of WikiGame does not show a distinct



(a) Textual analysis of T_{WS} (b) Textual analysis of T_{WG} (c) Structural analysis of T_{WS} (d) Structural analysis T_{WG}

Figure 3: Analysis of click trails for WikiSpeedia T_{WS} and WikiGame T_{WG}. Values are obtained by scaling all trails to the same length [0, 100] and averaging at each specific position over all trails. The black line shows all sessions, the green, dashed line show all successful ones and the red, dotted line shows the unsuccessful ones.

zoom-in phase, but rather a stagnation on high out-degree levels. Although the analysis shows otherwise, assuming a random selection process, the target node should have a much lower out-degree on average, as is the case for WikiSpeedia. We can only speculate, but a possible explanation is that less difficult targets are chosen, to make the game more approachable under the two minute time rule.

5 EXPERIMENTAL SETUP

In this section, we describe our experimental setting in detail. We outline the different evaluated node representations (structural and semantic), the construction of evaluation datasets, the utilised Recurrent Neural Network (RNN) classifier as well as evaluation metrics and baselines.

5.1 Data Sets for Evaluation

We use different evaluation strategies for the two datasets after applying the preprocessing described in Sections 3.2 and 3.3. For d_{WG} we randomly split the dataset with a ratio of 80:20 (train:test) and ensure that the label distribution in train and test set is similar. Due to the smaller amount of data, we use 10-fold cross-validation for the d_{WS} dataset.

In this work we want to approach an online setting, where the user has already made a number of clicks. Hence we extract input sequences of different lengths for each click trail by using only the first clicks. By choosing the sequence size between $[2, 30]$ we generate distinct datasets, with which we study how the prediction quality of our model varies with increasing click-trail length. If a click-trail exceeds this size, all further nodes are removed. To ensure no information about the target node is used, we additionally remove the last node for all successful click trails. To remain consistent in the preprocessing, the last node in unsuccessful click trails is removed as well.

Our analysis in Section 4 shows that click-trail properties tend to vary more towards the end of a game. To study these dynamics, we also experiment with sequences selected from the rear of click trails with a sequence size in between $[2, 5]$.

For example, from the following click trail from start *Hanukka* to target node *Oceania*

Hanukkah → Jerusalem → Middle East →
Turkey → Europe → Oceania

we construct two samples using a sequence size of 3:

start phase: Hanukkah → Jerusalem → Middle East
end phase: Middle East → Turkey → Europe

To avoid overfitting, we randomly undersample the majority class during training, but keep the original ratios for the test data.

5.2 Input Representations

Based on our analysis we consider two different representations for nodes.

- a semantic representation, where each node is represented by a vector computed from its content
- a structural representation, where only structural information from the underlying graph is used

Semantic. According to Figure 3 successful and unsuccessful users navigate differently in terms of TF-IDF similarity. Since TF-IDF vectors are commonly not suitable as an input to RNN models because of their size and sparsity, we use paragraph vectors[5] [23] to compute dense document embeddings with a size of 64 from the complete article text for every node.

Structural. Apart from textual features we will use structural representations of nodes. Graph properties seem to contain valuable information for the distinction of successful and unsuccessful games (cf. Figure 3). To this end, we construct embeddings with respect to the link structure of WikiSpeedia and WikiGame respectively. There exist several graph embedding approaches, e.g. LINE by Tang et al. [24] or Node2Vec by Grover and Leskovec [26]. Due to its popularity, simple application and restriction to the graph structure, we opt for DeepWalk by Perozzi, Al-Rfou', and Skiena [25].

We tested different settings to create the Deepwalk embeddings and found that increasing most of the values results in better classification performance. Since the default settings are meant to create local embeddings by only considering nearby nodes, we found that large scale embeddings work better in our scenario, because they contain more information about the absolute position in the link graph. We report results for the following settings (defaults are mentioned in brackets):

- Number of walks: 100 (10)
- Representation size: 64 (64)

[5] https://radimrehurek.com/gensim/models/doc2vec.html

- Walk length: 400 (40)
- Window size: 25 (5)

5.3 Description of the Classifier

In order to properly capture the sequential aspect of click-trails, we use a LSTM as a classifier. Each node of the input sequence is embedded before being fed into a Long Short-Term Memory (LSTM) [27] with a layer size of 128. The final state is then used as input to a feed-forward layer of size 2 with a softmax activation function. The node embedding layer is pre-initialised with either *semantic* or *structural* embeddings. Our experiments showed better results when we allowed the model to adapt the pre-initialized embeddings during final training. The model is trained for 10 epochs using the Adam optimizer [28] and a cross-entropy loss function. These parameters were found to perform best during a preliminary study.

5.4 Evaluation Metrics and Baselines

Our models are evaluated using two metrics: accuracy, which is commonly used as metric for classification, and Area Under the Curve (AUC) score [29], which is specialised on prediction without an optimal threshold.

We compare our model on WikiSpeedia against two baselines, one being the model used by Scaria et al. and the other one being the majority vote. To the best of our knowledge, there exists no prior work on success prediction of WikiGame sessions. Hence we only use the majority vote as baseline for this dataset.

Majority Vote. This baseline (MV) predicts the major target label of the whole dataset, which is successful in case of d_{WS} and unsuccessful for d_{WG}. For WikiSpeedia the majority vote is 0.66 and for WikiGame 0.61. The MV changes only slightly when removing short click trails (up to a length of 5), hence we report only the accuracy of the overall MV.

LR+target. Scaria et al. [8] predict, whether a click trail is abandoned using logistic regression and up to 30 different hand-crafted features. All these features are generated using the relation to the target node. Hence, this baseline differs from our approach with respect to model (regression versus RNN), the features (hand-crafted versus embedding) and the knowledge about the target. As we use the same preprocessing steps as they do in their work, we report their results.

6 RESULTS

After describing our experimental setup in the previous section, we now present our results for predicting successful navigation. We examine performance in different dimensions:

- Datasets: WikiSpeedia and WikiGame
- Input Representations: Semantic and structural
- Input Sequence: different sequence sizes from start or end phase
- Metric: Accuracy and AUC

We compare our results to the baselines explained in Section 5.4. Finally we analyse the certainty of our model by using different sequence positions.

6.1 Success Prediction without Target Knowledge

As stated in Section 5.1, we evaluate each dataset for different sequence sizes from $[2; 30]$ at the start phase, or from $[2; 5]$ at the end phase.

WikiSpeedia. Table 2a depicts the results for different sequence lengths and input representations for WikiSpeedia. We present accuracy as well as AUC and highlight the top scores of each setting.

First, we notice that prediction performance increases with input sequence length, especially for very short sequences of 2 to 3 nodes. On the one hand, this indicates the comparatively low information content for short sequences. On the other hand, the model is still capable of surpassing the majority vote baseline of 0.66 accuracy for these sequences, which shows its capability to extract information in general, even for very short trails.

For longer sequences, the results improve significantly up to a length of 7. Increasing the number of input nodes further results in only small improvements. At a length of 7, the dataset already includes the whole trail for a majority of all trails, hence only few information can be gained using larger input sequences. Overall, the top scores are 0.84 AUC and 0.74 accuracy when using document representations for nodes and 0.86 AUC and 0.77 accuracy for structural representations.

Furthermore, we compare results of start to end phase approaches (upper part compared to lower part of Table 2a). When comparing sequences of equal length, the end phase approach performs better than the start phase approach at predicting successful session outcomes. This behaviour can be explained by our analysis. Successive clicks in Section 4 show an increasing difference between successful and unsuccessful sessions towards the end, which occurs independent of how a game started. The prediction results indicate, that our model can make use of these differences. E.g. for a sequence size of 5 with content embeddings, the end phase has 0.84 AUC and 0.75 accuracy in contrast to 0.79 AUC and 0.70 accuracy in the start phase.

In general using structural input representations results in better performance than semantic features for all used approaches.

Finally, we compare our approach with Scaria et al. [8], whose results are shown in column "LR+target". Although their approach relies on features that contain information about the target node, we are able to outperform their model regardless of the used representation with a sequence length of 6 or higher on WikiSpeedia.

Overall, the proposed model outperforms the baselines in most of the settings, which shows its applicability to success prediction in navigation games. In addition it is able to make a prediction based on partial information about the click trail by using different input sequence lengths and thus represents a step toward an online applicable model.

WikiGame. Table 2b shows the corresponding results for WikiGame. In general, we use WikiGame to validate our approach on a second dataset with different user behaviour.

The overall trends in result scores are similar to WikiSpeedia. First, small input sequences contain the least information, but are still able to outperform the majority vote baseline by a small margin as in WikiSpeedia. With increasing length the score improves

Table 2: Comparison of our results with our baselines. The left table shows results for WikiSpeedia and on the right side results for WikiGame are presented.

(a) Results on WikiSpeedia including the [8] baseline (LR+Target). Majority Vote baseline is 0.66 accuracy.

		LR+target	Semantic		Structural	
	Size	AUC	AUC	Acc	AUC	Acc
start +	2	0.75	0.68	0.63	0.70	0.64
	3	0.77	0.73	0.67	0.76	0.68
	4	**0.80**	0.76	0.70	0.78	0.70
	5	-	0.79	0.70	0.81	0.73
	6	-	0.80	0.70	0.82	0.74
	7	-	0.82	0.73	0.83	0.75
	8	-	0.82	0.73	0.84	0.75
	9	-	0.82	0.73	0.84	0.76
	10	-	0.83	0.74	0.85	0.76
	15	-	0.83	0.73	0.86	0.76
	20	-	**0.84**	**0.74**	0.86	0.77
	30	-	0.83	0.73	**0.86**	**0.77**
end -	5	-	**0.84**	**0.75**	**0.84**	**0.75**
	4	-	0.83	0.74	0.83	0.74
	3	-	0.82	0.72	0.82	0.74
	2	-	0.82	0.73	0.82	0.74

(b) Results on WikiGame with the different input representations. The Majority Vote baseline is 0.61 accuracy.

		Semantic		Structural	
	Size	AUC	Acc	AUC	Acc
start +	2	0.69	0.62	0.74	0.65
	3	0.75	0.67	0.81	0.71
	4	0.78	0.70	0.84	0.75
	5	0.81	0.73	0.86	0.77
	6	0.82	0.74	0.87	0.77
	7	0.84	0.75	0.88	0.79
	8	0.84	0.77	0.89	0.80
	9	0.84	**0.77**	0.89	0.80
	10	0.84	0.76	0.89	0.80
	15	**0.85**	0.77	0.90	0.80
	20	0.85	0.76	**0.90**	0.81
	30	0.84	0.76	0.90	**0.81**
end -	5	0.79	0.73	**0.90**	**0.82**
	4	**0.79**	**0.73**	0.89	0.82
	3	0.78	0.71	0.87	079
	2	0.77	0.70	0.84	0.75

significantly for both representations up to a sequence size of 7. For even longer sequences the results still increase, but less steep.

Overall, the structural representation performs better than the semantic one. Especially for the end phase approach, which performs similarly to using the whole click trail (sequence size of 30) as input with an AUC score of 0.90 and an accuracy of 0.82. This indicates a very distinct advantage of the structural perspective in the end phase for WikiGame.

Results on WikiGame confirm that our model is able to predict game success from partially observed click-trails on datasets with different characteristics. Additionally, embeddings derived from the graph structure seem to perform better for this task compared to embeddings based on node content. Finally, sequences from the end phase yield better performance compared to the start phase, since successful and unsuccessful games exhibit greater differences in the end phase as confirmed by our analysis in Section 4.

Including the target node. Finally, we investigated how knowledge about the target node would influence the performance of our model to find an upper bound. To that end, we included the target node in the input sequence. This results in a top score of 0.91 AUC/0.82 accuracy for WikiSpeedia with a sequence size of 15. For WikiGame, we achieve 0.98 AUC/0.93 accuracy using the same setting.

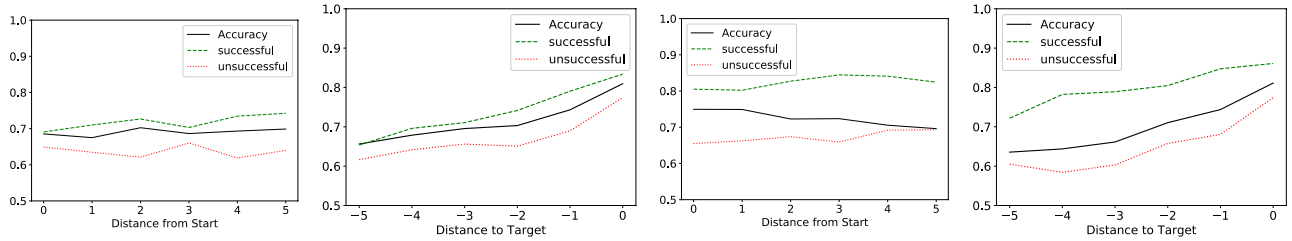
This shows the overall ability of our model to make better predictions on WikiGame than on WikiSpeedia. This observation can be traced in most of our results. Due to the time limit in WikiGame, user are forced to approach the target page fast. Failing at this indicates an unsuccessful outcome. On the other hand, in WikiSpeedia the users are not forced to this behaviour. Hence they can still finish a game successfully, even after many clicks in the wrong direction. This can cause noise in the data and may deteriorate prediction performance on WikiSpeedia.

6.2 Analysing Certainty of the Models

Based on our experiments and analysis in previous sections, we want to analyse the certainty of our model. It performs better when using a sample closer to the end phase for sequences of the same size. We assume this is caused by the different behaviour of users in successful and unsuccessful sessions towards the end. Hence we expect the model to be more certain at this stage. In contrast we expect it to be less certain if trained on samples closer to the start phase. Hence we trained models on fixed length sequences at different positions in the click trail.

In order to generate these positional sequences, we are using a moving window approach from either the start of the click trail or the end. For both, we are using the structural representation, a fixed sequence size of 5 and generate six sequences (distance to start/target starting at 0 up to 5 in Figure 4). We will further relate to these different distances as "sequence positions". To generate the sequence position 5, the click trail needs to contain at least 10 nodes (5 nodes for the distance and 5 more for the actual sequence). Hence for reasons of comparability between the different sequence positions we remove all click trails with less than 10 clicks. This guarantees the same dataset for all approaches and hence comparability. For each sequence position we trained and evaluated a separate model.

When predicting on the test set, we use the soft-max outputs as a measurement for certainty [30]. For each prediction i we calculated an error e_i, which is the margin between true value t_i to prediction



(a) Using the start phase of WikiSpeedia **(b) Using the end phase of WikiSpeedia** **(c) Using the start phase of WikiGame** **(d) Using the end phase of WikiGame**

Figure 4: Analysis of prediction certainty for different positions of the input sequence. The models are using structural-based input representations and a fixed sequence-size of 5. The x-axis indicates the distance from either the start (Figures 4a and 4c) or the target (Figures 4b and 4d). The black line represents the accuracy, the green, dashed line the certainty on successful games and the red, dotted line the certainty on unsuccessful games.

p_i. We calculate the certainty using:

$$\text{certainty} = 1 - \frac{\sum_i^n |t_i - p_i|}{n} \quad (1)$$

In Figure 4 we show the accuracy and certainty of successful and unsuccessful sessions for the just explained approaches.

Figures 4a and 4c shows the accuracy (black line) of different input positions for the start phase approach . For both models, the overall accuracy stays in the range between 0.7 and 0.8. In Figure 4c, it decreases when distancing further from the start. We assume this behaviour is due to the different game stages, from which the samples are generated. Considering a sequence at position 5, some trails are already on their last clicks towards finishing the game and show a distinct signature. On the other hand, some games are just started and continue issuing more clicks until they finish the game. Hence the model needs to capture at least two different stages, which it is not able to.

The green and red dotted lines represent the certainty for successful and unsuccessful click trails respectively. They are calculated using Equation (1). Both certainty lines remain on the same level for all positions. Overall our model is more certain on successful sessions than on unsuccessful ones. We assume this can be explained by the fact, that successful sessions finish in similar ways, whereas there are many different ways to finish a game unsuccessfully. Hence it is less certain in classifying unsuccessful outcomes.

Figures 4b and 4d show the corresponding results of the end-phase approach. In both datasets (Figure 4b for WikiSpeedia and Figure 4d for WikiGame) the accuracy and both certainty lines are at a climax at the last possible sample. They are continuously decreasing when distancing further from the end, but the models are always more certain to classify successful outcomes than unsuccessful ones. This result meets our expectation, since successful games have a more distinct signature in comparison to unsuccessful games towards the end.

7 CONCLUSION

In this paper we developed a RNN based classifier that is able to predict game success based on partial click trails in Wikipedia navigation games. We evaluate our model on two navigation game datasets from WikiSpeedia and WikiGame. Our model is able to outperform all given baselines with an AUC score of 0.86, including the baseline by Scaria et al. [8], that extracts features using knowledge about the target node, which our approach explicitly discards. Additionally, our model performs even better in terms of AUC (0.90) and Acc (0.82) on WikiGame, a much larger dataset. A deeper analysis of the two datasets reveals significant differences in user navigation behaviour, which in turn highlights the general applicability of our model in different settings. Furthermore, we study the prediction certainty of the proposed model and find that it becomes more confident towards the end of the click trail. In future work, we plan to apply the model to real-world settings, e.g. e-commerce click trails.

REFERENCES

[1] Robert West, Ashwin Paranjape, and Jure Leskovec. 2015. Mining missing hyperlinks from human navigation traces: a case study of wikipedia. *CoRR*.

[2] Philipp Singer, Florian Lemmerich, Robert West, Leila Zia, Ellery Wulczyn, Markus Strohmaier, and Jure Leskovec. 2017. Why we read wikipedia. In *Proceedings of the 26th International Conference on World Wide Web* (WWW '17). International World Wide Web Conferences Steering Committee, Perth, Australia, 1591–1600.

[3] Dimitar Dimitrov, Florian Lemmerich, Fabian Flöck, and Markus Strohmaier. 2018. Query for architecture, click through military: comparing the roles of search and navigation on wikipedia. In *Proceedings of the 10th ACM Conference on Web Science, WebSci 2018, Amsterdam, The Netherlands, May 27-30, 2018*, 371–380.

[4] Ryen W. White and Dan Morris. 2007. Investigating the querying and browsing behavior of advanced search engine users. In *SIGIR*. Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando, (Eds.) ACM, 255–262.

[5] Ryen W. White and Jeff Huang. 2010. Assessing the scenic route: measuring the value of search trails in web logs. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR '10). ACM, Geneva, Switzerland, 587–594.

[6] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. 2004. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '04). ACM, Vienna, Austria, 415–422.

[7] Jane Li, Scott B. Huffman, and Akihito Tokuda. 2009. Good abandonment in mobile and pc internet search. In *32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 43–50.

[8] Aju Thalappillil Scaria, Rose Marie Philip, Robert West, and Jure Leskovec. 2014. The last click: why users give up information network navigation. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 213–222.

[9] C. Trattner, D. Helic, P. Singer, and M. Strohmaier. 2012. Exploring the differences and similarities between hierarchical decentralized search and human navigation in information networks. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*. ACM, 14.

[10] Denis Helic, Markus Strohmaier, Michael Granitzer, and Reinhold Scherer. 2013. Models of human navigation in information networks based on decentralized search. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media* (HT '13). ACM, Paris, France, 89–98.

[11] Daniel Lamprecht, Kristina Lerman, Denis Helic, and Markus Strohmaier. 2017. How the structure of wikipedia articles influences user navigation. *New Review of Hypermedia and Multimedia*, 23, 1, 29–50.

[12] Robert West and Jure Leskovec. 2012. Human wayfinding in information networks. In *Proceedings of the 21st International Conference on World Wide Web* (WWW '12). ACM, Lyon, France, 619–628.

[13] Robert West and Jure Leskovec. 2012. Automatic versus human navigation in information networks. In *ICWSM*. John G. Breslin, Nicole B. Ellison, James G. Shanahan, and Zeynep Tufekci, (Eds.) The AAAI Press.

[14] Robert West. 2016. *Human Navigation of Information Networks*. Ph.D. Dissertation. Stanford University.

[15] Philipp Singer, Thomas Niebler, Markus Strohmaier, and Andreas Hotho. 2013. Computing semantic relatedness from human navigational paths: a case study on wikipedia. *International Journal on Semantic Web and Information Systems*, 9, 4, 41–70.

[16] Alexander Dallmann, Thomas Niebler, Florian Lemmerich, and Andreas Hotho. 2016. Extracting semantics from random walks on wikipedia: comparing learning and counting methods. In *The Workshops of the Tenth International AAAI Conference on Web and Social Media Wiki*. Robert West, Leila Zia, Dario Taraborelli, and Jure Leskovec, (Eds.), 33–40.

[17] Thomas Niebler, Martin Becker, Christian Pölitz, and Andreas Hotho. 2017. Learning semantic relatedness from human feedback using relative relatedness learning. In *Proceedings of the ISWC 2017*. Nadeschda Nikitina, Dezhao Song, Achille Fokoue, and Peter Haase, (Eds.)

[18] Rodrigo Nogueira and Kyunghyun Cho. 2016. End-to-end goal-driven web navigation. In *Advances in Neural Information Processing Systems*, 1903–1911.

[19] Abdigani Diriye, Ryen White, Georg Buscher, and Susan T. Dumais. 2012. Leaving so soon?: understanding and predicting web search abandonment rationales. In *CIKM*. Xue wen Chen, Guy Lebanon, Haixun Wang, and Mohammed J. Zaki, (Eds.) ACM, 1025–1034.

[20] Atish Das Sarma, Sreenivas Gollapudi, and Samuel Ieong. 2008. Bypass rates: reducing query abandonment using negative inferences. In *KDD*. Ying Li, Bing Liu, and Sunita Sarawagi, (Eds.) ACM, 177–185.

[21] L. Han, K. Roitero, U. Gadiraju, C. Sarasua, A. Checco, E. Maddalena, and G. Demartini. 2018. All those wasted hours: on task abandonment in crowdsourcing. © 2019 ACM. (November 2018).

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*.

[23] Andrew M. Dai, Christopher Olah, and Quoc V. Le. 2015. Document embedding with paragraph vectors. *CoRR*, abs/1507.07998. eprint: 1507.07998.

[24] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: large-scale information network embedding. In *WWW*. ACM, Florence, Italy, 1067–1077.

[25] Bryan Perozzi, Rami Al-Rfou', and Steven Skiena. 2014. Deepwalk: online learning of social representations. In *KDD*. Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang 0010, and Rayid Ghani, (Eds.) ACM, 701–710.

[26] Aditya Grover and Jure Leskovec. 2016. Node2vec: scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 855–864.

[27] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9, 8, 1735–1780.

[28] Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[29] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 8, 861–874.

[30] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, 1321–1330.