Building A Model Smarter Than An 8th Grader

Abhishek Naik, Mohan Bhambhani, Shiva Krishna M, and YSSV Sasi Kiran

Department of Computer Science and Engineering
Indian Institute of Technology Madras
{anaik, mohanpb, shiva, sasi}@cse.iitm.ac.in,

Abstract. One of the central problems in NLP and in AI in general is to understand the world around it and answer some basic questions about it. This involves collecting information, storing it in a proper format and learning from the data it sees. In this project we aim to implement this pipeline to a simpler yet challenging problem of understanding and answering primary school questions particularly those asked in standardized tests. This is broadly based on a problem posed by Allen Institute for Artificial Intelligence[1].

Keywords: Question Answering, Information Extraction

1 Introduction

Turing test[17] is the most widely accepted notion of measuring the intelligence of a machine. The test asks if the machine is indistinguishable from a human in a conversation with a set of judges. However in recent times many companies and labs have come up with chat bots that can easily trick humans into believing that they are talking to another human. However these models are not truly intelligent. To put it in the words of John Markoff [15] Turing test *is merely an indication of human gullibility*. Considering that there is no one way to measure human intelligence, he argues that there must not be a single test to judge machines either.

One way of measuring human intelligence is by standardized tests such as NTSE, SAT, GRE etc. In [3], Clarke et. al have proposed that such standardized tests act as a better way to test machine intelligence than the Turing test. Many questions in these tests are designed in such a way that all the options seem correct at first glance for an unprepared student. Answering questions in a standardized tests involves some of the most non trivial advances in Information Extraction, NLP and Computer Vision to solve. For example a simple arrow pointing to an object in a pulley system is not easily understood by a machine. Solving physics questions based on diagrams, chemical equations, solving math questions needing non trivial substitutions etc are extremely hard for computers to do. This may be considered one of the grand challenges of AI in general.

1.1 Why 8th grade?

In this work we target a very specific instance of this problem, namely, solving 8th grade multiple choice science questions. We restrict ourselves to 8th grade to avoid complications such as diagrams in physics, problems requiring mathematical calculations, chemical equations etc.

However we discover that even this restricted domain is hard to solve due to the level of semantic and background knowledge and understanding needed to solve these questions. Some questions are basically just information look-ups. Others involve significant amount of reasoning and background knowledge to answer. For such questions mere IR based look-up in the corpus leads to similar scores for all options. Some examples :

Example 1 : In the human body, which system functions primarily to defend the body against disease?

A Digestive
B Immune (*)
C Nervous
D Respiratory

This question can be answered by a simple look-up in the Knowledge Base.

Example 2 : What do earthquakes tell scientists about the history of the planet?

A Earth's climate is constantly changing.
B The continents of Earth are continually moving. (*)
C Dinosaurs became extinct about 65 million years ago.
D The oceans are much deeper today than millions years ago.

This question, which has 4 (seemingly accurate) statements as options, requires reasoning beyond a simple information look-up.

1.2 Objective

We analyze the comparative strengths and the lack thereof of various approaches of capturing the semantics and background knowledge in tackling this problem of answering domain-specific multiple choice questions. Additionally, we also explore the possibility of building a machine learning model to weigh the various evidences we obtain from various techniques. We then perform a case-by-case analysis of how and why some techniques worked better than others.

2 Question-Answering Systems - A Brief Overview

2.1 The Problem

Question Answering (referred to as QA henceforth) is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language.

QA research attempts to deal with a wide range of question types including: fact, list, definition, *How, Why,* hypothetical, and so on. There are 2 main domains of QA:

- **Open-domain QA** deals with questions about nearly anything, and hence can only rely on very generic ontologies and world knowledge. These systems usually have much more data available from which to extract the answer.
- **Closed-domain QA** deals with questions under a specific domain (like Football or History). They may also accept only a limited type of questions. This problem is considered relatively easy because NLP systems can exploit unstructured data as well as formal convenient domain-specific knowledge representations like ontologies.

By targeting 8th grade science questions, our problem fits into the closed-domain QA category. Moreover, having to choose one of the multiple choices requires a conceptually-strong flow of heuristics to rank-order the given choices, as compared to the open-domain category of descriptive questions, wherein after extracting relevant information, the model needs to use pre-defined templates or other NLG techniques to give out the final answer.

2.2 The Human Cognitive Approach

How do humans answer questions? Given a question, what is the underlying human cognitive process which returns an answer?

1. We read and 'make sense' of the question - what is the content, what is being asked.
2. Then we 'look up' the information we have about the specific domain of the problem, acquired through years of education and offline learning.
3. Now, we can either directly answer the question based on information nuggets that we already know, or we may have to connect different pieces of information and create a new nugget of knowledge relevant to the question.
4. Finally, we choose the best and most relevant answer. If this is a multiple choice question, we rank the options, and report the best one.

2.3 The Equivalent Machine Task

So how can this cognition process be incorporated in machines?

1. **Creating a Knowledge Base**

   The first step to answer any question is to have a good search corpus - for without documents containing the answer, there is little any QA system can do. A larger knowledge base generally leads to better QA performance, unless the question domain is orthogonal to the collection. Intuitively, if a person 'knows' a lot about a subject, he is expected to be able to answer more questions related to it correctly.

2. **Extracting keywords from the question**

   The next step is to identify some discriminative keywords which best represent the question. It is very important to find out 'what' information is required in order to answer the question correctly.
   For example, from the question - *'What is the process in plants to convert sunlight into energy*?', keywords like *'plants', 'process', 'sunlight', 'energy'* are representative of the question.

   - **The need for query expansion**
     There are cases when just extracting the keywords from the given question are not enough to represent the underlying intent of the question entirely. A precise example is given later.

3. **Information Retrieval**

   Once the keywords have been identified, an Information Retrieval (IR) system is used to find a set of documents containing the correct key words, after querying an ontology or a local search engine, which is the Knowledge Base.

4. **Ranking/Reporting the best result**

   On the basis of the relevance of the documents retrieved, a model needs to use pre-defined templates or other NLG techniques to formulate the final answer from the set of answer-relevant entities in case of descriptive questions. In our case, with multiple choice questions, we need to design an accurate and robust heuristic to rank-order the 4 options.

3 Notation

- **QA system :** Question Answering system
- **IR system :** Information Retrieval system
- **SEO :** Science Entity Overlap
- **Assertion :** For every question we have four options to choose from, of which only one is correct. Combining one question with one option gives an *assertion*, which are used to by the IR system to retrieve relevant documents.

4 Literature Survey

In this section, we briefly describe a few existing approaches for this problem, as proposed in the literature.

4.1 IR look-up [4]

Clark et. al propose many simple and basic methods which are surprisingly very strong. One of the suggested approaches is : For each assertion we retrieve a document and choose the option whose assertion gives the highest score.

This method gives good results **(42%)**, particularly on questions based on very direct information look-up and have only one assertion heavily dominating the other options. Examples of this are documented in the appendix.

4.2 QA as a Textual Entailment problem[5]

M. Sachan et. al formulate the question answering system as a Textual Entailment [7] problem. For every question, the paper considers the question as a *text* and each of the option as a *hypothesis*.

A Latent Structure Support Vector Machine (LSSVM) model is trained on a number of features such as scores from the baseline IR indexing, n-gram overlap in the query and the retrieved document at various depth levels such as book, chapter, topic etc. and features using Rhetorical Structure Theory. This approach captures some deeper semantic information compared to the naive baseline implementation describes above. They also present the accuracy scores obtained by various other baseline methods proposed in [4].

The paper reports an accuracy of **47.84%** on the data set provided by Allen Institute [2]. One important thing that the paper does not include is the background information needed to capture the questions that do not come directly from any of the text books prescribed for the exams. These can include questions that need knowledge of general facts, common sense etc. These indeed are hard to capture but improvements can be handled as described in the next paper.

4.3 Background knowledge and Expanded queries[18]

Modeling the background knowledge of a human is an extremely hard problem by itself. However methods such as domain specific ontologies, word2vec[16], LSA[12], ESA[10] have proved successful in many applications. The paper deals with question answering in university level history exams and uses Wikipedia as a source of background knowledge. The entire wikipedia dump is indexed using Lucene[9] and the best passages are retrieved based on IR with BM25 scoring.

Another key idea in the paper is expand queries and the documents to better capture relevant documents during indexing and retrieving. For every named entity in the assertion, similar entities are found using DBPedia-Spotlight. The retrieved documents are weighed based on a exponentially decreasing(2^{-n}) function w.r.t rank. Once the documents are retrieved the passages are re-indexed and the best passages are collected. Evidences are collected from various similarities between the assertion and the retrieved passages such as n-gram similarity, Jaccard similarity etc. History assertions include a lot of named entities, for example

Yan Zhenqing was a prominent Chinese calligrapher of the Tang Dynasty, and remains one of the most famous and emulated calligraphers today [4].

Hence using named entity overlap between the assertion and the retrieved passage as a feature helps in identifying the correct option. They also use similarity between the named entities is used as a feature.

Connectedness in a semantic graph is a good measure of semantic similarity between the entities. The paper uses personalized PageRank[13] to measure this connectedness with Wikipedia pages as nodes and links/redirects from one page to another as the edges to measure this connectedness. Personalized PageRank differs from the naive PageRank in the fact that the random walker can teleport to only a predefined set of nodes. This feature is important particularly in the context of history based questions because similar named entities in history would be connected to each other within fewer number of links.

4.4 Miscellaneous

Clark et.al in [6] provide the broad overview of the overall architecture needed to tackle the problem. They also show some of the requirements of the knowledge base needed to solve various kinds of questions are discussed.

Many questions in science examinations involve inferring from a lengthy comprehension or paragraphs. Recurrent neural networks are proving to be particularly successful in answering fact based comprehension based questions. QANTA by Iyyer et. al [11] when combined with IR based models was able to beat humans in answering such questions. However we do not address comprehension based questions in this work.

5 Methodology

5.1 Dataset and Corpus

Dataset : AI2 8th Grade Science Questions

- Publicly released by Allen Institute for Artificial Intelligence (founded by Paul Allen), this dataset contains multiple choice single correct science questions that are targeted for 8th grade students.
- The questions are taken from MCAS (Massachusetts Comprehensive Assessment System), TAKS (Massachusetts Comprehensive Assessment System), MEAP (Michigan Educational Assessment Program), and other such standardised tests.

Corpora :

- **Science textbooks**: A text file created from the Science Concept textbooks available freely at the CK-12 website, having a total of 124 chapters.
- **Wikipedia - Science Articles**: About 8000 articles obtained from the 'Science' category were used in Word2Vec and ESA for query expansion using this background knowledge.

5.2 Information Retrieval[14]

TF-IDF : In TF-IDF, similarity between the the query and the document is measured by for each term, its frequency is multiplied with the term's importance. The importance of the term is expressed using IDF (Inverse Document Frequency). IDF [14] is defined as

$$idf_t = \log \frac{N}{df_t}$$

Thus, the idf of a rare term is high, whereas that of a frequent term is likely to be low. Hence, the product of TF x IDF of a word gives a product of how frequent this word is in the document multiplied by how unique the word is w.r.t. the entire corpus of documents. Summing this for all terms the similarity score between the query and document can be obtained.

$$similarity\,(q, d) = \sum_{t \in q} \left(\log \frac{N}{df_t} \right) \cdot tf_{td}$$

Here, tf_{td} is the normalised frequency of term t in document d and df_t is the document frequency for term t.

Apache Lucene : Apache Lucene creates a local search engine, based on an full-text inverted index. It is able to achieve fast search responses because, instead of searching the text directly, it searches on the index instead. This would be the equivalent of retrieving pages in a book related to a keyword by searching the index at the back of a book, as opposed to searching the words in each page of the book. This type of index is called an inverted index, because it inverts a page-centric data structure (document → terms) to a keyword-centric data structure (term → documents).

Now given a query, similarity between the query and a document can be measured using TF-IDF. All the documents that contain at least one of the query terms are considered. For each term, IDF can be easily obtained by just taking the number of document occurances for the term from its inverted index. Also the inverted index also stores the term frequency of the term in the document. So ranking documents based on ID-IDF similarity is very fast.

BM25 similarity : It is a improved variation of TF_IDF. It introduces 2 parameters. One for varying the weightage of TF verses IDF. Another for scaling the similarity score for varied length documents. Similarity using BM25 [14]:

$$similarity\,(t,d) = \sum_{t \in q} \log \frac{N}{df_t} \cdot \frac{(k_1+1)\,tf_{td}}{k_1\left((1-b)+b\left(\frac{L_d}{L_{avg}}\right)\right)+tf_{td}}$$

Here, tf_{td} is the frequency of term t in document d, and L_d and L_{avg} are the length of document d and the average document length for the whole collection. The variable k_1 is a positive tuning parameter that calibrates the document term frequency scaling. A k_1 value of 0 corresponds to a binary model (no term frequency), and a large value corresponds to using raw term frequency. b is another tuning parameter ($0 \leq b \leq 1$) which determines the scaling by document length: $b = 1$ corresponds to fully scaling the term weight by the document length, while $b = 0$ corresponds to no length normalization. Experiments have shown reasonable values are to set k_1 to a value between 1.2 and 2 and $b = 0.75$.

5.3 Science entity overlap

In [18] Named Entity Overlap and similarity are taken as features in their model. It is well suited to their problem as they deal with history based questions which contain many name based assertions. This is not very efficient in our problem, however we can use overlap of the entities which are science related. The key intuition here is that correct assertion is likely to have more overlap of science entities than a wrong assertion.

To identify the entities related to science, we tried using a science dictionaries from some sources. But these hand made dictionaries have very few words. General words such as *plant*,*seed* etc are not usually found in the dictionary.

Wikipedia topic hierarchy : Each document in Wikipedia is arranged in segments, each with a topic name heading. These topics are further arranged in a hierarchy. Some of the topic headings are very generic such as *History, Applications, Examples* etc. Some of the topic headings provide very good science entity words. For example, the Wikipedia page for *Solar System* has all the planet names, important satellite names, and the names of other objects and asteroids in the solar system. However it also has topic headings such as *See also, Notes* etc.

We use intuition from Wordnet[8] to extract the science entities from the topic hierarchy. The key intuition is that deeper topics in the hierarchy are more specific than the shallow topics. Hence we consider all the topic names from a depth level of 2 onward as science entities. However we cannot discard all the shallow entities since some of them are indeed science entities. For example the word *Sun* occurs at the same depth as *See also* in the above example of *Solar System*. Hence at shallow depths we check if the topic name also has a main Wikipedia article about it. If yes we conclude that it is a science entity. This method gave good recall on the science entities extracted.

5.4 Query Expansion

One major problem with directly retrieving documents based on assertions generated is that the assertions could have keywords that are not exactly present in any of the indexed documents. Even if it is present in some, IR module could still miss some of the important documents.

A standard technique used to overcome this problem is to expand the keywords of the query to construct an expanded query. The intuition is that some of these key words occur in the indexed corpus and additional relevant documents are also retrieved. There are two steps in this process, identifying the words of the query that must be expanded and getting the closely related words to the identified words.

One naive way is to select all the words in the stemmed assertion and expand them, however this method gave worse results compared to the non-expanded query. This is because of the reason that a large number of irrelevant words are added causing irrelevant documents to be retrieved. In [18] all the named entities are expanded. This does good for history based questions but not for the science questions. Hence there is a need to identify a proper way to recognize these words. We use lemmatized Wikipedia topics and sub topics as the entity set. We expand any word occurring in this entity set.

Now the task is to identify the other relevant words for each entity word. For this we tried two approaches:

1. **Explicit Semantic Analysis[10] :**

   Using the Wikipedia documents as the concepts, we embed all the words in this concept space. Each word is represented as a vector of D dimensions where D is the number of Wikipedia documents. Each component $V_{i,d}$ in the vector V_i is a *term frequency* of the word in the document d normalized by the maximum frequency of the word in a document.

   $$V_{i,d} = \frac{freq_d(w_i)}{max_{d'}\left(freq_{d'}(w_i)\right)}$$

   To improve computational efficiency, we consider only the words that occur at least once in the science text book. Also only the Wikipedia documents tagged under Science category are considered.

2. **Word2Vec[16] :**

   Word2Vec is a shallow word embedding model, which takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. The intuition behind this is that the word vectors are positioned in the vector space in a way that words that share common contexts in the corpus are located in close proximity to one another in the space.

   Word2Vec can utilize either of two model architectures to perform a 'fake' task using a shallow (1 hidden layer) neural network to indirectly produce a distributed representation of words, i.e. the word vectors:

   (a) **Continuous Bag of Words (CBOW) :**
   - Task: Given the context, predict the missing word.
   - Input: A window of context words.
   - Output: Probability distribution over the vocabulary, predicting the missing word.
   - Outcome: Feature vectors for each word in the vocabulary.

   (b) **Skip Gram :**
   - Task: Given a word, predict its context.
   - Input: A single word.
   - Output: Probability distribution over the window-size
   - Outcome: Feature vectors for each word in the vocabulary

   Now that we have the word-embeddings, given a word, we can find the words that are most similar to it. So given a question, we first identify which words to 'expand', find the words most similar to them (above a certain threshold), and then use these additional words as well for informational retrieval.

6 Experiments

6.1 Evaluation Metric

We use accuracy as the evaluation measure defined as

$$Accuracy = \frac{Number\ of\ questions\ correctly\ answered}{Total\ number\ of\ questions}$$

This is in line with the general marking scheme in many standardized tests where one(or k) mark(s) is(are) awarded for every correct question and no penalty for wrong answers. The data set[2] provided to the public by Allen Institute has 293 questions on which experiments were done.

6.2 Baselines

We consider the following two baselines:

1. Random guess : Since there are four options in each of the questions, a random guess would result in **25%** of the answers being correct on expectation.

2. IR based baseline : This is a better baseline proposed in [4] which is surprisingly very strong. For each assertion we retrieve a document and choose the option whose assertion gives the highest score. This baseline performs surprisingly very good (133/293 or **45%**). It handles the look-up based questions where only one option has high score compared to others very well.

6.3 Indexing paragraph windows

A sliding window protocol was used to combine sentences into passages. Next, these passages were similarly indexed and the best relevant passage was retrieved. For this a paragraph constituted of 5 sentences. The next document started with the 4^{th} document of the previous document. This helps in maintaining the context of previous document in the next paragraph.
With this corpus the performance improved and now the accuracy obtained was **48.8%** (143/293).

6.4 Combining evidencers

Instead of just looking at the similarity score of the query with the first document weighted summation was done across Top K (of order of 1000s). Decreasing weights were given with decreasing ranks. The scoreing function used for each choice was:

$$score = \sum_{k=1}^{K} \left(\frac{a_1}{k + a_1} \right)^{\alpha} s_k^{\beta}$$

Here a_1, α, β and K are the tunable parameters. We took $a_1 = 3$, $\alpha = 2.2$, $\beta = 2$ and K = 2000. With this accuracy got boosted significantly (157/293 or **53.5%**).

6.5 Science entity overlap

We measure the overlap between each of the expanded assertion and the top 10 retrieved passages. We boost the scores of each of the assertion proportional to the science entity overlap(SEO) defined as follows:

$$SEO = \frac{\#\ of\ common\ science\ entities}{\#wordsinphrase + \#wordsinretrievedparagraph}$$

The weightage λ given to the SEO is a hyper parameter which is tuned. The best results were observed at $\lambda = 0.15$.

- Using SEO helped the model to get a couple of more correct answers (159/293 or **54.2%**).

6.6 Query Expansion using ESA

From all the words of the Wikipedia Science corpus, about 15000 remain in the concept space by considering only the words that occur in the CK12 textbook corpus. The normalized ESA vectors are computed as described in 5.3.1. For every science entity, we consider the words that are closely related to entity. Cosine-similarity based distance measure used. However this approach did not give good results. We attribute this to choosing only the words in the science textbook. However considering the whole Wikipedia gave vectors of dimension close to one million. Computations on these sparse vectors was time consuming, hence this approach was abandoned.

- After 'expanding' the words from the query which appear in the constructed science-dictionary, we see a significant deterioration in the model, with an accuracy of 147/293 (**50.1%**) after running for close to five hours.

6.7 Query Expansion using Word2Vec

The Continuous Skipgram model was used to generate the feature vectors. According to the authors, this is slower but better for infrequent words[16]. We chose this, since our domain of science words are relatively infrequent as compared to generic usage English words. The training algorithm used was Hierarchical Softmax, which again is claimed to be better for infrequent words, with a window size of 5, and dimensionality of 200.

To prevent the words of numbers such as *one, two* etc, we blacklist these words and do not expand them. Similarly the chemical formulas such as *Br, Cl, Fl* are also blacklisted. To prevent false positives such as *flourine, bromine* etc for *chlorine*, we put a threshold cutoff(0.75) and we take a cautious path and do not consider the word expansion if it results in too many words with high confidence. Taking all these measure helped in improving the accuracy.

- After 'expanding' the words from the query which appear in the constructed science-dictionary, we get an accuracy of 165/293 **(56.31%)**.

6.8 Machine Learning Models

Using all scores calculated above, we tried to build a machine learning model to learn optimal weights to the various scores. Linear models worked well but gave similar accuracy as the hand adjusted weights by validation. Other models considered were Logistic Regression, SVM, Decision trees etc. However these models did not perform well. We attribute this to less training data and non-exhaustive features.

7 Observations

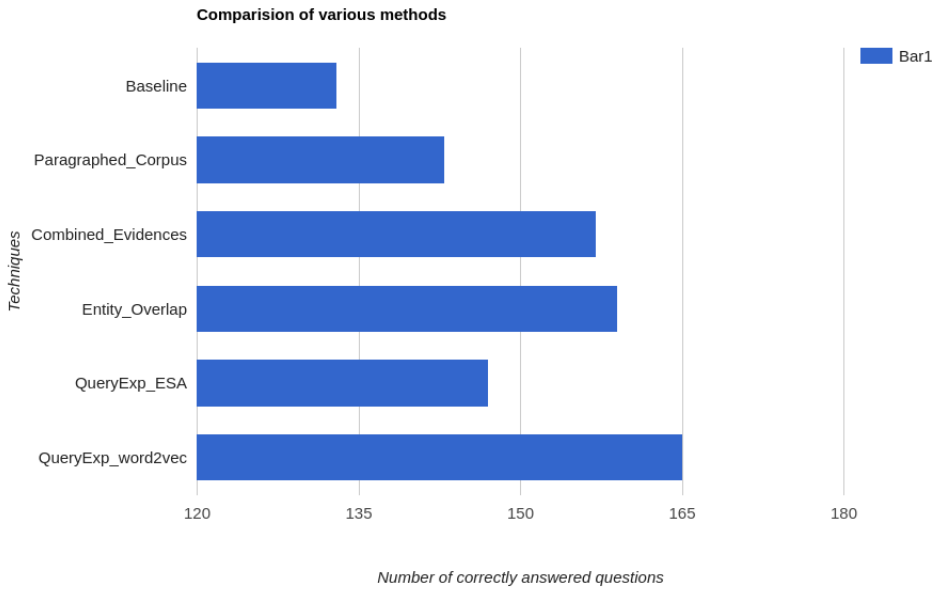

Fig. 1. Comparison between the techniques

Table 1. Performance on various type of questions

Technique	Negation based	Others	Look-up based	Inference based
Baseline	2	131	63	70
Paragraphed corpus	4	161	69	74
Combined evidences	3	154	76	81
Entity overlap	4	155	77	82
QueryExp - Word2Vec	4	161	79	86
Total	**10**	**283**	**122**	**171**

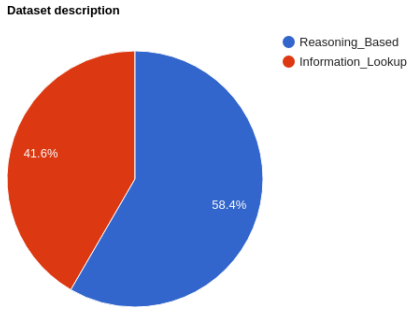

Fig. 2. Number of questions with just information look up and reasoning based

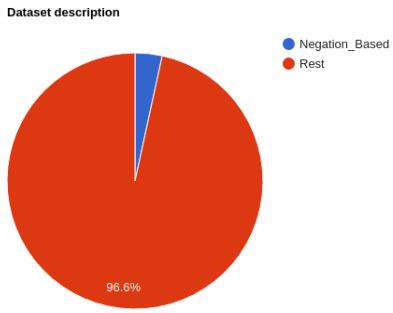

Fig. 3. Number of questions asking for least similar output

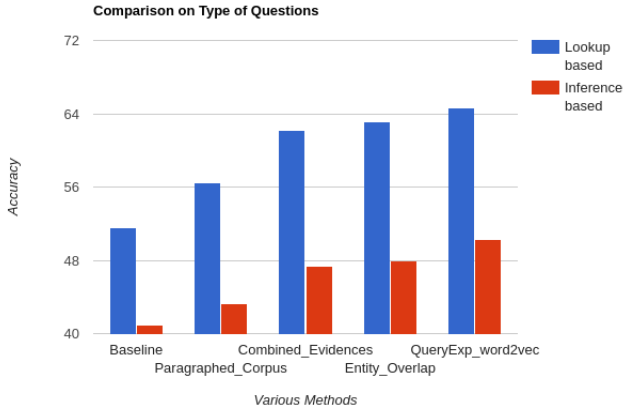

Fig. 4. Comparison between the techniques

8 Examples

Example 3 : Scientists think that dolphins and whales may have evolved from a common ancestor. What evidence supports this hypothesis?

A They swim the same way.
B They eat the same food.
C They live in the same area of the ocean.
D They have similar anatomies. (*)

This question illustrates the fact that naive query expansion of all the words in assertion does not do well. This is because of the fact that query expansion adds words like *know*, *ask*, *tell*, *must*, *should*, *descendants*, *ancestral*, *idea*, etc. which are not relevant to the main focus of the question. This gives an answer of (A), whereas without query expansion, the answer is (B).

Example 4 : Which property would best help a student determine if two substances are made of two different elements?

A mass.
B shape.
C density. (*)
D volume.

This question again illustrates the fact that naive query expansion using word2Vec does not give good results. This is because of the fact that word *two* gets expanded and results in words like *one*, *three*, etc. This example motivates the reasoning for the need to choose proper words for query expansion.

Example 5 : Which of the following areas is most likely to form metamorphic rocks such as gneiss and schist?

A a sea floor
B a windblown desert
C a site deep underground (*)
D a site covered by a glacier

This example illustrates the case where naive query expansion using word2Vec expands indicative science word but gives incorrect answer. In this case, the rocks *gneiss* and *schist* expand and give results like *sedimentary*, *siliciclastic*, etc. which are the rocks found in sea-floor and hence the answer turns out to be (A).

Example 6 : What do the elements sulfur (S), nitrogen (N), phosphorus (P), and bromine (Br) have in common?

A They are noble (inert) gases.
B They are nonmetals. (*)
C They have the same thermal conductivity.
D They have the same number of protons.

This example illustrates the case where naive query expansion using word2Vec results in correct answer compared to basic information retrieval approach. Here, the query expansion adds words like *chlorine*, *iodine*, etc. which are all non-metals and hence the answer becomes (B).

It also illustrates the scenario where science entity overlap performs better compared to top 1000 paragraph based Information Retrieval. Here, the documents retrieved for *noble (inert) gases* have higher BM-25 index compared to *non-metals* option due to idf-score. This is not the intuition which we want to capture. But the passage retrieved for *non-metals* option has '*...group 15 of the periodic table is also called the nitrogen group the first element in the group is the nonmetal nitrogen followed by phosphorus...*' which basically captures the requirements of question through entity overlap.

Example 7 : Which of the following elements has the atomic number of 9?

A Florine. (*)
B Chlorine.
C Bromine.
D Iodine.

This example illustrates the fact that query expansion of options is not a good choice. This is because of the fact that query expansion of *Florine* results in *Chlorine*, *Bromine*, *Iodine* and similarly query expansion of each option results

in all the other options coming up as similar words.

Example 8 : Heat from deep in Earth's interior is transferred to its crust by which of the following?

A conduction in the ocean
B convection in the mantle (*)
C radiation from the solid core
D evaporation at mid-ocean ridges

This example illustrates the scenario where restricted query expansion does better compared to entity overlap. Due to query expansion *mantle* gets added into the expanded words and this results in retrieval of paragraph - '*...convection and conduction describe the movement of heat in the mantle...*' which describe the actual motion of heat from earth's interior.

Without query expansion, we retrieve '*...magnetic polarity reveal the different ages of the seafloor in some places the oceanic crust comes up to continent the moving crust pushes...*' which contains lots of references to "oceanic crust" preferring (A) as the answer.

Example 9 : A galaxy is best described as a cluster of ?

A hundreds of stars.
B thousands of stars.
C millions of stars.
D billions of stars. (*)

This example illustrates the typical advantage of query expansion. Query Expansion of *galaxy* adds *milky* describing milky way and since the number of stars in milky way is more commonly described in corpora than that of a galaxy, it retrieved '*... astronomers estimate that the milky way contains 200 billion to 400 billion stars ...*'.

Without Query expansion, the top paragraph retrieved is '*... near the center of globular cluster the stars are closer together. the heart of the globular cluster m13 has hundreds of thousands of stars ...*' which describes a single galaxy m13 and since it contains the word 'cluster' few times, the information retrieval focuses onto this which describes hundreds of thousands of stars in the center of galaxy m13 and hence gives the answer of (B).

Example 10 : Which two processes in the water cycle are primarily responsible for the creation of a lake?

A evaporation and runoff
B evaporation and condensation
C precipitation and runoff (*)
D precipitation and condensation

This example illustrates another example where query expansion helps in inferring the correct answer. Due to query expansion the word *river* gets added. Since precipitation and run-off in the documents generally describe the formation of rivers over lakes (though both form from same phenomenon), addition of *river* gave '*if the air is cold the water may freeze and fall as snow sleet or hail most precipitation falls into the oceans some falls on land runoff is precipitation that flows over the surface of the land this water may travel to river lake or ocean ...*'

But without query expansion, we retrieve '*...two sources of water for evaporation in the water cycle after water evaporates...condensation occurs relate dew point to condensation...*' and it identifies condensation and evaporation as answer. Hence, addition of *river* increased the emphasis on *river* and *lake* and correct answer was obtained.

Example 11 : Which of the following structures is not present in animal cells ?

A cell membrane
B cell wall (*)
C mitochondrion
D nucleus

This example illustrates a question which contains negative sense. This also illustrates the scenario where science entity overlap performs better than top 1000 paragraph based Information Retrieval. The top documents retrieved for option *cell membrane* had higher BM-25 similarity score compared to those retrieved for *cell wall*. This is because of the fact that *membrane* has higher idf compared to *wall*.

Even though *cell wall* has lower BM-25 score, it is the answer to this question and the top document retrieved for this is '*...cell wall large central vacuole and plastids these three features are not found in animal cells...*' which contains the required information for the question. This is captured by the entity overlap between assertion and retrieved documents. The negation in the question is also captured directly in the entity overlap between assertion and retrieved documents.

Example 12 : Which of the following causes a ships iron anchor to sink to the ocean floor when it is released overboard?

A chemical forces
B gravity (*)
C magnetism
D nuclear forces

This example illustrates comparison between baseline(one-line retrieval) versus paragraph(sliding-window retrieval). Intuitively sliding window has a greater

context window than a single line and allows the retrieval of more relevant documents.

In the case where a single sentence is considered as a document, '*scientists don't know for certain why magnetic reversals...the evidence comes from rocks on the ocean floor*' is retrieved as the top relevant. But if we consider a higher length context window, we get '*if steel ball with the same weight as the ship were placed in water it would sink to the bottom ... the buoyant force is not as great as the force of gravity pulling down on the ball ...*' which is more relevant to the question.

Example 13 : Sugar is composed of carbon, hydrogen, and oxygen. Sugar is an example of which of the following?

A an atom
B a compound (*)
C an electron
D a mixture

This example illustrates another scenario for comparison between baseline(one-line retrieval) versus paragraph(sliding-window retrieval).

Using sentence based retrieval (baseline), we get '*...consider carbon as an example carbon atoms have six protons they also have six electrons all carbon atoms ...*' giving *atoms* as the answer. But a sliding window of passages retrieves '*...glucose is simple sugar that living cells use for energy. all other compounds ...molecule of glucose an organic compound composed of carbon hydrogen and oxygen ...*' which captures the required context for the question to be correctly answered.

Example 14 : Which of the following keeps the planets in our solar system in orbit around the Sun?

A atmospheric pressure
B gravitational force (*)
C electromagnetic energy
D thermal energy

This example illustrates case where weighted average of top-1000 paragraphs based retrieval performs better than single paragraph based retrieval.

The single top document retrieved for the *gravitational force* assertion does not have the words *gravitational force* in it and hence the assertion scores less compared to *atmospheric pressure* assertion. But if we retrieve top-1000 documents instead and take their weighted average, we get the *gravitational force* assertion as the highest score.

---

9 Future work

There is a lot of scope for improvement. Better features can be researched to train a machine learning model on the features to achieve better accuracy score. The techniques we used in this work can be broadly categorized into IR based methods augmented with semantic and background information. Another way to approach this problem is from a logical reasoning perspective. Given a set of true logical assertions say from the text book, an inferencer that takes in questions as logical queries and tries to infer the truth or falsity of the query. These models are based on Ontologies. We conjecture that a proper utilization of both the approaches could result in better understanding of the text books and background knowledge coupled with better reasoning on the questions.

10 Acknowledgments

We thank our mentors Neha Dubey and Shashank Shrivastava for their suggestions in formulating our problem and for providing the initial direction. We also thank Ayesha Siddiqa, Ditty Matthew and Professor Sutanu Chakraborti for their comments and review of the problem formulation.

References

1. The Allen AI Science Challenge, Is Your Model Smarter Than An 8th Grader? [Link].
2. Allen AI Science Questions Datasets. [Link].
3. Peter Clark and Oren Etzioni. My Computer Is an Honor Student, But How Intelligent Is It? Standardized Tests as a Measure of AI. [Link].
4. Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, and Peter Turney. Combining Retrieval, Statistics, and Inference to Answer Elementary Science Questions. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. [Link].
5. Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, and Peter Turney. Science Question Answering using Instructional Materials. 2016. [Link].
6. Peter Clark, Phil Harrison, and Niranjan Balasubramanian. A Study of the Knowledge Base Requirements for Passing an Elementary Science Test. 2013. [Link].
7. Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges Lecture Notes in Computer Science Vol 3944 pp 177 to 190, Springer*, 2006. [Link].
8. Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database*. The MIT Press, 1998. [Link].
9. Apache Software Foundation. Apache Lucene - Scoring, 2011. [Link].
10. Evgeniy Gabrilovich and Shaul Markovitch. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, San Francisco, CA, USA, 2007. [Link].
11. Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. A Neural Network for Factoid Question Answering over Paragraphs. In *Empirical Methods in Natural Language Processing*, 2014.
12. Thomas K Landauer, Peter W Foltz, and Darrell Laham. An Introduction to Latent Semantic Analysis. In *Discourse Processes No25, pp 259-284*, 1998. [Link].
13. Page Lawrence, Brin Sergey, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.
14. Christopher D Manning, Prabhakar Raghavan, and Hinrich Schutze. Introduction to information retrieval, 2008.
15. John Markoff. Software Is Smart Enough for SAT, But Still Far From Intelligent. In *The New York times*, Sept20 2015. Link.
16. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013.
17. Alan Turing. Computing Machinery and Intelligence. 1950.
18. Di Wang, Leonid Boytsov, Jun Araki, and Alkesh Patel. CMU Multiple-choice Question Answering System at NTCIR-11 QA-Lab. In *Proceedings of the 11th NTCIR Conference*, 2014. [Link].