Allen AI Challenge - System Description

MALL Lab, IISc, Bangalore, India

1 Introduction

Question Answering has been an interesting task in AI and Machine learning. A successful question answering system indicates the ability of machine to understand underlying concepts. Allen AI conducted a challenge spanning over 4 months time. The challenge consisted of objective type science subject questions for 8th grade students.

Members of the MALL Lab (`http://malllabiisc.github.io/`) at the Indian Institute of Science (IISc), Bangalore, headed by Partha Talukdar (`http://talukdar.net/`), participated in the challenge with team name textbfwhatsinateamname. Our research is in Machine Reading which is very relevant to the challenge and hence we got interested.

A **Working demo** of our system is available @ `http://mall-lab.serc.iisc.in/ensemble/`.

2 Approach

The challenge had questions of varied difficulty levels. Some questions were straightforward fact based questions, while some needed natural language understanding and reasoning capabilities. We approached the solution using ensemble of various models. Models complemented each other and were designed to perform well on different kind of questions. Our training approach was a mix of supervised and un-supervised methods. The ensemble and representation learning approaches (M4) were supervised, while the IR (M1) and PMI (M3) approaches were unsupervised.We used CK12 and other similar textbooks, multiple existing knowledge graphs (e.g., conceptnet, wordnet, wikidata, etc), Wikipedia science articles, Science related web corpus. We think sentence-level micro-indexing is one of the novel aspects of our approach. Overall, an ensemble involving multiple loosely correlated models seems interesting. Incorporation of a custom-built knowledge graph with millions of nodes and edges focused on the school-level science domain is also promising.

For the **ensemble**, we used a hierarchical approach to group our models and generate predictions.

We first pruned our models with very correlated (>90%) output predictions and bucketed the rest according to their underlying mechanism (e.g all IR/solr models were in one bucket). Then, a classifier was trained over each of those buckets to generate a unified

confidence score. This confidence score for each bucket was used in the next level as input features for a final classifier. We used logistic regression on question/option pairs at each level.

Each model gave an un-normalized score for the option candidates, the scales of which hindered the learning process. To counter this, we generated 3 normalized versions (sum normalization, exp normalization, and zero mean-unit variance normalization) of the score and used them in place of the raw score. Normalizing the scores from each model resulted in the largest boost. The hierarchical groupings were particularly helpful as passing all models in a one-level approach made optimizing parameters difficult and resulted in a dip in performance. Architecture of ensemble model is given in Figure 1.

3 Models

We used an ensemble method consisting of the following approaches:

– M1: Solr and Elasticsearch-based information retrieval

– M2: textual entailment model

– M3: a statistical association-based model (PMI)

– M4: a neural network-based representation learning approach over question and answer pairs

– M5: inference over a knowledge graph constructed out of existing knowledge graphs and by running OpenIE over large unstructured text data

– M6: retrieval over the constructed KG

– M7: matrix factorization for question type prediction

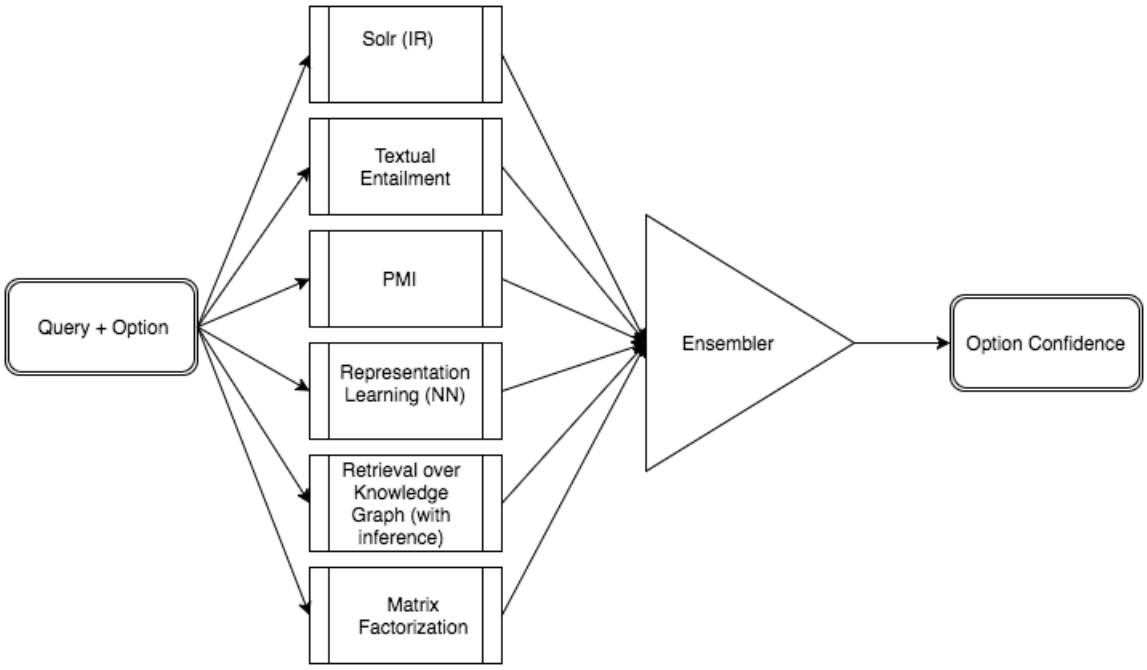


Figure 1: System Architecture

We used multiple variants of each of the above approaches. We also experimented with a variety of other techniques which ultimately didn't make it to the final model: learning a ranking function, multi-relational inference using subgraph features, natural logic inference, biased personalized pagerank, learned entailment model, tensor factorization for finer question analysis, etc. Detailed explanation of each model follows.

3.1 Solr and Elasticsearch-based information retrieval

We started with document level indexing. Later we observed that for a given question option pair if a valid supporting text is present in the corpus then in majority of the cases the supporting text consists of very few consecutive sentences. Keeping this observation in mind we shifted our attention from document level to sentence level indexing. We indexed a window of K sentences with a jump size of 1 sentence (meaning our window slides one sentence at a time), where value of K is in the set (1,2,3,4,5). This approach of indexing gave significantly better results comapared to document level indexing. We stemmed the corpus and removed stopwords (gave noticeable boost).

Instead of directly using max score from solr we used discounted score of top 10 valid supporting texts.

We built three models of solr:

- model 1: Uses window size of 1,2,3,4 and 5. Good for all types of questions.

- model 2: Uses window size of only 1 and 2. This model performs best on easy and medium difficulty questions. This model has the best overall accuracy.
- model 3: Uses window size of 1 and 2. This model considers only last sentence of a question. This model gives significant boost for questions where first part of the question is not relevant to the actual question.

We observed that using an ensemble of this 3 solr model gives better score compared to any individual model.

3.2 Textual Entailment

This unsupervised approach ranked answers based on how well they were entailed by the question+supporting_text. We used a textual entailment model provided by Excitement open platform[2] for the task. Among all the pre-trained models provided by the platform, MaxEntClassificationEDA_Base+OpenNLP_EN, performed the best. Question along with the supporting text (IR) was used to entail the answer. The answer which entailed with most confidence was selected as the right answer. This approach gave a total of 36 % accuracy on the train data of task. We tried various text and hypothesis pair for the entailment task. The pair which gave best accuracy consisted of "Question and Supporting text" as text and "Answer option" as the hypothesis pair.

3.3 Statistical Association-Based Model (PMI)

We built a PMI model by supervised learning on the entire corpus (book, wiki etc.). In test scenario, given a question and answer pair, pair-wise-word PMI score is aggregated. PMI score was then used as confidence on the answer option.

3.4 Neural Nets for Representation Learning

The idea behind this model was to learn a representation for the question and option pairs, and calculate a similarity metric between those representations to score each option. The features utilized were: - similarity between question and answer embedding (from rnnlm [4]) - pairwise average similarity between words in the question and answer text - similarity between the average word embeddings of the question/option text We used both euclidean and cosine distance as similarity metrics. These features were fed to a classifier that predicted the answer among the 4 options presented.

Generating the embeddings themselves was an unsupervised approach (we trained word2vec models [3], and recurrent neural network language models), and aggregating the various similarity metrics was supervised. We tried various models for the aggregation step including SVM-rank, logistic regression, and neural networks (NN). We settled on a NN for the learning phase.

To generate the question/option embeddings, we trained various embedding models over multiple corpora (CK12 textbooks, science articles from wikipedia, science articles from simple wiki, etc). The diversity in the content and style from each source generated significantly different word embeddings. (the nearest neighbors in each space were different– textbook data gave simple, topic specific nearest neighbors, whereas pretrained w2v was more general).

An important observation was that, the predictions of the embedding model were least correlated with all the other models.

3.5 Matrix factorization for question type prediction

We divided train dataset questions into various types. 6 such types were assumed (reasoning, direct, fill in the blanks kind etc.). We used a supervised approach to learn matrix factors to predict question type. These types were further used to customise weights on model prediction in the ensemble.

3.6 Background Knowledge/Reasoning

We built a knowledge graph with millions of nodes and edges focused on the school-level science domain by harvesting data from multiple existing knowledge graphs (e.g., conceptnet, wordnet, wikidata, etc), Wikipedia science articles, Science related web corpus. We built models to perform inference (e.g., biased personalized pagerank) and retrieval over this graph. We also experimented with a variety of other more sophisticated inference methods over this KG, although they didn't make it to the final submission.

4 Discussion

We profiled the accuracy of each of our models on held-out validation data from development dataset. Table 4 lists the accuracy of each model independent of the others. Ensemble of these models was then able to reach 52.7 % accuracy on the development set.

Model	**Accuracy on Validation data**
Solr_All_Models	49.3 %
Solr_All_Models (with atmost chuck size of 2 sentences)	49.9 %
Solr & Knowledge_base (triples)	48.7%
PMI	41.5 %
Textual Entailment	34 %
SVM (Embeddings)	38.3 %

Table 1: Accuracy of each model used in ensemble on validation data

4.1 Experiments which did not work - Natural Language Inference

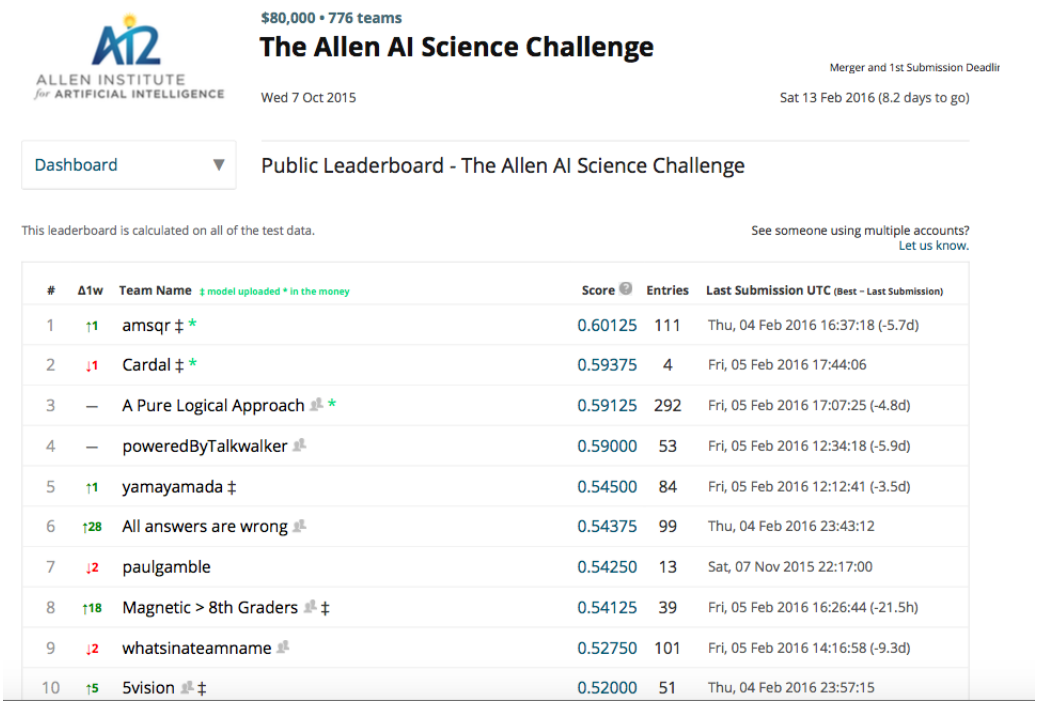
We also experimented with Natural Language Inference (NLI) [1]. NLI is a computational model for textual inference over natural language. Its a mid-way between the textual inference methods and first-order logic methods. Given a premise and a hypothesis, it finds the final entailment by composing the atomic entailments of edits over a low-cost edit sequence from premise to hypothesis. It combines the flexibility of textual entailment methods with the precision of first-order logic based methods.

Usage: In our data, we had questions with four options each. Initially, we tried to create four hypothesis' by pairing up the question with each of the four options. Then using it as search query, we extracted facts from our corpus using the information extraction engine(Solr). These facts were used as premises and then NLI system was used to find the entailment. The option with highest confidence was selected as answer. This method didn't work as NLI couldn't find any relation between the premises and hypothesis for almost all questions.

Since entailment over longer sentences would be difficult, we tried breaking the question into sentences and extract the main question sentence. The extracted question sentence is used as hypothesis, while rest of the sentences in the question are added as premises along with other facts retrieved the information extraction engine. It increased the coverage a bit but still NLI wasn't able to find answers for most of the questions.

While NLI was able to handle textual inferences well (eg FraCaS test suite), it was not very impressive in incorporating facts probably due to limited information about domain-specific terms and usage of words. Also, it became really slow when the edit sequence between premise and hypothesis was long and abstained in most of these cases.

5 Results



$80,000 • 776 teams

The Allen AI Science Challenge

ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

Wed 7 Oct 2015

Merger and 1st Submission Deadlir

Sat 13 Feb 2016 (8.2 days to go)

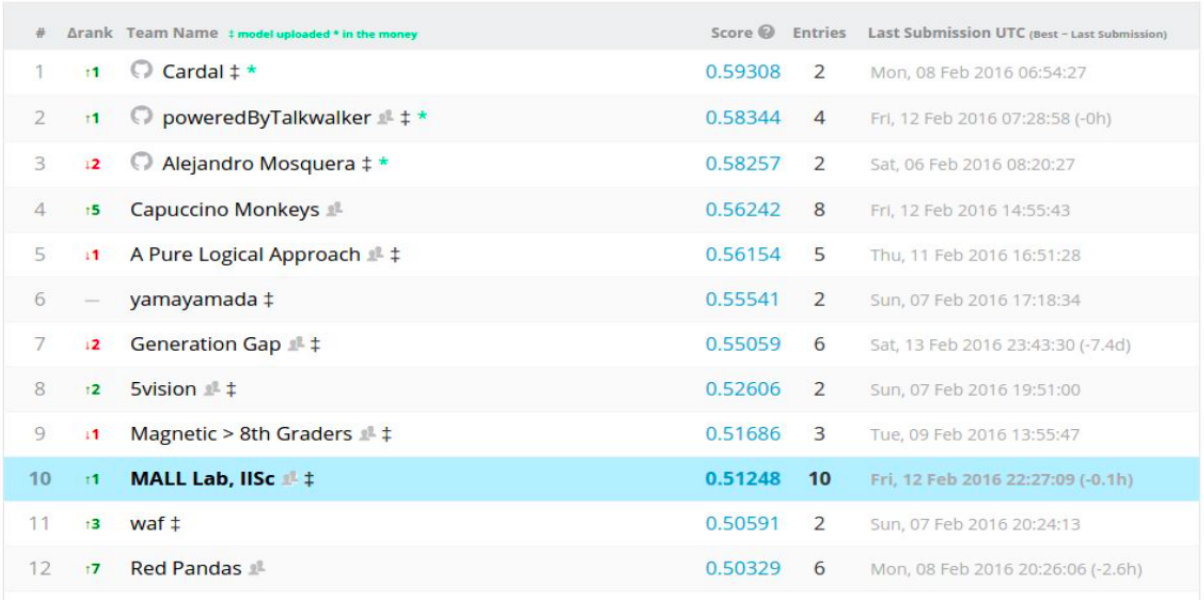
Dashboard ▼ Public Leaderboard - The Allen AI Science Challenge

This leaderboard is calculated on all of the test data.

See someone using multiple accounts? Let us know.

#	Δ1w	Team Name ‡ model uploaded * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑1	amsqr ‡ *	0.60125	111	Thu, 04 Feb 2016 16:37:18 (-5.7d)
2	↓1	Cardal ‡ *	0.59375	4	Fri, 05 Feb 2016 17:44:06
3	–	A Pure Logical Approach *	0.59125	292	Fri, 05 Feb 2016 17:07:25 (-4.8d)
4	–	poweredByTalkwalker	0.59000	53	Fri, 05 Feb 2016 12:34:18 (-5.9d)
5	↑1	yamayamada ‡	0.54500	84	Fri, 05 Feb 2016 12:12:41 (-3.5d)
6	↑28	All answers are wrong	0.54375	99	Thu, 04 Feb 2016 23:43:12
7	↓2	paulgamble	0.54250	13	Sat, 07 Nov 2015 22:17:00
8	↑18	Magnetic > 8th Graders ‡	0.54125	39	Fri, 05 Feb 2016 16:26:44 (-21.5h)
9	↓2	whatsinateamname	0.52750	101	Fri, 05 Feb 2016 14:16:58 (-9.3d)
10	↑5	5vision ‡	0.52000	51	Thu, 04 Feb 2016 23:57:15

Figure 2: Development set leader board position



#	Δrank	Team Name ‡ model uploaded * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	↑1	Cardal ‡ *	0.59308	2	Mon, 08 Feb 2016 06:54:27
2	↑1	poweredByTalkwalker ‡ *	0.58344	4	Fri, 12 Feb 2016 07:28:58 (-0h)
3	↓2	Alejandro Mosquera ‡ *	0.58257	2	Sat, 06 Feb 2016 08:20:27
4	↑5	Capuccino Monkeys	0.56242	8	Fri, 12 Feb 2016 14:55:43
5	↓1	A Pure Logical Approach ‡	0.56154	5	Thu, 11 Feb 2016 16:51:28
6	—	yamayamada ‡	0.55541	2	Sun, 07 Feb 2016 17:18:34
7	↓2	Generation Gap ‡	0.55059	6	Sat, 13 Feb 2016 23:43:30 (-7.4d)
8	↑2	5vision ‡	0.52606	2	Sun, 07 Feb 2016 19:51:00
9	↓1	Magnetic > 8th Graders ‡	0.51686	3	Tue, 09 Feb 2016 13:55:47
10	**↑1**	**MALL Lab, IISc ‡**	**0.51248**	**10**	**Fri, 12 Feb 2016 22:27:09 (-0.1h)**
11	↑3	waf ‡	0.50591	2	Sun, 07 Feb 2016 20:24:13
12	↑7	Red Pandas	0.50329	6	Mon, 08 Feb 2016 20:26:06 (-2.6h)

Figure 3: Final leader board Position

Real World Testing: We made our demo publicly available on open science day at IISc. Most of the participants were school students (better at answering these set of questions). Computer answered 483 questions correctly as opposed to 512 correctly answered by humans (end of the day statistics).

6 Future Directions

There were a number of threads which we couldn't integrate into our final model due to time constraint. It would be interesting to see the effect of KB inference, query optimization, more resources on the final performance of the system.

7 Conclusion

We developed a working system for AIlen AI Kaggle challenge. The system utilised an ensemble based system to answer questions. It gave reasonable performance with a final rank of 10 among 170 teams worldwide. A Working demo of our system is available @ `http://mall-lab.serc.iisc.in/ensemble/`.

References

[1] Bill MacCartney and Christopher D. Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, RTE '07, pages 193–200, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[2] Bernardo Magnini, Roberto Zanoli, Ido Dagan, Kathrin Eichler, Günter Neumann, Tae-Gil Noh, Sebastian Pado, Asher Stern, and Omer Levy. The excitement open platform for textual inferences. 2014.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[4] Anoop Deoras Lukar Burget Jan Honza Cernocky Tomas Mikolov, Stefan Kombrink. Rnnlm - recurrent neural network language modeling toolkit. IEEE Automatic Speech Recognition and Understanding Workshop, December 2011.