Fast Heuristic Detection of Offensive Words in Wordwheel Puzzles

Anand D. Blum, R. Mitchell Parry
Appalachian State University
{blumad,parryrm}@appstate.edu

Abstract

Offensive words appear in Wordwheel-type puzzles with a high frequency. Previous approaches to eliminating these words have focused largely on eliminating puzzles that might give rise to an offensive word. This work presents a fast, heuristic approach to detecting an offensive word within a puzzle. After a preprocessing stage, the detection occurs with a single bitwise operation on a 64-bit word. Tests show that as long as there are at least 3 taboo words possible in a puzzle, the heuristic approach is faster than a depth-first search of the puzzle. In addition to being fast, the approach is guaranteed to detect all offensive words, and has a low false positive rate..

Introduction

The Times of London's Polygon game, shown in Figure 1, challenge players to make as many words as possible from the letters in the puzzle. All words must use the center letter, and typically there is a minimum word length, often four or five letters. The puzzles are also commonly referred to as Wordwheel puzzles, for example, in the Irish Examiner (Irish Examiner 2021).

The New York Times (NYT) has a version of the puzzle called Spelling Bee (Ezersky 2021), with some important adaptations. When adapting the Polygon puzzle, the editors at the NYT decided that they would allow letters to be used multiple times when forming words (Amlen 2020). In addition, rather than having nine letters in the puzzle, the Spelling Bee contains only seven.

These puzzles appear in a number of other contexts. A number of online tools are aimed at educators wishing to create puzzles for a classroom setting, see for example (EduGames 2021) and (Muclahy 2021). Wordwheel puzzles have also been used in research studies, including a study of gender bias in evaluation of colleagues (Hochberg 2020).

There are two distinct approaches to the letter placement in these puzzles. One approach, used by the Sunday NYT Magazine puzzle, shows the letters on the outside edge in an alphabetical order, either clockwise or counterclockwise. The other approach presents the outer letters in a random permutation. Indeed, the online version of the Spelling Bee puzzle includes a button that will randomly permute the letters in a puzzle.

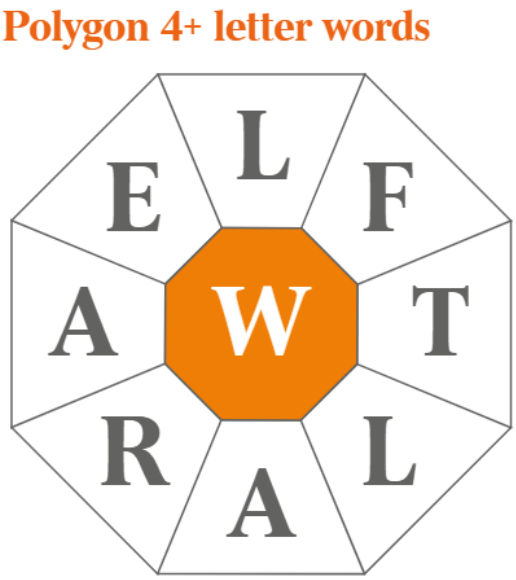


Figure 1. The Polygon Game from the Times of London (Times of London 2021)

Avoiding offensive terms in the puzzles is clearly desirable, particularly given these contexts in which the puzzles are being used. This avoidance can occur at two distinct levels. Certain long words could be avoided if they contain all of the letters in an offensive term. However, there are a large number of instances in the search space for these types of puzzles that contain the letters in one or more offensive terms.

A second approach would place outside letters so that offensive terms are not obvious when looking at the puzzle. A

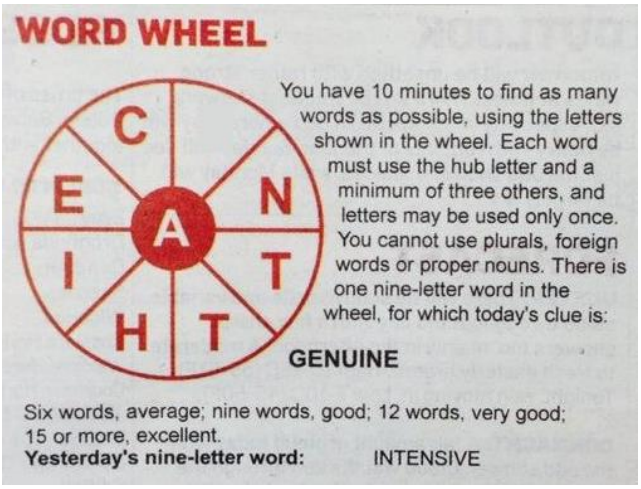


Figure 2. An example of a failure to detect an offensive word from the April 25, 2014 edition of the London Independent. The second letter of the offensive word has been removed. The long word was authentic. (Molloy 2014)

obvious failure to achieve this goal occurred in the print edition of the Irish Independent in 2014, shown in Figure 2.

For the purpose of this research, we define a word as obvious, if it can be made from a simple path through the puzzle. A simple path is made up of contiguous, non-repeated letters in the puzzle. This definition would include any words that appear in order around the outside rim, but also ones that appear in a row of letters in the puzzle, or indeed any path through it. As an example, it would include the word "FLEW" from Figure 1. It would not include "FLAW", since the path from the F to the L, which is next to the A, must skip over a letter.

This paper presents a heuristic designed to quickly detect offensive terms for online games that allow the player to permute the outer letters. The heuristic is designed to be fast. In an online game where the permutations of words is done server-side, this property improves the scalability of the system. The heuristic requires just one bitwise computation on a 64-bit word per offensive word. There is preprocessing of the puzzle prior to the application of the heuristic, which makes the heuristic faster than a depth-first search if there are 3 or more possible taboo words in the puzzle. If there are fewer than 3 taboo words, a standard depth-first search of the puzzle for taboo words is faster.

Moreover, the heuristic is extremely accurate. It has a false negative rate of zero, meaning that it will always flag offensive terms. It has a low false positive rate (FPR). This rate averages 1.25% over all puzzles. Moreover, for more than 80% of puzzles, the FPR is zero in all of the possible configurations of the puzzle.

Background

While there is concern about the unintended effects of offensive language in word puzzles, there is an argument that context is very important when determining if language is actually offensive. As an example, an analysis of NYT crossword puzzles determined that clues and answers contain bias against marginalized groups including the LGBT community, people of color, and women (Graber 2018). The individual words chosen for clues or answers were not so problematic; rather, the bias was found in the relationship of the clues to the answers.

Indeed, when determining if language is offensive, a system to detect hate speech in social media posts determined that context was key in people's determination about whether a particular word or phrase was offensive (Davidson 2017). In the realm of games, an illustration of this principle occurred for the game of Scrabble. In 1993, the maker of the game, Hasbro, announced plans to remove between 50 and 100 offensive words from the official list of acceptable words in Scrabble (Salen and Zimmerman 2005). A large, vocal segment of the Scrabble community objected to this change, arguing that it was unnecessary censorship. Salen and Zimmerman argue that this outcry is a result that the targeted words, when occurring in the context of a game, do not have the offensive meaning that is present in other contexts.

One might argue that the presence of potentially offensive words in Wordwheel puzzles, therefore, is not problematic. As long as these puzzles are not being used in an educational setting, perhaps it is sufficient that the rules, for example for the NYT Spelling Bee, indicate that curse words are not in the dictionary and will not appear in the answer list (Ezersky 2021).

This argument is not supported by the effort that puzzle makers expend in avoiding certain offensive terms. For example, when an offensive word appears in the solution to a potential puzzle, the creators of the Spelling Bee take one of two approaches: they will either leave the word off of the official answer list or, if the word is deemed too offensive, they will create a different puzzle (Amlen 2020). They note however that some letters are so common that some offensive words are unavoidable.

Finding a word made up of unique contiguous cells in a puzzle is an instance of a string search problem. There are algorithms, like the KMP algorithm to quickly find substrings in time that is linear in the length of the string to be searched and the substring (Knuth 1977). However, because the string to be searched is not a single linear string, but rather formed from cells in a graph, one would need to use a graph traversal algorithm to search for substrings. These algorithms would be linear in the number of edges in the graph.

N-gram-based search approach have been applied for string matching in previous research. For example, an n-gram approach was used for information retrieval systems in order to match queries with target documents (Owolabi and McGregor 1988). Similarly, n-gram based searching was used to address the challenge of matching text in noisy documents produced through optical character recognition (Harding *et al.* 1997).

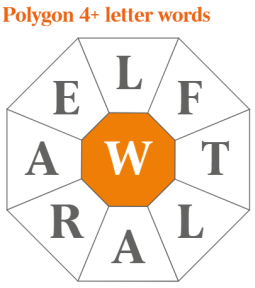
Heuristic for Taboo Word Detection

As noted earlier, the heuristic that we introduce finds possible taboo words that are obvious in the sense that they can be formed by contiguous letters in the puzzle. The heuristic provides for fast detection based on the presence of two-character sequences, or bigrams.

There are two distinct phases in the detection algorithm. First, when a puzzle is selected, a preprocessing phase finds any taboo words that might appear in the puzzle. Note that this portion of the preprocessing phase would also have to be done for any search algorithm. Then, for each taboo word, a mask that represents a subset of the bigrams is created. The second phase occurs, when the position of the letters are set for a puzzle. In this phase, bitwise operations are performed on the mask to detect the possible presence of taboo words.

The preprocessing phase begins by selecting a long word and a center letter. As shown in Figure 3, each unique letter along the outside of the puzzle is assigned a consecutive, non-negative integer, starting at zero. Note that since there are a maximum of eight unique letters, these integers will range from 0 to 7, inclusive.

Then, all of the taboo words are scanned to find those whose letters are a subset of the letters in the puzzle. As shown in Figure 4, for each taboo word that might appear in the puzzle, we find only the bigrams needed from the outside letters. In the figure, we are assuming that the taboo word is "wallart," rather than using an offensive word in the



Letter	Integer Value
A	0
E	1
F	2
L	3
R	4
T	5

Figure 3. Step 1 assigns integer values to letters around the outside of the puzzle.

Bigrams	Count
AL	2
LL	1
AR	1
RT	1

Figure 4. The bigrams needed to make the word "wallart" from letters in the outside of the puzzle. Notes: we ignore the bigrams with the center letter; and we put the letters in a bigram in alphabetical order, rather than the order that the appear in the puzzle.

example. We ignore the bigrams that are made with the center letter, because these bigrams will be present in all possible letter placements, once the center letter is fixed.

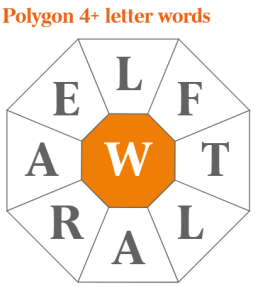
To construct the letter mask, we find a pair (i, j) that represents the two letters in the bigram, where i represents the smaller of the integers assigned to the letters in the bigram, and j represents the larger of the integers assigned to the letters in the bigram. The bit in the mask at position, $8i + j$, is set to 1 if there is at least 1 bigram of type (i, j) needed for the taboo word. If $i \neq j$, the bit in the mask at position $8j + i$ is set to 1 if there are at least 2 bigrams of (i, j) needed for the taboo word. Table 1 shows the masks for our running example.

When the positions for the letters for a potential puzzle configuration are chosen, the heuristic creates a mask from the bigrams present in the consecutive letters on the outside ring of the puzzle configuration. As shown in Figure 5 and Table 2, the mask is constructed in the same way as for the taboo words.

When the positions for the letters for a potential puzzle configuration are chosen, the heuristic creates a mask from

Bigram	Letter Codes	Count	Set Bit Position
AL	0,3	2	3 (8·0+3) and 24 (8·3+0)
LL	3,3	1	27 (8·3+3)
AR	0,4	1	4 (8·0+4)
RT	4,5	1	37 (8·4+5)

Table 1. The construction of the mask for the taboo word "wallart". Note that two bits are set to 1 for the bigram "AL", since it occurs twice. The mask would consist of bits that are all zero, except for the bit at the 3, 4, 24, 27, and 37th bit positions.



Bigrams	Count
FL	1
FT	1
LT	1
AL	1
AR	2
AE	1
EL	1

Figure 5. The bigrams present in consecutive letters around the outside of the puzzle

Bigram	Letter Codes	Count	Set Bit Position
FL	2,3	1	19 (8·2+3)
FT	2,5	1	21 (8·2+5)
LT	3,5	1	29 (8·3+5)
AL	0,3	1	3 (8·0+3)
AR	0,4	2	4 (8·0+4) and 32 (8·4+0)
AE	0,1	1	1 (8·0+1)
EL	1,3	1	11 (8·1+3)

Table 2. The construction of the mask for the puzzle configuration shown in Figure 5. The mask would consist of bits that are all zero, except for the bit at the 1, 3, 4, 11, 19, 21, 29, and 32nd bit positions.

the bigrams present in the consecutive letters on the outside ring of the puzzle configuration. As shown in Figure 5 and Table 2, the mask is constructed in the same way as for the taboo words.

To check to see if a taboo word may be present in the puzzle configuration, we bitwise-and the puzzle mask with the taboo word mask. If the result is equal to the taboo word mask, all of the bigrams required for the taboo word are present. The heuristic, then, would reject the puzzle configuration because of the possible presence of a taboo word in a path of contiguous cells in the puzzle.

In our example, our taboo word mask had bits set at the following positions: (3, 4, 24, 27, 37). Our puzzle had bits set at positions (1, 3, 4, 11, 19, 21, 29, 32). The puzzle mask contains the required set bits at position 3 and 4, corresponding to the bigrams AR and AL. However, it is missing the needed bit at position 24 (for the second AL bigram in "wallart"), 27 for the bigram LL, and 37 for the bigram RT.

		Predicted		
		Positive	**Negative**	
Actual	**Positive**	123,529,608	0	True Positive Rate 100%
	Negative	11,852,352	933,591,960	False Positive Rate 1.25%
		Precision 91.2%	Specificity 98.7%	Accuracy 98.9%

Table 3. Confusion matrix for the heuristic detection of taboo words in a puzzle configuration.

Results

In order to evaluate the heuristic, we examined all possible Wordwheel puzzles that could be created from the word list in the Linux Operating System. For the taboo word list, we used a list of 327 words that reportedly have been banned by Google (Gabriel 2021).

The heuristic is guaranteed to produce no false negatives, and detect all taboo words. However, it is possible for the heuristic to produce false positives, in which the heuristic flags a puzzle configuration for a possible taboo word that is not present.

The false positives arise when there are repeated letters in the puzzle. While the puzzle masks encode all of the adjacent letters in the puzzle configuration, if there are repeated letters, which letters are actually adjacent becomes ambiguous. As a result, the heuristic performs perfectly for NYT-style puzzles, which do not allow for repeated letters in the puzzle.

Consider for example the puzzle configuration in the figure 5. Consider the taboo word is "rale", with the required bigrams AR, AL, and EL. Since all of the bigrams appear in the outside of the puzzle, the heuristic would incorrectly flag this puzzle configuration as containing the word "rale." The error occurs because the L that is adjacent to A in the puzzle configuration is different than the L that is adjacent to E.

There are 5,286 9-letter words, using words that start with a lowercase letter in the word list provided with the Linux operating system. If we ignore the ordering of the outside letters, 38,270 distinct puzzles are possible. Note that we have fewer than 9 · 5,286, or 47,574, puzzles, because some puzzles share multiple 9-letter words, and puzzles that have duplicated letters have fewer than 9 possible puzzles, since placing any duplicated letter in the center creates the same puzzle.

We generated all possible permutations of the letters in the outside of these puzzles, and examined the performance of the heuristic in detecting taboo words. As shown in Table

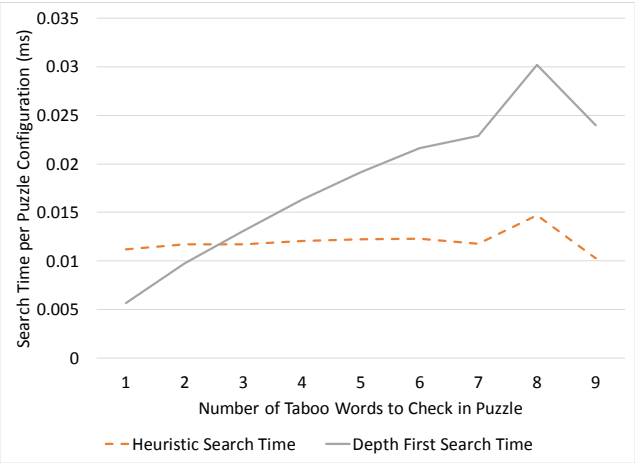


Figure 6. Comparison of the running time for the heuristic and a depth-first search algorithm as a function of the number of taboo words that may be present in a puzzle.

3, the heuristic had a true positive rate of 100%, a false positive rate of 1.25%, a precision of 91.2%, a specificity of 98.7%, and an accuracy of 98.9%. Table 1 also shows that Taboo words appear commonly in an obvious form, appearing in 11.6% of puzzle configurations.

We can also detect the puzzles for which the heuristic performs perfectly. Puzzles can contain multiple taboo words, and each puzzle has 40,320 (8!) permutations of the outer letters. Of the 38,270 possible puzzles, the heuristic had a false positive in at least one configuration in 7,152 (18.7%) of them. The heuristic performed perfectly in more than 80% of the puzzles. Moreover, given the low false positive rate, the heuristic correctly flags puzzle configurations that contain taboo words in many configurations in the 18.7% where there was one or more false positives.

In order to evaluate the speed of the heuristic, tests were run with all configurations of puzzles comparing the running time of a depth-first search for taboo words and the heuristic. Before both algorithms, a list of taboo words that could appear in the puzzle was collected and bigram masks were constructed for the taboo words. Note that the construction of theses masks occur only once for puzzle, and then they can be reused for all permutations of outside letters. Therefore, this processing time is not considered.

As shown in Figure 6, the time taken by the depth first search algorithm grows much faster than the time taken by the heuristic, as the number of taboo words increases. Due to the preprocessing needed for a puzzle configuration, in particular the construction of the bigrams for the outside letters in a puzzle, the depth-first search algorithm is faster for puzzles with two or fewer taboo words. However, the construction of this bigram mask does not depend on the number of taboo words, resulting in the lack of growth in the time needed for the heuristic. It is interesting that both algorithms require less time to detect puzzles with 9 possible taboo words than puzzles with 8 possible taboo words. It may be the result of the algorithms terminating as soon as they detect a taboo word. As the number of taboo words increases, it becomes more likely that the algorithms will terminate without having to examine all words. It is also possible that this phenomenon is the result of the particular puzzle letters required to have 9 possible taboo words.

Conclusions and Future Work

A heuristic, based on two letter bigrams, performs well for detecting the presence of taboo words in adjacent cells of Wordwheel puzzles. These taboo words appear often in contiguous, non-repeating cells of these puzzles. With the wordlists considered by this paper, they appeared in 11.6% of puzzle configurations.

This heuristic has a low false positive rate of 1.25% and works well for the vast majority of Wordwheel puzzle configurations. Provided that there are sufficient taboo words, the heuristic is fast, with the detection done with bitwise operations, and thus its speed is advantageous for minimizing server-side processing for online games.

Wordwheel puzzles offer a range of further, interesting opportunities for AI-assisted game design. The authors are interested, for example, in exploring the effect of letter placements on the difficulty for people to find words. If some configurations make it too easy to solve the puzzle, the heuristic approach presented here might be applicable in choosing good configurations that avoid not only taboo words, but also puzzle configurations that are too easy to solve.

References

Amlen, D. 2020. The Genius of Spelling Bee. The New York Times. https://www.nytimes.com/2020/10/16/crosswords/spellingbee-puzzles.html. Accessed: 2020-10-16.

Davidson, T. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. Association for the Advancement of Artificial Intelligence, 512-515.

Ezersky, S. 2021. Spelling Bee. New York Times. https://www.nytimes.com/puzzles/spelling-bee. Accessed: 2021-06-21

Graber, S. M. 2018. A Puzzling Problem. Journalism Practice. doi:10.1080/17512786.2018.1478745

Harding S.M., Croft W.B., Weir C. 1997. Probabilistic retrieval of OCR degraded text using N-grams. In: Peters C., Thanos C. (eds) Research and Advanced Technology for

Digital Libraries. ECDL 1997. Lecture Notes in Computer Science, vol 1324. Springer, Berlin, Heidelberg.

Hochberg, J. 2020. Evaluating High Performance Female Colleagues: The Roles of Race, Gender, and Task Performance. Ann Arbor: University of Michigan. Retrieved from http://hdl.handle.net/2027.42/155356.

Irish Examiner. 2021. Wordwheel. https://www.irishexaminer.com/puzzles/wordwheel/. Accessed 2021-06-26.

Knuth, D. 1977. Fast pattern matching in strings. SIAM Journal on Computing, 6(2), 323-350.

Molloy, D. 2014. The Indo's word wheel today is … oh dear. Irish Examiner. Issue Date: April 25.

Muclahy, D. 2021. Word Wheel. Retrieved from ESLKids Games: https://eslkidsgames.com/word-wheel. Accessed: 2021-06-26.

NYT Bee. 2021. NYT Spelling Bee Answers and Analysis. http://www.nytbee.com. Accessed: 2021-07-21.

Owolabi, O., McGregor, D.R. 1988. Fast approximate string matching. Software: Practice and Experience. 19(4), 387-393.

Times of London. 2021. Polygon 4+ letter words. https://www.thetimes.co.uk/puzzles/polygon-lzxjvg6ss. Accessed: 2021-06-26.

Edu-Games. 2021. Word Wheel. 2021. https://www.edu-games.org/word-games/word-wheel/word-wheel.php. Accessed 2021-06-26.

K. Salen, K.E. Zimmerman. 2005. Game design and meaningful play. Handbook of computer game studies, 59, 79.