Adapting the Contrast-Consistent Search Method to Multiclass Classification

Stanford CS224N Custom Project

Diego Zancaneli
Department of Computer Science
Stanford University
diegozan@stanford.edu

Santiago Hernández
Department of Computer Science
Stanford University
imsanti@stanford.edu

Tomás Pfeffer
Department of Computer Science
Stanford University
tomasp@stanford.edu

Abstract

Burns et al. (2022) [1] propose a new method for discovering latent knowledge within the internal activations of a language model ("LM") in an unsupervised manner. They propose the Contrast-Consistent Search ("CCS") method to accurately answer "yes-no" questions based on unlabeled model activations, leveraging logical consistency properties as a "constraint" in their optimization setup. Given the promising initial results described on the paper and our motivation to explore the epistemological properties of LMs, we built upon their work and extended the CCS method to multi-class classification (as opposed to binary). After exploring various combinations of models, datasets, loss functions, and hyper-parameters through multiple iterations, we were able to achieve a noteworthy accuracy of 0.436 in the QA multiple choice task by leveraging the RACE dataset.

1 Key Information to include

- Mentor: Irena Gao
- Sharing project: N/A

2 Introduction

LMs often generate inaccurate text due to their inability to effectively represent the concept of truth. While these models are trained to internalize this essential attribute, they can still produce erroneous output when the training objective differs from the actual task requirements. This issue is not unique to any particular model but rather a function of the training objective, making it increasingly challenging to mitigate with human supervision, especially in complex domains. Simply increasing the size of the models is unlikely to address this misalignment (Evans et al., 2021 [2]; Shuster et al., 2021 [3]).

The development of explainable AI is essential to ensure that these systems reveal their 'thinking' completely and faithfully, which is critical to gaining trust and adoption. Along these lines, self-knowledge is necessary for AI to make accurate predictions about their own behavior and reasoning, with "awareness" about their core knowledge areas. Coupled with that, truthfulness is a crucial factor that determines whether AI can provide factually accurate information, including finding, using, and evaluating source materials correctly (Kadavath et. al (2022) [4]). Together, these factors play an essential role in the mitigation of hallucinations and the development of more robust and reliable AI models that can operate effectively and ethically in real-world scenarios.

While most current methods for ensuring model accuracy rely on human supervision to define what constitutes correctness, this approach may not always be feasible or desirable in certain settings. However, exciting recent research (detailed in the following section) suggests that it may be possible to achieve truthful modeling without relying on an external source of ground truth or any form of supervision.

The continued advancement of this question could open up new avenues for developing more robust and reliable models in scenarios where human oversight is limited or impractical. The ultimate goal of the work we are expanding is to develop models that are truthful and capable of accurately assessing their own level of confidence in their knowledge and reasoning. This requires the ability of AI systems to recognize their own limitations and areas of uncertainty as a fundamental prerequisite.

3 Related Work

In *Discovering Latent Knowledge in Language Models without Supervision*, Burns et al. (2022) [1] introduce a novel approach for unsupervised discovery of latent knowledge within a language model's internal activation. Their proposed method, CCS, utilizes logical consistency properties as a constraint in the optimization setup to enable accurate answering of "yes-no" questions based on unlabeled model activations.

With a related motivation, Kadavath et. al (2022) [4] explore whether language models can assess the accuracy of their responses and predict which questions they can answer correctly. The study finds that larger language models are well-calibrated on various types of multiple-choice and true/false questions, and the models can predict the probability of their answers being correct. The paper proposes that these observations could help create more honest models in the future.

Despite the predominant focus on improving AI models through human supervision, there has been notable research in exploring the potential of AI systems to move beyond direct supervision. For instance, Christiano et al. (2018) propose Iterated Amplification as a training strategy that uses no external reward function and progressively builds up a training signal for difficult problems by combining solutions to easier sub-problems [5]. Perez et al. (2022) suggest a new approach to identify harmful behaviors in LMs by "red teaming" (i.e., generating test cases) using another LM, flagging a wide range of diverse undesirable LM behaviors that can be fixed before deployment [6]. Although this approach could potentially broaden the scope of supervisory applications, most of these proposals have yet to move beyond the theoretical stage, and it remains unclear how well these methods can generalize. Differently, CSS addresses this issue as an empirical problem that can be tackled with current models, as opposed to a theoretical evaluation.

Our work builds upon the CCS method described in the original reference paper, which entails training a linear projection of hidden states that remains consistent across negations. However, we take this approach a step further by extending it from binary to multiclass classification. We leverage the insight that truth representations can be significant in models and can be extracted by identifying the principal component of a modified representation space. Our method helps to advance the research on the recovery of knowledge from model representations, thereby improving the interpretability of complex models.

4 Approach

Our objective is to use a pre-trained neural language model and the CCS method to predict the correct answer option for a given set $q_1, ..., q_n$ of questions with multiple answer options but with only one that is correct. Like in the original paper, q_i needs to be a question (procedural or factual) with a well-defined answer. We will be using the model's hidden representations $\phi(x) \in \mathbb{R}^d$ on a language input x. The task continues to be answering questions $q_1, ..., q_n$ by training a model that has $\phi(x)$ as its only input.

4.1 Adapted CCS Method

In our reference paper, the original CCS method leverages the idea that truth has structure that can satisfy consistency properties in a way that few other features in a LM are likely to satisfy. Extending this idea to the multiple choice answer domain, for some question q_i with j discrete answer choices such that p_i^k is the probability of k being the correct answer, we have:

$$\sum_{k=1}^{j} p_i^k = 1$$

Rather than building contrast pairs for each question q_i, we construct contrast groups, with j possible labels, x_i^k, each corresponding to a natural language statement. The discrete nature of these labels means that we can answer a given question q_i with all possible answers j, such that only one natural language statement is true (x_i^t), and $j-1$ statements x_i^k (for $k \neq t$) that are false.

After some iterations, we also tweaked the original construction of the labels to make it more explicit that the end of the string refers to the alternative chosen to be the "correct" one in each case. As we append the identifier of the answer only (i.e., "A", ..., "D" or "0", ..., "3"), including the excerpt `"The correct answer for the question is option"` makes it clearer that the last character of the label has a meaning and is not just a random character.

Implementation following original CSS label construction:

```
Read the article and select the best answer. nArticle: {{article}}
nQuestion: {{question}} nOptions: {{"A"}}: {{options.0}} \n
{{"B"}}: {{options.1}} \n {{"C"}}: {{options.2}} \n{{"D"}}:
{{options.3}} {{answer}}"
```

Our implementation:

```
Read the article and select the best answer. nArticle: {{article}}
nQuestion: {{question}} nOptions: {{"A"}}: {{options.0}} \n
{{"B"}}: {{options.1}} \n {{"C"}}: {{options.2}} \n{{"D"}}:
{{options.3}} \The correct answer for the question is option
"{{answer}}"
```

For each contrast group $\{x_i^1, ..., x_i^j\}$, we construct normalized representations $\{\phi(x_i^1), ..., \phi(x_i^j)\}$ with feature extractor $\phi(\cdot)$. Like in the original CCS method, we normalize each representation $\phi(x_i^k)$ independently so that the manipulation done (i.e., identifying the answer $\{1, ..., j\}$ associated to each x_i^k):

$$\tilde{\phi}(x_i^k) := \frac{\phi(x_i^k) - \mu^k}{\sigma^k}$$

where $\{(\mu^1, \sigma^1), ..., (\mu^j, \sigma^j)\}$ are the means and standard deviations of all $\tilde{\phi}(x_i^k)$ for $i \in \{1, ..., n\}$.

Then, we learn a probe $p_{\theta,b}(\tilde{\phi}) = \sigma(\theta^T \tilde{\phi} + b)$ to represent the probability that the statement x with normalized hidden state $\tilde{\phi}(x)$ is true. We kept the sigmoid implementation used in the original paper.

To define our training objective, we draw inspiration from the original paper and first come up with a statement that, by the consistency property described before, recognizes that the probabilities need to sum to 1.

$$L_{consistency}(\theta, b, q_i) := \Big[1 - \Big(p_{\theta,b}(x_i^1) + p_{\theta,b}(x_i^2) + ... + p_{\theta,b}(x_i^j)\Big)\Big]^2$$

We also define a confidence loss to prevent degenerate solutions where $p_{\theta,b}(x_i^k)$ is the same for all $k \in \{1, ..., j\}$ (i.e., $p_{\theta,b}(x_i^k) = \frac{1}{j}$) and to minimize entropy across the probabilities.

$$L_{confidence}(\theta, b, q_i) := \Big\{ - \Big(p_{\theta,b}(x_i^1) \cdot log(p_{\theta,b}(x_i^1)) + ... + p_{\theta,b}(x_i^j) \cdot log(p_{\theta,b}(x_i^j))\Big)\Big\}^2$$

Hence, the total unsupervised loss is the sum of the consistency and confidence losses, averaged across all contrast pairs:

$$L_{CSS-MC}(\theta, b) := \frac{1}{n} \sum_{i=1}^{n} L_{consistency}(\theta, b, q_i) + L_{confidence}(\theta, b, q_i)$$

Finally, to infer what is the right answer to the question, we output $k \in \{1, ..., j\}$ where:

$$p_{\theta,b}(x_i^k) = max\Big(p_{\theta,b}(x_i^1) + ... + p_{\theta,b}(x_i^j)\Big)$$

5 Experiments

5.1 Data

We selected datasets with multi-choice question answering with availability from Hugging Face and a significant number of academic references. The main tasks to be tested in these datasets are reading comprehension and commonsense inference. It is worth noting that we don't have any overlap in datasets used with the original CCS paper because of our constraint of multiclass classification (through multiple choice datasets), in contrast to binary classification.

- **RACE** (Lai et al., 2017): a large-scale reading comprehension dataset with more than 28,000 passages and nearly 100,000 questions. The dataset is collected from English examinations in China, which are designed for middle school and high school students. [7]
- **ARC** (Clark et al., 2018): the ARC dataset (the AI2 reasoning challenge) is a comprehensive collection of 7,787 natural science questions that were designed for use on standardized tests. These questions encompass a wide range of linguistic and inferential phenomena and have questions varying in level of difficulty. [8]
- **SWAG** (Zellers et al., 2018): a large-scale adversarial dataset with 113,000 multiple choice questions about a rich spectrum of grounded situations. [9]
- **Cosmos QA** (Huang et al., 2019): a large-scale dataset of 35,600 problems that require commonsense-based reading comprehension, formulated as multiple-choice questions. In contrast to many reading comprehension datasets where the questions focus on factual and literal understanding of the context paragraph, this dataset focuses on reading between the lines over a diverse collection of people's everyday narratives. [10]

5.2 Evaluation method

Similar to the original CCS, we will evaluate our new method with accuracy, i.e. the percentage of questions that we are able to classify correctly from the test datasets.

When testing CCS, we conduct 10 optimization runs ("tries") using AdamW (Loshchilov & Hutter, 2017) [11] with a learning rate of 0.01, and choose the run with the lowest unsupervised loss. Like the original implementation, we train CCS on all prompts as a single training set and then test it on the corresponding test split. Differently from the original paper, we chose a 75% / 25% train/test split.

Inspired by the reference paper, we also run logistic regression (LR) on the training split for each dataset using $\tilde{\phi}(x_i^k)$ for $i \in \{1, ..., n\}$ as the covariates, evaluating on the corresponding test split.

5.3 Experimental details

To optimize multiclass CCS performance, we ran several experiments changing one variable at a time to understand the model's accuracy change under each test. We focused our analysis on four key attributes: (i) datasets, (ii) models we extract the hidden representation $\phi(x)$ from, (iii) CCS loss function, and (iv) CCS hyper-parameters such as the number of examples used to train CCS and the number of tries.

Initially, we weren't getting great results but as we tweaked the experimental setup we increased significantly (by $\sim 10p.p.$).

5.4 Results

We started by running the same model (gpt2) will all the datasets so we could identify which datasets seemed more promising for us to conduct more thorough experimentation. In all tables below, we ran the experiments with $1,000$ examples, 10 tries, and a 0.01 learning rate, unless stated otherwise. All numerical values presented in this section represent the mean of the outputs obtained from running each experiment twice.

Table 1. GPT-2 with all datasets

Model	gpt2	gpt2	gpt2	gpt2	Mean
Data	race	ai2_arc	swag	cosmos_qa	Mean
Multiclass CCS	0.24	**0.256**	0.196	0.24	0.233
LR	0.252	0.276	0.296	0.248	0.268

After determining that the race (random accuracy but inference-based task) and ai2_arc datasets had the most potential, we proceeded to test them using various models. However, we faced some restrictions with the AWS setup, which prevented us from running CCS on some of the models used in the original paper, such as gpt-j, T0pp, unifiedqa, T5, and deberta. Instead, we conducted experiments using deberta-mnli and roberta-mnli, which were also featured in the original paper, as well as smaller versions of the other models.

Table 2. Testing different models with race dataset

Model	gpt2	gpt2-l	roberta-mnli	distilbert	deberta-mnli	unifiedqa-t5-sm	Mean
Data	race	race	race	race	race	race	
Multiclass CCS	0.24	0.228	0.264	0.296	**0.32**	0.204	0.259
LR	0.252	0.248	0.332	0.264	0.436	0.244	0.296

Table 3. Testing different models with ai2_arc dataset

Model	gpt2	gpt2-l	roberta-mnli	distilbert	deberta-mnli	unifiedqa-t5-sm	Mean
Data	ai2_arc	ai2_arc	ai2_arc	ai2_arc	ai2_arc	ai2_arc	
Multiclass CCS	0.256	0.252	0.256	**0.272**	0.24	0.252	0.255
LR	0.276	0.244	0.256	0.28	0.436	0.28	0.295

Afterwards, we proceeded to test the most favorable model-dataset combinations (race with roberta-mnli, distilbert, and deberta-mnli, and ai2_arc with deberta-mnli and distilbert) with a larger set of examples $(5,000)$. We report the findings of these tests below.

Table 4. Testing promising pairs with 5,000 examples

Model	roberta-mnli	distilbert	deberta-mnli	roberta-mnli	distilbert	Mean
Data	race	race	race	ai2_arc	ai2_arc	
Multiclass CCS	0.2264	0.2576	**0.276**	0.228	0.242	0.246
LR	0.2712	0.2416	0.512	0.278	0.25	0.311

Next, we wanted to verify whether using a model fine-tuned for the multiple-choice QA task could help us achieve higher accuracy. As all the datasets we have selected are complex/ nuanced and require specific-domain knowledge, we suspected that we were not getting maximum potential performance with the standard models.

Note that we returned to testing with 1,000 examples: unintuitively (although might have been by chance), we found that using a higher number of examples didn't make a significant difference in CCS (although it was helpful for LR), and using fewer examples enabled us to conduct more experiments. However, in the final experiment, we reverted back to using a larger set of examples, as it is inherently beneficial to increase accuracy.

Table 5. Testing models fine-tuned with the datasets that we are using

Model	distilbert-race	roberta-race	bert-swag	t5-cosmos	Mean
Data	race	race	swag	cosmos_qa	
Multiclass CCS	0.228	0.26	0.248	**0.292**	0.257
LR	0.32	0.42	0.244	0.276	0.315

Then, we identified the most promising dataset-model combinations from Tables 2, 3, and 5, and experimented with various versions of our loss function. Our objective was to determine the ideal balance between consistent and informative loss by adjusting their respective ratios.

Table 6. deberta-mnli (with race)

Consistency Loss Weight Informative Loss Weight	1.0 1.0	1.0 0.8	1.0 0.5	1.0 0.3	1.0 0.0	0.8 1.0	Mean
Multiclass CCS	0.32	0.288	0.332	**0.356**	0.204	0.244	0.291
LR	0.436	0.468	0.424	0.456	0.432	0.392	0.435

Table 7. t5-cosmos (with cosmos_qa)

Consistency Loss Weight Informative Loss Weight	1.0 1.0	1.0 0.8	1.0 0.5	1.0 0.3	1.0 0.0	0.8 1.0	Mean
Multiclass CCS	**0.292**	0.28	0.244	0.276	0.208	0.252	0.258
LR	0.276	0.216	0.264	0.276	0.256	0.268	0.259

Table 8. distilbert (with race)

Consistency Loss Weight Informative Loss Weight	1.0 1.0	1.0 0.8	1.0 0.5	1.0 0.3	1.0 0.0	0.8 1.0	Mean
Multiclass CCS	**0.296**	0.272	0.268	0.248	0.232	0.244	0.26
LR	0.264	0.24	0.3	0.26	0.3	0.228	0.265

Having identified a construction that exhibits significantly superior performance compared to the others (deberta-mnli with 1.0 consistency as opposed to 0.3 informative losses), we now aim to adjust the CCS architecture by modifying two crucial hyper-parameters: the number of attempts and the learning rate.

Table 9. Changing number of tries in CCS

Number of tries	1	5	10	15	25
Multiclass CCS	0.316	0.346	0.356	0.264	**0.362**
LR	0.496	0.48	0.456	0.444	0.436

Having found a superior performance for 25 tries, we fix this hyper-parameter and now evaluate accuracy under different learning rates.

Table 10. Learning rate

Number of tries	10^{-4}	10^{-3}	10^{-2}
Multiclass CCS	0.284	**0.362**	0.284
LR	0.488	0.436	0.452

Finally, we have decided to explore a new idea for the model's architecture. Instead of using the Multiclass CCS suggested in the *Approach* section, we aim to determine how accuracy is affected when we run the regular binary CCS four times for a given question q_i, each time selecting a fictitious alternative as the correct answer, producing a fictitious prompt q_i^f with $f \in \{0, ..., 3\}$.

This strategy, which we call "Multiple-Fold Cross-Validation CCS", involves adding a fictitious answer after the question and the possible answers to create a new prompt structure that looks something like: `(question) (possible answers) (fictitious answer)`. For instance, let's assume we have the following prompt: "What is the capital of France?" with possible answers (A) Paris, (B) London, (C) Rome, and (D) Madrid. To implement our proposed architecture, we create each fictitious prompt q_i^f by building a contrast pair from each alternative:

- `What is the capital of France? A Paris B London C Rome D Madrid. The correct answer is "A"` (which is a natural language statement we denote x_i^+)
- `What is the capital of France? A Paris B London C Rome D Madrid. The correct answer is not "A"` (which we likewise denote x_i^-)

Like in the original paper, we learn to classify x_i^+ and x_i^- as true or false following their original consistency and informative losses, finding $p(x_i^+)$ and $1 - p(x_i^-)$ that represent the probability that the answer to q_i^f is "Yes". We average these values to find the final prediction $\tilde{p}(q_i)$:

$$\tilde{p}(q_i^f) = \frac{1}{2}\Big(p(x_i^+) + 1 - p(x_i^-)\Big)$$

Then, we output the f that corresponds to the highest $\tilde{p}(q_i^f)$. Intuitively, this means that after the individual CCS assessment of each alternative, we choose the correct one based on CCS degree of confidence. The Multiple-fold Cross-Validation CCS exhibits a consistent performance with the regular Multiclass CCS, albeit slightly worse.

Table 11. Multiple-fold Cross-Validation CCS

	Accuracy
Multiple-fold CCS	0.296
Multiclass CCS	**0.362**
LR	0.706

6 Analysis

We find that our best performance approximately matches the Stanford AR [12] model performance on the RACE dataset (43.2 and 43.3, respectively), exceeds the Sliding Window [13] model performance, and nears Gated-Attention [14] performance (within one percentage point). The human performance on the dataset is 94.5%, indicating that the data is clean, but there is a significant gap between turkers' performance and human performance.

We have included the performance achieved by BERT models on the RACE dataset [7] reference, even though a direct comparison may not be entirely fair. Nonetheless, it serves as a good benchmark to keep in mind.

Table 12. Comparison Benchmarks

	RACE
Multiclass CCS Best Performance	43.2
Sliding Window	32.2
Stanford AR	43.3
Gated-Attention	44.1
Turkers	73.3
Human Ceiling Performance	94.5
BERT	72.0
RoBERTa	83.2
DeBERTa	86.8

In hindsight, we acknowledged that selecting the RACE dataset for machine comprehension was more challenging than anticipated, particularly considering that we would be working with smaller scale models. We could have chosen alternative datasets such as CNN/Daily Mail, Children's Book Test, and Who-Did-What, as demonstrated by the lower accuracy achieved by models such as Stanford AR and Gated AR on RACE, but these datasets require more complex data structures and model architectures (because of the variable number of possible answers depending on the question, for example).

Furthermore, we observed significant variability in the results of running the same experiment multiple times, which can be reduced by running each experiment multiple times†or by increasing the number of examples used in training (the variance in results is an indicative of the different levels of difficulty amongst questions; note that most experiments used $1,000$ examples out of the $100,000$ in the dataset). However, we should also focus on improving our model architecture to minimize this variance in the first place.

†We ran each experiment twice due to time constraints and included specific insights about each experiment directly in the Results section under Experiments

7 Conclusion

We generalized the CCS methodology beyond the yes-no question-answering setting, building a promising tool to improve multi-class classification question-answering while at the same time studying potential more ambitious applications to the original CCS. Our approach contributes to the progress in recovering knowledge from model representations, which enhances the interpretability of complex models.

As highlighted by the original paper, CCS depends on the availability of a specific orientation in the activation space, which can effectively and consistently differentiate between accurate and inaccurate inputs. However, this relies on the ability of the model used to get the hidden states to assess the accuracy of an input. This means that CCS depends on the probes having directionally correct labels to achieve high precision. The datasets we chose are relatively difficult for the smaller models we used and don't seem to meet these circumstances completely.

In addition to replicating the work with a more robust infrastructure, potential avenues for future work include improving the model reliability and experimenting with different prompt architectures. We are also excited about the potential application of the method to open-ended questions.

References

[1] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *ArXiV*, 2022.

[2] Owain Evans, Owen Cotton-Barratt, Lukas Finnveden, Adam Bales, Avital Balwit, Peter Wills, Luca Righetti, and William Saunders. Truthful ai: Developing and governing ai that does not lie, 2021.

[3] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation, 2021.

[4] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. 2022.

[5] Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts, 2018.

[6] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models, 2022.

[7] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. RACE: large-scale reading comprehension dataset from examinations. *CoRR*, abs/1704.04683, 2017.

[8] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[9] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. *CoRR*, abs/1808.05326, 2018.

[10] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos QA: machine reading comprehension with contextual commonsense reasoning. *CoRR*, abs/1909.00277, 2019.

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017.

[12] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. *CoRR*, abs/1606.02858, 2016.

[13] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[14] Bhuwan Dhingra, Hanxiao Liu, William W. Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. *CoRR*, abs/1606.01549, 2016.