Automated Crossword Solving

Eric Wallace[⋆] UC Berkeley
Nicholas Tomlin[⋆] UC Berkeley
Albert Xu[⋆] UC Berkeley
Kevin Yang[⋆] UC Berkeley
Eshaan Pathak[⋆] UC Berkeley
Matthew L. Ginsberg Matthew Ginsberg, LLC
Dan Klein UC Berkeley
{ericwallace, nicholas_tomlin, albertxu3, klein}@berkeley.edu

Abstract

We present the Berkeley Crossword Solver, a state-of-the-art approach for automatically solving crossword puzzles. Our system works by generating answer candidates for each crossword clue using neural question answering models and then combines loopy belief propagation with local search to find full puzzle solutions. Compared to existing approaches, our system improves exact puzzle accuracy from 57% to 82% on crosswords from *The New York Times* and obtains 99.9% letter accuracy on themeless puzzles. Our system also won first place at the top human crossword tournament, which marks the first time that a computer program has surpassed human performance at this event. To facilitate research on question answering and crossword solving, we analyze our system's remaining errors and release a dataset of over six million question-answer pairs.

1 Introduction

> *"The key to solving crosswords is mental flexibility. If one answer doesn't seem to be working out, try something else."*
>
> — Will Shortz, NYT Crossword Editor

Crossword puzzles are perhaps the world's most popular language game, with millions of solvers in the United States alone (Ginsberg, 2011). Crosswords test knowledge of word meanings, trivia, commonsense, and wordplay, while also requiring one to simultaneously reason about multiple intersecting answers. Consequently, crossword puzzles provide a testbed to study open problems in AI and NLP, ranging from question answering to search and constraint satisfaction. In this paper, we describe an end-to-end system for solving crossword puzzles that tackles many of these challenges.

Figure 1: A partially-solved example crossword puzzle from the 2021 American Crossword Puzzle Tournament, where our system won first place against 1,100 top human solvers. The highlighted fill KUNGFU answers the wordplay clue: *Something done for kicks?*

1.1 The Crossword Solving Problem

Crossword puzzles are word games consisting of rectangular grids of squares that are to be filled in with letters based on given clues (e.g., Figure 1). Puzzles typically consist of 60–80 clues that vary in difficulty due to the presence of complex wordplay, intentionally ambiguous clues, or esoteric knowledge. Each grid cell belongs to two words, meaning that one must jointly reason about answers to multiple questions. Most players complete crosswords that are published daily in newspapers and magazines such as *The New York Times* (NYT), while other more expert enthusiasts also compete in live events such as the American Crossword Puzzle Tournament (ACPT). These events are intensely competitive: one previous winner reportedly solved twenty puzzles per day as practice (Grady, 2010), and top competitors can perfectly solve expert-level puzzles with over 100 clues in just 3 minutes.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 3073 - 3085

Category	Clue	Answer	QA Recall
Knowledge (37%)	Birds on Minnesota state quarters	LOONS	✔
	Architect Frank	GEHRY	✔
Definition (33%)	First in a series	PILOT	✔
	Tusked savanna dweller	WARTHOG	✔
Commonsense (14%)	Like games decided by buzzer beaters	CLOSE	✔
	Opposite of *luego*	AHORA	✔
Wordplay (8%)	Frequent book setting	SHELF	✔
	One followed by nothing?	TEN	✗
Phrase (8%)	"Is it still a date?"	AREWEON	✔
	"Post ___ analysis"	HOC	✔
Cross-Reference (2%)	See Capital of 52-Down	GHANA	✗
	Oft-wished-upon sighting	SHOOTINGMETEOR	✗

Table 1: Types of reasoning used in *The New York Times Crossword*. We compute each type's frequency by manually analyzing 200 clues. See Appendix A for category definitions. We also indicate if our QA model correctly predicts each answer based on top-1000 recall. Cross-reference clues mention other clues or themes, e.g., SHOOTINGMETEOR replaces the clued phrase SHOOTINGSTAR based on the context from the puzzle.

Automated crossword solvers have been built in the past and can outperform most hobbyist humans. Two of the best such systems are Proverb (Littman et al., 2002) and Dr. Fill (Ginsberg, 2011). Despite their reasonable success, past systems struggle to solve the difficult linguistic phenomena present in crosswords, and they fail to outperform expert humans. At the time of their respective publications, Proverb achieved 213th place out of 252 in the ACPT, while Dr. Fill achieved 43rd place.

1.2 A Testbed for Question Answering

Answering crossword clues involves challenges not found in traditional question answering (QA) benchmarks. The clues are typically less literal; they span different reasoning types (c.f., Table 1); and they cover diverse linguistic phenomena such as polysemy, homophony, puns, and other types of wordplay. Many crossword clues are also intentionally underspecified, and to solve them, one must be able to "know what they don't know" and defer answering those clues until crossing letters are known. Crosswords are also useful from a practical perspective as the data is abundant, well-validated, diverse, and constantly evolving. In particular, there are millions of question-answer pairs online, and unlike crowdsourced datasets that are often rife with artifacts (Gururangan et al., 2018; Min et al., 2019), crossword clues are written and validated by experts. Finally, crossword data is diverse as it spans many years of pop culture, is written by thousands of different constructors, and contains various publisher-specific idiosyncrasies.

1.3 A Testbed For Constraint Satisfaction

Solving crosswords goes beyond just generating answers to each clue. Without guidance from a constraint solver, QA models cannot reconcile crossing letter and length constraints. Satisfying these constraints is challenging because the search space is enormous and many valid solutions exist, only one of which is correct. Moreover, due to miscalibration in the QA model predictions, exact inference may also lead to solutions that are high-likelihood but completely incorrect, similar to other types of structured decoding problems in NLP (Stahlberg and Byrne, 2019; Kumar and Sarawagi, 2019). Finally, the challenges in search are amplified by the unique long tail of crossword answers, e.g., "*daaa bears*" or "*eeny meeny miny moe*," which makes it highly insufficient to restrict the search space to solutions that contain only common English words.

1.4 The Berkeley Crossword Solver

We present the Berkeley Crossword Solver (BCS), which is summarized in Figure 2. The BCS is based on the principle that some clues are difficult to answer without any letter constraints, but other (easier) clues are more standalone. This naturally motivates a multi-stage solving approach, where we first generate answers for each question independently, fill in the puzzle using those answers, and then rescore uncertain answers while conditioning on the predicted letter constraints. We refer to these stages as first-pass QA, constraint resolution, and local search, and we describe each component in Sections 3–5 after describing our dataset

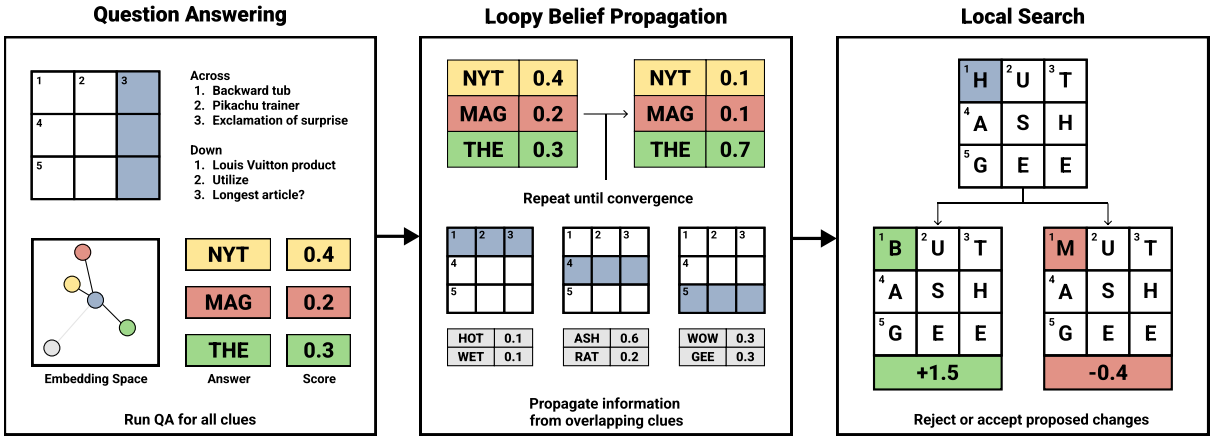


Figure 2: *An overview of the Berkeley Crossword Solver.* We use a neural question answering model to generate answer probabilities for each question, and then refine the probabilities with loopy belief propagation. Finally, we fill the grid with greedy search and iteratively improve uncertain areas of the puzzle using local search.

in Section 2. In Section 6, we show that the BCS substantially improves over the previous state-of-the-art Dr. Fill system, perfectly solving 82% of crosswords from *The New York Times*, compared to 57% for Dr. Fill. Nevertheless, room for additional improvement remains, especially on the QA front. To facilitate further exploration, we publicly release our code, models, and dataset: https://github.com/albertkx/berkeley-crossword-solver.

2 Crossword Dataset

This section describes the dataset that we built for training and evaluating crossword solving systems. Recall that a crossword puzzle contains both question-answer pairs and an arrangement of those pairs into a grid (e.g., Figure 1). Unfortunately, complete crossword puzzles are protected under copyright agreements; however, their individual question-answer pairs are free-to-use. Our dataset efforts thus focused on collecting numerous question-answer pairs (Section 2.1) and we collected a smaller set of complete puzzle grids to use for final evaluation (Section 2.2).

2.1 Collecting Question-Answer Pairs

We collected a dataset of over six million question-answer pairs from top online publishers such as *The New York Times*, *The LA Times*, and *USA Today*. We show qualitative examples in Table 1, summary statistics in Table 2, and additional breakdowns in Appendix B. Compared to existing QA datasets, our crossword dataset represents a unique and challenging testbed as it is large and carefully labeled, is varied in authorship, spans over 70 years of pop culture, and contains examples that are difficult for even expert humans. We built validation and test sets by splitting off every question-answer pair used in the 2020 and 2021 NYT puzzles. We use recent NYT puzzles for evaluation because the NYT is the most popular and well-validated crossword publisher, and because using newer puzzles helps to evaluate temporal distribution shift.

Word Segmentation of Answers Crossword answers are canonically filled in using all capital letters and without spaces or punctuation, e.g., "*whale that stinks*" becomes WHALETHATSTINKS. These unsegmented answers may confuse neural QA models that are pretrained on natural English text that is tokenized into wordpieces. To remedy this, we trained a word segmentation model that maps the clues to their natural language form.[1] We collected segmentation training data by retrieving common n-grams from Wikipedia and removing their spaces and punctuation. We then finetuned GPT-2 small (Radford et al., 2019) to generate the segmented n-gram given its unsegmented version. We ran the segmenter on all answers in our data. In all our experiments, we train our QA models using segmented answers and we post-hoc remove spaces and punctuation from their predictions.

2.2 Collecting Complete Crossword Puzzles

To evaluate our final crossword solver, we collected a validation and test set of complete 2020 and 2021 puzzle grids. We use puzzles from *The New York Times*, *The LA Times*, *Newsday*, *The New Yorker*, and *The Atlantic*. Using multiple publishers for

[1]More simplistic algorithms that segment the answer into known English words are insufficient for many crossword answers, e.g., DAAABEARS and EENYMEENYMINYMOE.

	Train	Validation	Test
QA Pairs	6.4M	30.4K	21.3K
Answer Set	437.8K	17.2K	13.4K
Timeframe	1951-2019	2020	2021

Table 2: *Summary statistics of our QA dataset.* We collect question-answer pairs from 26 sources (*The LA Times*, *The New York Times*, etc.) for training, and we hold out the latest data from NYT for validation and testing. Our dataset is large and contains a wide range of authors, answers, puzzle sources, and years.

evaluation provides a unique challenge as each publisher contains different idiosyncrasies, answer distributions, and crossword styles. We use 2020 NYT as our validation set and hold out all other puzzles for testing. There are 430 total test puzzles.

3 Bi-Encoder QA Model

The initial step of the BCS is question answering: we generate a list of possible answer candidates and their associated probabilities for each clue. A key requirement for this QA model is that it does not output unreasonable or overly confident answers for hard clues. Instead, this model is designed to be used as a "first-pass" that generates reasonable candidates for every clue, in hope that harder clues can be reconciled later when predicted letter constraints are available. We achieve this by restricting our first-pass QA model to only output answers that are present in the training set. As discussed in Section 5, we later generate answers outside of this closed-book set with our second-pass QA model.

Model Architecture We build our QA model based on a bi-encoder architecture (Bromley et al., 1994; Karpukhin et al., 2020) due to its ability to score numerous answers efficiently and learn using few examples per answer. We have two neural network encoders: $\mathrm{E_C}(\cdot)$, the clue encoder, and $\mathrm{E_A}(\cdot)$, the answer encoder. Both encoders are initialized with BERT-base-uncased (Devlin et al., 2019) and output the encoder's [CLS] representation as the final encoding. These two encoders are trained to map the questions and answers into the same feature space. Given a clue c, the model scores all possible answers a_i using a dot product similarity function between feature vectors: $\mathrm{sim}(c, a_i) = \mathrm{E_C}(c)^{\mathrm{T}}\mathrm{E_A}(a_i)$. Our answer set consists of the 437.8K answers in the training data.[2]

Training We train the encoders in the same fashion as DPR (Karpukhin et al., 2020): batches consist of clues, answers, and "distractor" answers. The two encoders are trained jointly to assign a high similarity to the correct question-answer pairs and low similarity to all other pairs formed between the clue and distractor answers. We use one distractor answer per clue that we collect by searching each clue in the training set using TFIDF and returning the top incorrect answer. We tune hyperparameters of our bi-encoder model based on its top-k accuracy on the NYT validation set.

Inference At test time, for each clue c, we compute the embedding $v_c = E_C(c)$ and retrieve the answers whose embeddings have the highest dot product similarity with v_c. We obtain probabilities for each answer by softmaxing the dot product scores. To speed up inference, we precompute the answer embeddings and use FAISS (Johnson et al., 2019) for similarity scoring.

3.1 Top-k Recall of Our QA Model

To evaluate our bi-encoder, we compute its top-k recall on the question-answer pairs from the NYT test set. We are most interested in top-1000 recall, as we found it to be highly-correlated with downstream solving performance (discussed in Section 7). As a baseline, we compare against the QA portion of the previous state-of-the-art Dr. Fill crossword solver (Ginsberg, 2011). This QA model works by ensembling TFIDF-like scoring and numerous additional modules (e.g., synonym matching, POS matching). Our bi-encoder model considerably outperforms Dr. Fill, improving top-1000 recall from 81.2% to 94.6% (Figure 3). Also note that approximately 4% of test answers are not seen during training, and thus the oracle recall for our first-pass QA model is $\approx 96\%$.

4 Resolving Letter Constraints Using BP

Given the list of answer candidates and their associated probabilities from the first-pass QA model, we next built a solver that produces a puzzle solution that satisfies the letter constraints. Formally, crossword solving is a weighted constraint satisfaction problem, where the probability over solutions is given by the product of the confidence scores produced by the QA model (Ginsberg, 2011). There

[2]Our bi-encoder model is a "closed-book" QA model because it does not have "open-book" access to external knowledge sources such as Wikipedia (Roberts et al., 2020). We found in preliminary experiments that open-book models struggle as most crossword answers are not present or are difficult to retrieve from knowledge sources such as Wikipedia.

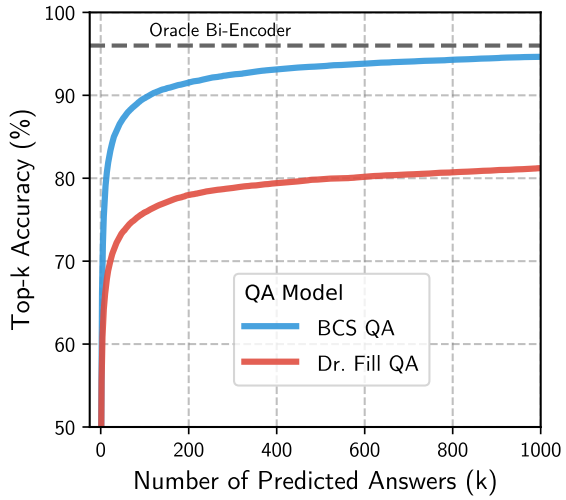


Figure 3: *Top-k accuracy on the 2021 NYT test set*. Dr. Fill QA is an existing crossword QA system that ensembles TFIDF-like scoring with numerous additional scoring modules. Our neural bi-encoder model improves top-1000 accuracy from 81.2% to 94.6%.

are numerous algorithms for solving such problems, including branch-and-bound, integer linear programming, and more.

We use belief propagation (Pearl, 1988), henceforth BP, for two reasons. First, BP directly searches for the puzzle with the highest *expected overlap* with the ground-truth puzzle, rather than the puzzle with the highest likelihood under the QA model (Littman et al., 2002). This is advantageous as it maximizes the total number of correct words and letters in the solution, and it also avoids strange solutions that may have spuriously high scores under the QA model. Second, BP also produces marginal distributions over words and characters, which is useful for generating an n-best list of puzzle candidates (used in Section 5).

Loopy Belief Propagation We use loopy BP, inspired by the Proverb crossword solver (Littman et al., 2002). That is, we construct a bipartite graph with nodes for each of the crossword's clues and cells. For each clue node, we connect it via an edge to each of its associated cell nodes (e.g., a 5-letter clue will have degree 5 in the constructed graph). Each clue node maintains a belief state over answers for that clue, which is initialized using a mixture of the QA model's probabilities and a unigram letter LM.[3] Each cell node maintains a belief state over letters for that cell. We then iteratively apply BP with each iteration doing message passing for all clue nodes in parallel and then for all cell nodes in parallel. The algorithm empirically converges after 5–10 iterations and completes in just 10 seconds on a single-threaded Python process.

Greedy Inference BP produces a marginal distribution over words for each clue. To generate an actual puzzle solution, we run greedy search where we first fill in the answer with the highest marginal likelihood, remove any crossing answers that do not share the same letter, and repeat.

5 Iteratively Improving Puzzle Solutions

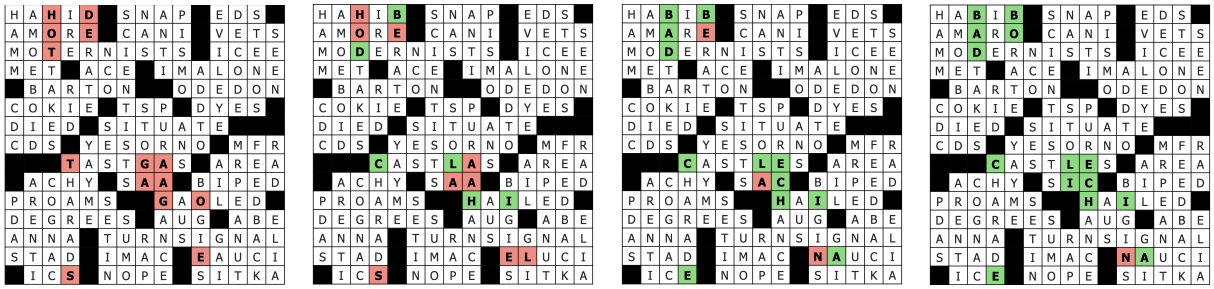
Many of the puzzle solutions generated by BP are close to correct but have small letter mistakes, e.g., NAUCI instead of FAUCI or TAZOAMBASSADORS instead of JAZZAMBASSADORS, as shown in Figure 4.[4] We remedy this in the final stage of the BCS with local search (LS), where we take a "second-pass" through the puzzle and score alternate proposals that are a small edit distance away from the BP solution. In particular, we alternate between proposing new candidate solutions by flipping uncertain letters and scoring those proposals using a second-pass QA model.

Proposing Alternate Solutions Similar to related problems in structured prediction (Stahlberg and Byrne, 2019) or model-based optimization (Fu and Levine, 2021), the key challenge in searching for alternate puzzle solutions is to avoid false positives and adversarial inputs. If we score *every* proposal within a small edit distance to the original, we are bound to find nonsensical character flips that nevertheless lead to higher model scores. We avoid this by only scoring proposals that are within a 2-letter edit distance and also have nontrivial likelihoods according to BP or a dictionary. Specifically, we score all proposals whose 1–2 modified letters each have probability 0.01 or greater under the character marginal probabilities produced by BP.[5] We also score all proposals whose 1–2 modified letters

[3]The unigram letter LM accounts for the probability that an answer is not in our answer set. We build the LM by counting the frequency of each letter in our QA training set.

[4]These errors stem from multiple sources. First, 4% of the answers in a test crossword are not present in our bi-encoder's answer set. Those answers will be not be filled in correctly unless the solver can identify the correct answer for *all* of the crossing answers. Second, natural QA errors exist even on questions with non-novel answers. Finally, the BP algorithm may converge to a sub-optimal solution.

[5]The character-level marginal distribution for most characters assigns all probability mass to a single letter after a few iterations of BP (e.g., probability 0.9999). We empirically chose 0.01 as it achieved the highest validation accuracy.



(a) Before Local Search (b) Step #1 (c) Step #2 (d) Step #3

Figure 4: We show the result of our solver on a NYT puzzle after running greedy search and three consecutive steps of local search. Local search considerably improves accuracy but fails to fix the answer regarding Dr. Fauci (an error due to temporal shift in our QA models). Red squares indicate errors from the output of greedy search, while green squares indicate corrections from the local search. See Figure 12 for the clues and associated answers in the puzzle.

cause the corresponding answer to segment into valid English words.[6]

Scoring Solutions With Second-Pass QA Given the alternate puzzle solutions, we could feed each of them into our bi-encoder model for scoring. However, we found that bi-encoders are not robust—they sometimes produce high-confidence predictions for the nonsensical answers present in some candidate solutions. We instead use generative QA models to score the proposed candidates as we found these models to be empirically more robust. We finetuned the character-level model ByT5-small (Xue et al., 2022) on our training set to generate the answer from a given clue. We then score each proposed candidate using the product of the model's likelihoods of the answers given the clues, $\prod_j P(a_j \mid c_j)$.

After scoring all candidate proposals, we apply the best-scoring edit and repeat the proposal and scoring process until no better edits exist. Figure 4 shows an example of the candidates accepted by LS. Quantitatively, we found that LS applied 243 edits that improved accuracy and 31 edits that hurt accuracy across 255 NYT test puzzles.

6 End-to-End System Results

We evaluated our final system on our set of test puzzles and compare the results to the state-of-the-art Dr. Fill system (Ginsberg, 2011).[7] We compute three accuracy metrics: perfect puzzle, word, and letter. Perfect puzzle accuracy requires answering every clue in the puzzle correctly and serves as our primary—and most challenging—metric.

Table 3 shows our main results. We outperform Dr. Fill on perfect puzzle accuracy across crosswords from every publication source. For example, we obtain a 24.8% absolute improvement on perfect puzzle accuracy on crossword puzzles from *The New York Times*, which is a statistically significant improvement ($p < 0.01$) according to a paired t-test. We also observe comparable or better word and letter accuracies than Dr. Fill across all sources. Our improvement on puzzles from *The New Yorker* is relatively small; this discrepancy is possibly due to the small amount of data from *The New Yorker* in our training set (see Figure 7).

Themed vs. Themeless Puzzles Although the BCS achieves equivalent or worse letter accuracy on *Newsday* and *LA Times* puzzles, it obtains substantially higher puzzle accuracy on these splits. We attribute this behavior to errors concentrated in unique themed puzzles, e.g., ones that place multiple letters into a single cell. To test this, we break down NYT puzzles into those with and without special theme entries (see Appendix D for our definition of theme puzzles). On themeless NYT puzzles, we achieve 99.9% letter accuracy and 89.5% perfect puzzles, showing that themed puzzles are a major source of our errors. Note that the Dr. Fill system includes various methods to detect and resolve themes and is thus more competitive on such puzzles, although it still underperforms our system.

American Crossword Puzzle Tournament For our last evaluation, we competed live in the Amer-

[6]For instance, given a puzzle that contains a fill such as MUNNYANDCLYDE, we consider alternate solutions that contain answers such as BUNNYANDCLYDE and SUNNYANDCLYDE, as they segment to "*bunny and clyde*" and "*sunny and clyde*."

[7]Note that while the original Dr. Fill paper was published in 2011, the system has been consistently updated and has substantially improved. Dr. Fill can outperform all but the best human solvers (see Table 5 for statistics on its improvement). We run the latest system.

Source	# Puzzles	Perfect Puzzle (%)		Word Acc. (%)		Letter Acc. (%)	
		Dr. Fill	BCS	Dr. Fill	BCS	Dr. Fill	BCS
The Atlantic	47	76.6	**89.1**	98.5	**99.1**	99.7	**99.8**
Newsday	52	88.5	**94.2**	**99.7**	99.6	**99.9**	99.8
The New Yorker	22	77.3	**78.3**	98.2	**99.0**	99.6	**99.8**
The LA Times	54	87.0	**92.6**	99.6	**99.7**	99.9	99.9
The New York Times	255	56.9	**81.7**	98.1	**98.9**	99.4	**99.7**

Table 3: *Final results of the Berkeley Crossword Solver.* We compare the BCS to Dr. Fill (Ginsberg, 2011), the previous state-of-the-art crossword solving system, on a range of puzzle sources. The BCS produces significantly more perfect puzzles and achieves better or comparable letter-level and word-level accuracies.

ican Crossword Puzzle Tournament (ACPT), the longest-running and most prestigious human crossword tournament. We obtained special permission from the organizers to compete in the 2021 version of the tournament against 1,100 top human competitors. For the live tournament, we used a "version 1.0" of our system, which does not use belief propagation or local search but instead uses Dr. Fill's constraint-resolution system. Our system won first place—we had a total score of 12,825 compared to the top human who had 12,810 (scoring details in Appendix C). Figure 5 shows our scores compared to the top and median human competitor on the 7 puzzles used in the competition. We also retrospectively evaluated our final BCS system (i.e., using our solver based on belief propagation and local search), and achieved a higher total score of 13,065. This corresponds to getting 6 out of the 7 puzzles perfect and 1 letter wrong on 1 puzzle.

System Ablations We also investigated the importance of our QA model, BP inference, and local search with an ablation study. Table 4 shows results for perfect puzzle accuracy on NYT 2021 puzzles under different settings. The first ablation shows that our local search step is crucial for our solver to achieve high accuracy. The second and third ablations show that the BCS's QA and solver are both superior to their counterparts from Dr. Fill—swapping out either component hurts accuracy.

7 Error Analysis

Our system outperforms the best human solvers; does this mean that crosswords are solved? The answer is, of course, no. In this section, we show that substantial headroom remains on QA accuracy and the handling of special themed puzzles.

QA Error Analysis We first measured how well a QA model needs to perform on each clue in order for our solver to find the correct solution. We found that when our QA model ranks the true answer within the top 1,000 predictions, the answer is almost always filled in correctly (Figure 11). Despite top-1000 accuracy typically being sufficient, our QA model still makes numerous errors. We manually analyzed these mistakes by sampling 200 errors from the NYT 2021 puzzles and placing them in the same categories used in Table 1. Figure 6 shows the results and indicates that knowledge, wordplay, and cross-reference clues make up the majority of errors.

System	**Puzzle (%)**
BCS QA + BP + LS	81.7
BCS QA + BP	44.3
BCS QA + Dr. Fill Solver	73.7
Dr. Fill QA + Dr. Fill Solver	56.9

Table 4: *Ablations on NYT puzzles.* Our full system consists of a bi-encoder QA model, loopy belief propagation (BP), and local search (LS). We find that our QA and solver are both superior to that of Dr. Fill and that our local search step is key to achieving high accuracy.

End-to-end Analysis We next analyzed the errors for our full system. There are 43 NYT 2021 puzzles that we did not solve perfectly. We manually separated these puzzles into four categories:

- **Themes** (21 puzzles). Puzzles with unique themes, e.g., placing four characters in one cell.
- **Local Search Proposals** (9 puzzles). Puzzles where we did not propose a puzzle edit in local search that would have improved accuracy.
- **Local Search Scoring** (9 puzzles). Puzzles where the ByT5 scorer either rejected a correct proposal or accepted an incorrect proposal.
- **Connected Errors** (4 puzzles). Puzzles with

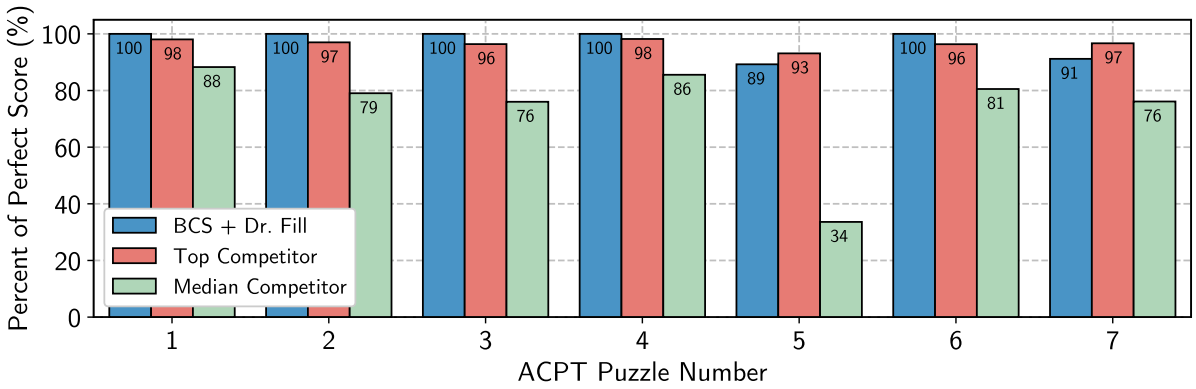


Figure 5: *A breakdown of our 2021 ACPT performance.* The 2021 ACPT consisted of 7 puzzles, for which our combined system achieves a perfect score and surpasses the top human competitor on 5 out of the 7 puzzles. We include the median competitor's performance to illustrate the difficulty of the puzzles.

errors that cannot be fixed by local search, i.e., there are several connected errors.

Overall, the largest source of remaining puzzle failures is special themed puzzles, which is unsurprising as our solver does not explicitly handle themes. The remaining errors are mostly split between proposal and scoring errors. Finally, connected errors typically arise when BP fills in an answer that is in our bi-encoder's answer set but is incorrect, i.e., the first-pass model was overconfident.

8 Related Work

Past Crossword Solvers Prior to our work, the three most successful automated crossword solvers were Proverb, WebCrow (Ernandes et al., 2005), and Dr. Fill. Dr. Fill uses a relatively straightforward TFIDF-like search for question answering, but Proverb and WebCrow combine a number of bespoke modules for QA; WebCrow also relies on a search engine to integrate external knowledge. On the solving side, Proverb and WebCrow both use loopy belief propagation, combined with A* search for inference. Meanwhile, Dr. Fill, uses a modified depth-first search known as limited discrepancy search, as well as a post-hoc local search with heuristics to score alternate puzzles.

Standalone QA Models for Crosswords Past work also evaluated QA techniques using crossword question-answer pairs. These include linear models (Barlacchi et al., 2014), WordNet suggestions (Thomas and S., 2019), and shallow neural networks (Severyn et al., 2015; Hill et al., 2016); we instead use state-of-the-art transformer models.

Ambiguous QA Solving crossword puzzles requires answering ambiguous and underspecified

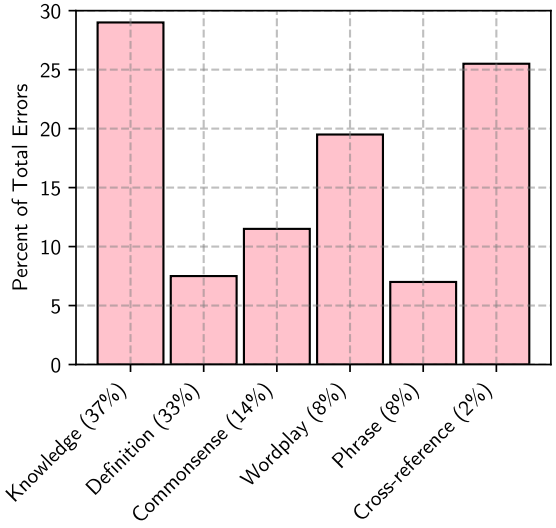


Figure 6: We manually categorize our QA failures using the categories from Table 1. The rate at which each category occurs in random examples is shown in parentheses. A disproportionate fraction of QA errors are due to cross-reference and wordplay clues.

clues while maintaining accurate estimates of model uncertainty. Other QA tasks share similar challenges (Ferrucci et al., 2010; Rodriguez et al., 2021; Rajpurkar et al., 2018; Min et al., 2020). Crossword puzzles pose a novel challenge as they contain unique types of reasoning and linguistic phenomena such as wordplay.

Crossword Themes We have largely ignored the presence of themes in crossword puzzles. Themes range from simple topical similarities between answers to puzzles that must be filled in a circular pattern to be correct. While Dr. Fill (Ginsberg, 2011) has a variety of theme handling modules built into it, integrating themes into our probabilis-

tic formulation remains as future work.

Cryptic Crosswords We solve American-style crosswords that differ from British-style "cryptic" crosswords (Efrat et al., 2021; Rozner et al., 2021). Cryptic crosswords involve a different set of conventions and challenges, e.g., more metalinguistic reasoning clues such as anagrams, and likely require different methods from those we propose.

9 Conclusion

We have presented new methods for crossword solving based on neural question answering, structured decoding, and local search. Our system outperforms even the best human solvers and can solve puzzles from a wide range of domains with perfect accuracy. Despite this progress, some challenges remain in crossword solving, especially on the QA side, and we hope to spur future research in this direction by releasing a large dataset of question-answer pairs. In future work, we hope to design new ways of evaluating automated crossword solvers, including testing on puzzles that are designed to be difficult for computers and tasking models with puzzle generation.

Ethical Considerations

Our data comes primarily from crosswords published in established American newspapers and journals, where a lack of diversity among puzzle constructors and editors may influence the types of clues that appear. For example, only 21% of crosswords published in *The New York Times* have at least one woman constructor (Chen, 2021) and a crossword from January 2019 was criticized for including a racial slur as an answer (Graham, 2019). We view the potential for real-world harm as limited since automated crossword solvers are unlikely to be deployed widely in the real world and have limited potential for dual use. However, we note that these considerations may be important to researchers using our data for question answering research more broadly.

Acknowledgements

We thank Sewon Min, Sameer Singh, Shi Feng, Nikhil Kandpal, Michael Littman, and the members of the Berkeley NLP Group for their valuable feedback. We are also grateful to Will Shortz and the organizers of the American Crossword Puzzle Tournament for allowing us to compete in the event. This work was funded in part by the DARPA XAI and LwLL programs. Nicholas Tomlin is supported by the National Science Foundation Graduate Research Fellowship.

References

Gianni Barlacchi, Massimo Nicosia, and Alessandro Moschitti. 2014. Learning to rank answer candidates for automatic resolution of crossword puzzles. In *CoNLL*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "Siamese" time delay neural network. In *NeurIPS*.

Jeff Chen. 2021. Women constructors in the Shortz era. *XWord Info*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Avia Efrat, Uri Shaham, Dan Kilman, and Omer Levy. 2021. Cryptonite: A cryptic crossword benchmark for extreme ambiguity in language. In *EMNLP*.

Marco Ernandes, Giovanni Angelini, and Marco Gori. 2005. WebCrow: a web-based system for crossword solving. In *AAAI*.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. 2010. Building Watson: An overview of the DeepQA project. *AI Magazine*.

Justin Fu and Sergey Levine. 2021. Offline model-based optimization via normalized maximum likelihood estimation. In *ICLR*.

Matthew L Ginsberg. 2011. Dr. Fill: Crosswords and an implemented solver for singly weighted CSPs. In *JAIR*.

Denise Grady. 2010. Across and down, the wizard who is fastest of all. *New York Times*.

Ruth Graham. 2019. The NYT crossword puzzle's use of an ethnic slur says a lot about the state of crossword puzzling. *Slate*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *NAACL*.

Felix Hill, Kyunghyun Cho, Anna Korhonen, and Yoshua Bengio. 2016. Learning to understand phrases by embedding the dictionary. In *TACL*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. In *IEEE Transactions on Big Data*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Michael L. Littman, Greg A. Keim, and Noam Shazeer. 2002. A probabilistic approach to solving crossword puzzles. In *Artificial Intelligence*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *EMNLP*.

Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. Compositional questions do not necessitate multi-hop reasoning. In *ACL*.

Judea Pearl. 1988. Probabilistic reasoning in intelligent systems.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *EMNLP*.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan Boyd-Graber. 2021. Quizbowl: The case for incremental question answering. In *JMLR*.

Josh Rozner, Christopher Potts, and Kyle Mahowald. 2021. Decrypting cryptic crosswords: Semantically complex wordplay puzzles as a target for NLP. In *NeurIPS*.

Aliaksei Severyn, Massimo Nicosia, Gianni Barlacchi, and Alessandro Moschitti. 2015. Distributional neural networks for automatic resolution of crossword puzzles. In *ACL*.

Felix Stahlberg and Bill Byrne. 2019. On NMT search errors and model errors: Cat got your tongue? In *EMNLP*.

Anu Thomas and Sangeetha S. 2019. Towards a semantic approach for candidate answer generation in solving crossword puzzles. In *CoCoNet*.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. In *TACL*.

A Details of Qualitative Analysis

In this section, we provide rough definitions for the categories used to construct Table 1 and conduct the manual QA error analysis in Figure 6:

Knowledge Clues that require knowledge of history, scientific terminology, pop culture, or other trivia topics. Answers to knowledge questions are frequently multi-word expressions or proper nouns that may fall outside of our closed-book answer set, and clues often involve additional relational reasoning, e.g., *Book after Song of Solomon* (ISAIAH).

Definition Clues that are either rough definitions or synonyms of the answer.

Commonsense Clues that rely on relational reasoning about well-known entities. These clues often involve subset-superset, part-whole, or cause-effect relations, e.g., *Cause of a smudge* (WETINK).

Wordplay Clues that involve reasoning about heteronyms, puns, anagrams, or other metalinguistic patterns. Such clues are usually (but not always) indicated by a question mark.

Phrase Clues or answers that involve common phrases or multi-word expressions. These clues are often written with quotation marks or blanks and their answers are frequently synonymous expressions, e.g., *Hey man!* (YODUDE).

Cross-Reference Clues that require knowledge of other elements in the puzzle, either through explicit reference (e.g., *See 53-Down*) or due to their usage of crossword themes.

B Additional Dataset Statistics

Figures 7–9 present a breakdown of the publishers, years, and answer lengths that are present in our crossword dataset.

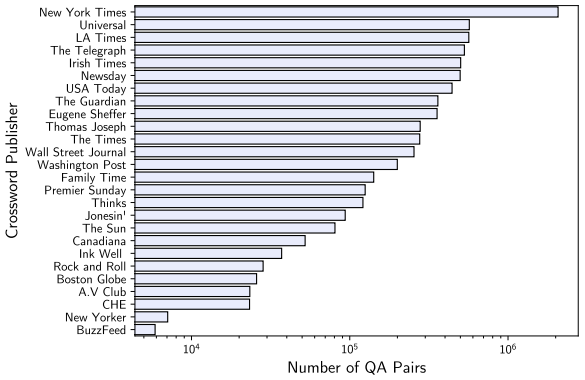


Figure 7: We build our dataset by collecting data from 26 publishers. Using a diverse set of publishers is beneficial as each publisher has different question types, answer distributions, and puzzle idiosyncrasies.

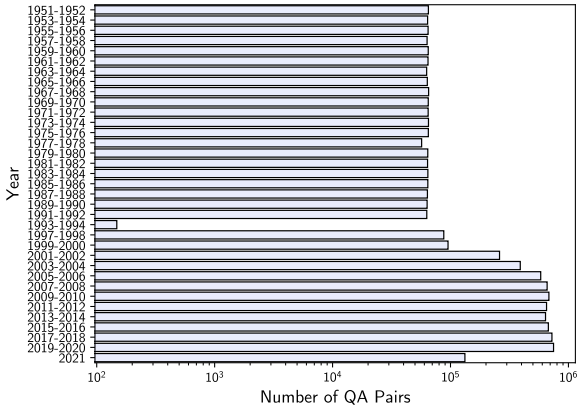


Figure 8: Our dataset spans over 70 years of crossword puzzles. The dip in puzzles in 1993-1996 is due to an unavailability of NYT puzzles from those years.

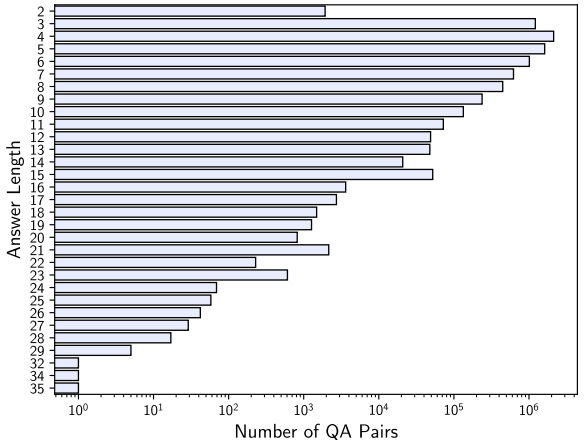


Figure 9: The answers in our dataset span many different lengths; longer answers are typically more difficult multi-word expressions or theme answers.

C American Crossword Puzzle Tournament Details

System	Year	Score	Rank
Proverb	1998	6,215	213th
Dr. Fill	2012	10,060	141st
Dr. Fill	2013	10,550	92nd
Dr. Fill	2014	10,790	67th
Dr. Fill	2015	10,920	55th
Dr. Fill	2016	11,205	41st
Dr. Fill	2017	11,795	11th
Dr. Fill	2018	10,740	78th
Dr. Fill	2019	11,795	14th
BCS QA + Dr. Fill	2021	12,825	1st
BCS QA + BP + LS	2021	13,065	1st

Table 5: Performance over the years in the American Crossword Puzzle Tournament. Dr. Fill has steadily improved due to system changes and increased training data. We also provide a retrospective evaluation of our final system (bottom row). Note that the 2020 ACPT was cancelled due to COVID-19.

Scoring System The main portion of the American Crossword Puzzle Tournament consists of seven crossword puzzles. Competitors are scored based on their accuracy and speed. For each puzzle, the judges award:

- 10 points for each correct word in the grid,
- 150 bonus points if the puzzle is solved perfectly,
- 25 bonus points for each full minute of time remaining when the puzzle is completed. This bonus is reduced by 25 points for each incorrect letter but can never be negative.

The total score for the seven puzzles determines the final results, aside from a special playoff for the top three human competitors. Table 5 shows scores over the years for the American Crossword Puzzle Tournament, including our 2021 submission.

D Additional Analysis Results

Figure 10 shows our accuracy broken down by day of the week. Monday and Tuesday NYT puzzles—ones designed to be easier for humans—are also easy for computer systems. On the other hand, Thursday NYT puzzles, which often contain unusual theme entries such as placing multiple letters into a single grid, are the most difficult. Our system is unaware of these special themes, but the

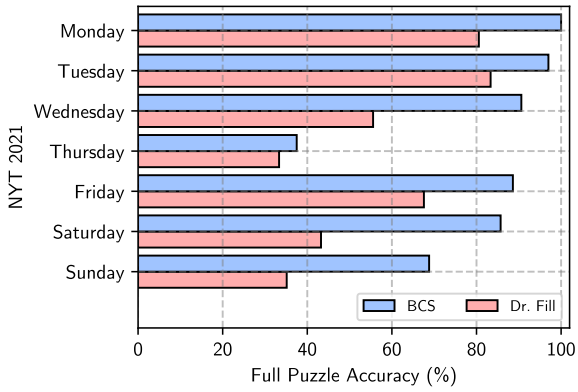


Figure 10: We compare our system's accuracy on NYT puzzles to the previous state-of-the-art Dr. Fill system and break down the results by day of the week. Both systems succeed on early week puzzles but struggle on Thursday puzzles that often contain unusual themes.

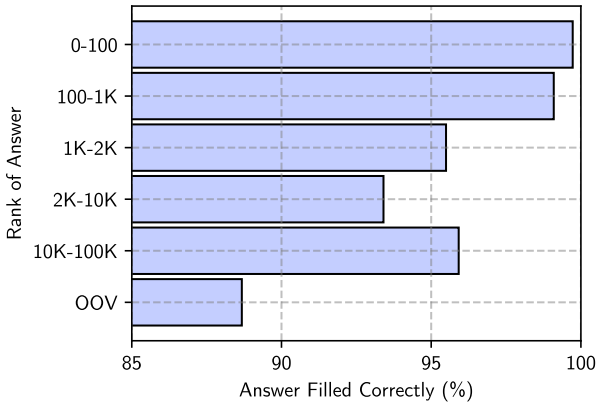
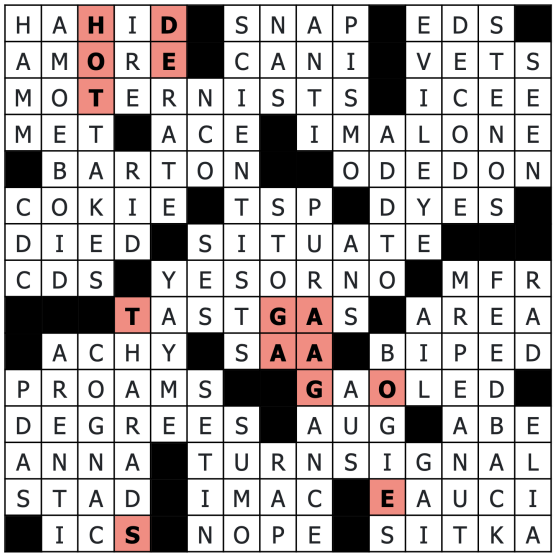


Figure 11: The chance that the BCS correctly fills in an answer as a function of the rank of the answer under its QA model. If the QA model predicts the answer in its top 1,000 candidates, it is usually filled in correctly.

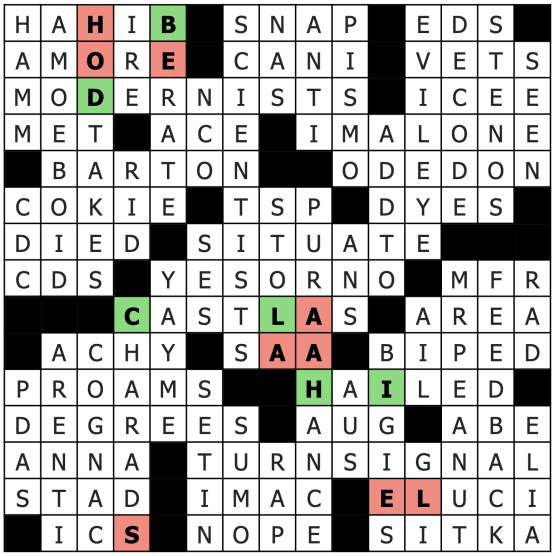
Dr. Fill system includes various methods to detect and resolve them and is thus more competitive on Thursday NYT puzzles. Finally, our system provides the largest gains on Saturday and Sunday NYT puzzles which often contains the hardest clues from a QA perspective.

We also compute results on themeless NYT puzzles. Themed puzzles range from topical similarity between answers in a puzzle, to multiple words ending with the same suffix, to multiple letters fitting inside a single square (i.e., rebus puzzles). For evaluation purposes, we consider themed puzzles to be any puzzle that contains a rebus[8] or a circled letter[9] according to XWord Info, but this does not capture all possible themes.

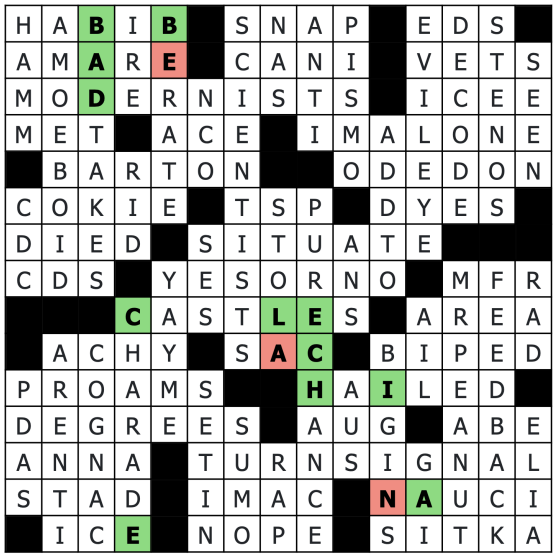
[8] https://www.xwordinfo.com/rebus
[9] https://www.xwordinfo.com/circles



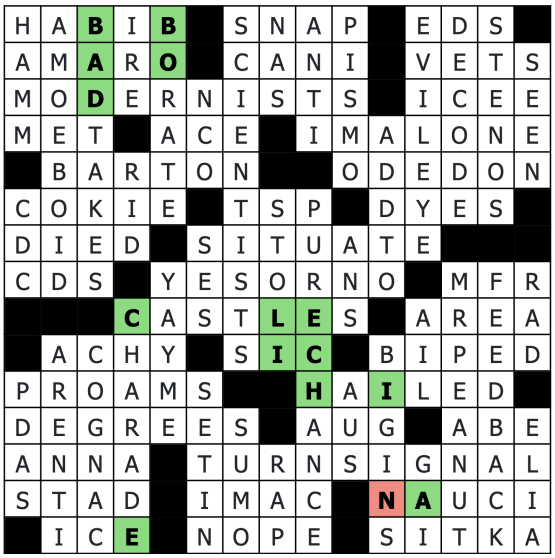
(a) Before Local Search



(b) Step #1



(c) Step #2



(d) Step #3

Clue	Gold	Before	Step 1	Step 2	Step 3
Beloved, in Arabic	HABIB	HADID	HAHIB	HABIB	HABIB
Ill-advised opinions	BADTAKES	HOTTAKES	HODTAKES	BADTAKES	BADTAKES
Feeling on a lo-o-ong car trip	BORATE	DERATE	BERATE	BERATE	BORATE
Italian herbal liqueur	AMARO	AMORE	AMORE	AMARE	AMARO
Not radical	MODERNISTS	MOTERNISTS	MODERNISTS	MODERNISTS	MODERNISTS
Long fur scarfs	STOLI	STOGA	STOLA	STOLA	STOLI
Outcome of a coin flip, e.g.,	PURECHANCE	PURAAGANCE	PURAAHANCE	PURECHANCE	PURECHANCE
Choose randomly, in a way	CASTLES	TASTGAS	CASTLAS	CASTLES	CASTLES
Like toreadors, again and again	CHARADE	THARADS	CHARADS	CHARADE	CHARADE
“Get ’em!”	SIC	SAA	SAA	SAC	SIC
Worrisome uncertainties	BIGIFS	BOGIES	BIGIES	BIGINS	BIGINS
Like taxis and Julius Caesar, once	HAILED	GAOLED	HAILED	HAILED	HAILED
Immunologist Anthony	FAUCI	EUACI	ELUCI	NAUCI	NAUCI
Suffix with coward	ICE	ICS	ICS	ICE	ICE

Figure 12: Top: We show a larger version of Figure 4. Bottom: The clues and associated answers after each step.