Integrating Question Classification and Deep Learning for improved Answer Selection

Harish Tayyar Madabushi
School of Computer Science,
University of Birmingham,
United Kingdom.
H.T.Madabushi@gmail.com

Mark Lee
School of Computer Science,
University of Birmingham,
United Kingdom.
M.G.Lee@cs.bham.ac.uk

John Barnden
School of Computer Science,
University of Birmingham,
United Kingdom.
J.A.Barnden@cs.bham.ac.uk

Abstract

We present a system for Answer Selection that integrates fine-grained Question Classification with a Deep Learning model designed for Answer Selection. We detail the necessary changes to the Question Classification taxonomy and system, the creation of a new Entity Identification system and methods of *highlighting* entities to achieve this objective. Our experiments show that Question Classes are a strong signal to Deep Learning models for Answer Selection, and enable us to outperform the current state of the art in all variations of our experiments except one. In the best configuration, our MRR and MAP scores outperform the current state of the art by between 3 and 5 points on both versions of the TREC Answer Selection test set, a standard dataset for this task.

1 Introduction and Motivation

Question Answering (QA) is the task of automatically generating answers to questions posed in natural language. The task has received significant attention from researchers over several decades with a renewed interest in recent times. Current interest has been partly due to significant improvements in Natural Language Processing, and partly due to the increase in demand for such systems, amongst both the general populace, and corporate entities.

An important subtask of QA is Question Classification (QC), which deals with the classification of questions based on the expected class of the answer. For example, the question "How much does the Capitol Dome weigh?" could be classified into the class "Numeric, Weight", while the question "Name the actress in the movie Titanic" could be classified into the class "Human, Individual". While there exist QA systems that do not make use of QC, the addition of QC to a QA system has been shown to improve its accuracy (Hovy et al., 2001). The specific classification that a QC system uses is called its taxonomy, and taxonomies vary quite widely in both specificity and form.

Intuitively, QC improves QA by reducing the search space of potential answers, thus making the discovery of answers more efficient and accurate. While research into QC is fairly mature, fine grained QC is not always used for QA. For example, Tsai et al. (2015) use very simple rules based partly on question wh-words (i.e. "what", "where", "who", etc.), resulting in 13 question classes. Not only is this number low compared to the 50 classes defined by (Li and Roth, 2002) or the over 120 used by (Hermjakob, 2001), the rules themselves are fairly weak. As an example, Tsai et al. (2015) classify all questions containing "name" under the class "Name" ("Name the country most famous for cheese.")

Most QA systems consist primarily of three components: **a)** a question analysis component, **b)** an Information Extraction (IE) component that extracts a set of candidate sentences, and **c)** an answer extraction component that prunes this set of sentences in order to extract the answer. QC is performed in the first component. Its results are sometimes useful in the IE component, but generally most useful in answer extraction. The other important aspect of the answer extraction component is the analysis of linguistic features. Together, these two elements can be used to prune a set of sentences, some of which might contain the answer to a given question.

Proceedings of the 27th International Conference on Computational Linguistics, pages 3283–3294
Santa Fe, New Mexico, USA, August 20-26, 2018.

This task of selecting, from a list of sentences produced by an IE component, a subset A (where $|A| \geqslant 0$) which contains the answer to a given question is called Answer Selection (AS). For example, given the question "Where is the group Wiggles from?", and two possible sentences (called *candidate sentences*): "the Wiggles are four effervescent performers from the Sydney area: Anthony Field, Murray Cook, Jeff Fatt and Greg Page", and "six of the Wiggles' videos have reached multiplatinum status in Australia", the task would require one to return the first (*positive*) candidate and not the second (*negative*) candidate. We note that AS leaves the task of extracting the Answer from a positive candidate to a downstream task.

Methods of AS rely on establishing some form of relation between the question and each of the answer candidates, such as bag-of-words, tree edit models (Heilman and Smith, 2010), semantic distances based on word embeddings (Wang and Ittycheriah, 2015), or deep learning methods such as Convolutional Neural Networks (Rao et al., 2016). To the best of our knowledge however, this task has not been attempted with extensive use of fine grained QC.

2 Related Work

A lot of the work into using QC for QA took place before the resurgence of Machine Learning. For example, Kwok et al. (2001) introduce a QA system "MULDER", that makes use of wh-phrases, which they define as the interrogative word followed by the words associated with it. Hermjakob (2001) used an extensive QC system consisting of 115 elementary question classes in their work on QA.

2.1 Question Taxonomy and Classification

The specific system of classes used by a QC system is known as a taxonomy, and while several taxonomies are available, we pick that proposed by Li and Roth (2002), for two reasons: **a)** This is one of the most widely used taxonomies, possibly because of the large training set that Li and Roth (2002) provide, **b)** while it has been pointed out that this taxonomy might not have the widest coverage (Mishra and Jain, 2016), we show below that it is most suited for domain independent QA.

This taxonomy originally consisted of fifty fine classes divided amongst six coarse classes. Table 1 provides a complete list of these classes along with the changes we make (described in Section 4.1).

While our work on QC is an extension of the work by Tayyar Madabushi and Lee (2016), a rule-based system that achieved an accuracy of 97.2%, other work on the same taxonomy has involved the use of Linear SVMs by Van-Tu and Anh-Cuong (2016) and Pota et al. (2016) which achieved accuracies of 91.6% and 89.6% respectively. Work using Convolutional Neural Networks (Kim, 2014) and Skip-Thought Vectors (Kiros et al., 2015) have not focused on fine-grained classification.

2.2 Answer Selection

Answer Selection became popular as a task after being proposed by Punyakanok et al. (2004), who modified the TREC QA task to that of AS. This modification does not simplify the task, as extracting relevant sentences is not only as hard as extracting the actual answer, but users often find it more useful to see answers to their questions in their original context (Wang et al., 2007).

Wang et al. (2007) follow a similar approach to Punyakanok et al. (2004), while providing a training set extracted from TREC 8-12 datasets and setting aside the TREC 13 dataset for development (84 questions) and testing (100 questions). AS has since become the standard in measuring the accuracy of Question Answering systems[1] and the dataset has since diverged into two versions: The "Clean Version" (Wang and Ittycheriah, 2015), which has been cleaned to remove questions with no candidate answer sentences and those that have no negative candidate sentences from the development and the test sets; the non-cleaned version called the "Raw Version". Rao et al. (2016) have shown that results from the two datasets are not comparable. Results for this task are reported using two measures, common in Information Retrieval and Question Answering Research: Mean Average Precision (MAP) and Mean

[1] http://www.aclweb.org/aclwiki/index.php?title=Question_Answering_(State_of_the_art)

Reciprocal Rank (MRR). These results are attained using the standard program *trec_eval*, provided by TREC.

Recent work on Answer Selection has depended heavily on word embeddings with the state of the art work on the Raw version by Rao et al. (2016) who rank candidate sentences using a Multi-Perspective Convolutional Neural Network and that on the Clean version by Shen et al. (2017) who use a novel method of sentence pair modelling. Previous work on the same dataset used similar models including that by dos Santos et al. (2016) that made use of Attentive Pooling CNNs and that by Bian et al. (2017) who use a Compare-Aggregate framework. A complete list of work on the dataset is available on the ACL wiki on Question Answering.

3 System Overview and Contribution

In working towards a method of integrating QC into AS, we first redefine the taxonomy provided by Li and Roth (2002) to better suit entity identification, before then modifying the Question Classification system developed by Tayyar Madabushi and Lee (2016) to match this modified taxonomy. We then create an entity identification method to extract entities belonging to those classes in our taxonomy. Finally, we use different methods of "highlighting" entities, so this information can be passed on to *any* model that uses word embeddings. We use the model developed by Rao et al. (2016), which performs AS, to test our method.

In addition to showing the significant impact that Question Classification has on Answer Selection, we make several datasets available so others might exploit QC in Question Answering tasks including a Question Classification API that reflects the modified taxonomy[2].

4 Question Classification

Our experiments with using the taxonomy proposed by Li and Roth (2002) showed the need for changes to allow the classification system to lend itself more easily to Entity Identification and AS. For one, we found that some categorisations would make entity identification harder. For example, the question "What's the world's longest suspension bridge?" is categorised under "Location" while we believe that it is more appropriate to consider a bridge an entity. We base this on the hierarchical classification provided by WordNet (Miller, 1995), an extensive human crafted hierarchical dictionary that has been used as a standard for several different tasks.

Similarly, we disagree with the prioritisation of the classes provided. Prioritisation is important as this particular taxonomy does not allow a question to be a part of two classes. As an example, the question "What country did the ancient Romans refer to as Hibernia?" can be classified as either belonging to the class "Location:Country" or "Entity:termeq" (Equivalent Term). While Li and Roth (2002) categorise this question under the first, we categorise it under the latter, because the question is not about where something is or happens. We also believe that this choice makes it easier for a downstream QA system.

We also found the need for a new class that constitutes either "Human:Individuals" *or* "Human:Groups" (such as companies, teams and universities). This specific requirement is a direct result of the restriction that a question must be classified without prior knowledge of the answer. For example, the question "Who won the Nobel Peace Price in 2012?" is impossible to classify without knowing if the answer was an organisation (as it was in 2012) or an individual (as in 2016), even if we were to ignore the possibility of multiple individuals (as in 2014). To get around this we introduce the class "Human:IndividualOrGroup". We retain the classes "Human:Individuals" and "Human:Groups" for instances where the distinction is clear.

Finally, we find that certain types of entities within certain classes are much more frequent than others in that class. While this could be because of the specific method we use for Entity Identification (Section 5), we create separate classes for these types of entities so as to avoid noise in our AS feature generation. We also expand the class "Location:State" to include the provinces of Canada and the counties of the U.K. We list the taxonomy thus modified in Table 1.

[2] `www.harishmadabushi.com/research/questionclassification/`

Coarse	Fine
ABBR	abbreviation*, expansion*
DESC	definition*, description*, manner*, reason*
ENTY	animal, body, colour, creation, currency, disease, event, food, instrument, language, letter*, other*, plant, product, religion, sport, substance*, symbol*, technique, term*, vehicle*, word*, **movie***, **book***, **extraterrestrial**
HUM	description*, group, individual, title, **individualOrGroup***
LOC	city, country, mountain, other, **state**
NUM	code, count, date, distance, money, order, other*, percent, percent, period, speed, temperature, size, weight, **year**, **volume (Size)**, **volume (Liquid)**, **time**, **numeric range***

Table 1: Question Taxonomy introduced by Li and Roth (2002), with our modifications in bold (Section 4.1) and those classes not used in AS starred* (Section 8)

4.1 Modifications to Question Classification

The QC system used by us is an extension of that by Tayyar Madabushi and Lee (2016). It primarily involves: **a)** extracting a Question's Syntactic Map (a structure they define for holding certain types of syntactic information), **b)** identifying the headword of the noun phrase in the question, while handling Entity Identification and phrase detection, and **c)** using rules to map words at different positions in the Syntactic Map to question classes using a hyrarchical structure.

Their system classifies questions as follows: Consider the question "What is the name of the actress from England in the movie 'The Titanic'?". The system identifies its Question Class by analysing the question's parse tree to generate the Question's Syntactic Map, which enables the identification of the headword "actress" using, what they call, "prepositional rolling". This process provides us with the question's wh-word ("What"), the auxiliary verb ("is"), and the headword ("actress"). This information is used by the system to check for the existence of a rule that classifies this question. Such a rule is found by matching the noun "actress" to the rule: 'occupation.n.01' and its hyponyms in this section of the question when the wh-word is 'what' indicate that the question class is hum:ind.

These rules are manually defined using sets of WordNet synsets they call Types. Types are defined by manually picking specific synsets within WordNet and associating them and all their hyponyms to a particular Question Class based on where in a question they appear. In the previous example, the relevant Type is the word occupation and all hyponyms of the synset 'occupation.n.01'. Similarly, the synsets 'people.n.01', 'organization.n.01', 'university.n.01', 'company.n.04', 'socialgroup.n.01', and all of their hyponyms are assigned to the Question Class "Human Group".

For a detailed description, we direct the reader to the original work. However, we describe herein some elements which we have modified, a necessity given our changes to the taxonomy.

4.2 Word Sense Disambiguation and Rule Extensions

A primary difficulty in identifying the specific rule to use once the correct head of the question has been identified arises due to the polysemous nature of some words. For example, the question "What rank did you achieve in the test?" and the question "What rank did she achieve in the military?" both have the same headword "rank" but differ in the meaning of that word (position in ordering versus military status such as captain). The question class assigned to each of these question must also change based on these meanings (Number:order versus Human:title).

As described by Tayyar Madabushi and Lee (2016), words useful in identifying the question class are often nouns, as in the case of the question "What is the name of the *actress* in the Titanic?". However, such words need not always be a noun. In the case of the question "How much does the President get paid?", for example, it is the adverb "much" which allows us to infer that the expected answer is a number and additionally, the word "paid" allows us to infer that the number represents money hence resulting in the question class "number:money" as opposed to the question class "number:weight" as in the case of "How much does the Big Ben weigh?"

Rules defined by the QC system map sub-trees in WordNet to specific question classes. We make changes to the rules to align the classification of questions with the modifications we make to the taxonomy (Table 1) and add further rules where possible to cover a larger section of WordNet. Additionally, there are instances wherein the system makes use of certain heuristics to find the appropriate rule to use, as in the case of questions starting with "How much ..." which sometimes leads to classification errors. To mitigate this problem, we modify the system to return a possible second class when there is ambiguity.

5 Named Entity Recognition

Given that our objective is to "highlight" all entities in candidate answers that belong to the class assigned to a particular question, we require a method of Named Entity Recognition (NER) at the same granularity as our taxonomy. Unsurprisingly, there is no off the shelf NER system that identifies entities with the exact granularity and classes that we classify questions into. To get around this problem we start by relating entities in text to Wikipedia titles and subsequently mapping those titles to our classes. This process of mapping entities in text to Wikipedia titles is called Wikification. The hierarchical tree-structure provided by Wikipedia helps in mapping a large number of titles to a given class by allowing us to map sub-trees to classes.

5.1 Wikification

Wikification was introduced by Mihalcea and Csomai (2007), as a means of automatic keyword extraction and Word Sense Disambiguation. It has since been used for a variety of tasks especially the semantic enrichment of text. A significant advantage of using Wikification is that entities, once identified, are in a normalised format, namely the title of the linked Wikipedia article, thus making entity matching (Section 6) easier.

While simple entity identification involves the direct matching of phrases to Wikipedia titles, more advanced versions of Wikification additionally involve mapping phrases to *related* titles based on the contents of the Wikipedia article. For example, one might choose to map the phrase "the first Briton in space" to the Wikipedia article on "Helen Sharman", who was the first Briton in space. We however, limit ourselves to the simpler version as we are only interested in finding entities and not concepts.

Typically, Wikification involves the identification of potential entities and the subsequent matching of those entities with Wikipedia titles. For example, given a sentence, one could potentially use a Parts of Speech (PoS) tagger to tag the sentences before then extracting sequences of PoS tags that match a predefined set (such as NNP+ or DT*NNP+ and so on). Entities thus extracted could then be matched with Wikipedia titles.

After experimenting with several off the shelf Wikification tools, we found them lacking in the ability to work with Wikipedia Disambiguation pages and topic specific pages. For example, when looking for entities of type "movie" in the sentence "He went to watch the movie 'New York"', we want to be able to match this to the Wikipedia article "New York (film)" and *not* "New York (state)" or "New York City".

5.2 Wikification without PoS Tagging

The obvious way to tag entities in text with Wikipedia titles would be to match every possible phrase in a sentence with every title on Wikipedia. This, however, is impractical as there are over 13.04 million titles in the English Wikipedia. To get around this we run through the titles, and for each title, we split it into its constituent words and save the rest of the title in a file whose name is the first word. Thus, all titles that begin with a particular word are clubbed into a single file and for those titles that are of length one, we add an empty line into the corresponding file. This results in just over 2.1 million files each of which are relatively short and easy to process.

As we sweep through each word in a sentence, we process the file containing Wikipedia titles starting with the same word, and check to see if it contains entities that match the current sentence. This greatly speeds up the process of matching titles to the words in a sentence and provides us with a list of titles that are contained in a given sentence.

We note that this method of Wikification can be used in languages where capitalisation is dissimilar to English or even those languages wherein there is no capitalisation.

5.3 Wikification to Question Classes

Once candidate answers are Wikified, we are then left with the task of mapping these titles to the question classes. We do this by first linking each Wikipedia title to the corresponding DBPedia entry. DBPedia is an attempt to extract structured information from Wikipedia and provides a list of labels and classes associated with each entry. We use these labels and classes to map Wikipedia (and so DBPedia) titles to question classes associated with our taxonomy.

5.4 NER without Wikification

We use the Stanford Named Entity Recogniser (Finkel et al., 2005) to identify entities belonging to the classes "Human Individual", "Human Group" (such as institutions, universities, etc.), and "Location Other", the three classes with compatible granularity. All numeric entities, such as Number:Money, Number:count, and Number:date are identified using an extensive list of regular expressions.

6 Entity Matching

When grading papers, a good maxim to identify plagiarism is "While there is only one way to get it right, there are several ways to get it wrong". We observe that this maxim works because the probability of two students answering a question incorrectly in *the same way* is extremely small, unless of course it's a trick question. Similarly, the chance of candidate answers having the same incorrect entity that also match the class of the question is exceedingly small and machine learning models can make use of this information. To this end, we count the number of occurrences of each entity across all answer candidates of a given question. This requires us to be able to match entities that have been written differently, but are in fact the same.

Entities that have been extracted through Wikification are often normalised "for free." However, there is no simple way to get around this problem in the case of entities extracted through regular expressions, as in the case of numbers and dates where it is common for sentences to contain approximations. For example, consider the question "How many lives were lost in the recent air-crash?", the answer is contained in all of the following sentences: "253 lives were lost in the recent air-crash", "241 passengers and 12 crew died in the recent air-crash", and "around 250 lives were lost in the recent air-crash". This problem is further expanded when the numbers we are dealing with become larger as it is more common for non-technical literature to approximate large numbers. To get around this we round down all numbers to the nearest billion, million, hundred thousand, thousand, hundred or ten.

7 Model Details

We use the Answer Selection model developed by Rao et al. (2016) who rank candidate sentences using a Multi-Perspective Convolutional Neural Network (He et al., 2015a) and a triplet ranking loss function which uses triplets of the question paired with a positive and a negative candidate answer. While other methods model this problem as a pointwise classification problem (He and Lin, 2016; Severyn and Moschitti, 2015), this method models the problem of Answer Selection as a pairwise ranking problem. This involves developing representations for positive and negative answer candidates paired with the question, the primary reason for us choosing this model.

Yet another advantage of this method is that it can make use of existing pointwise models to generate representations which can then be fed into the triplet ranking function. The authors make use of two such pointwise models, one that uses a sentence-level model (He et al., 2015b) and the other that uses a word-level model (He and Lin, 2016). We refrain from elaborating on these methods due to the limitations of space and refer to the reader to the original works.

We make use of the word-level model[3] as we introduce new representations for entities (Section 7.1) which requires us to modify them with answer candidates. The model is additionally initialised with the

[3] https://github.com/castorini/Castor

GloVe word embeddings (Pennington et al., 2014) which are also updated during training.

7.1 Highlighting Entities

Having extracted and normalised entities that are contained in each of the answer candidates, we are faced with the task of highlighting these entities within the answer candidates. Before we do this however, we perform some prepossessing steps. We discard any entities that also appear in the question as such entities are unlikely to be the answer. For example, the question "Who is the author of the book, 'The Iron Lady: a biography of Margaret Thatcher"' has, as an answer candidate, the sentence "The iron lady; a biography of Margaret Thatcher by Hugo Young" in which both "Margaret Thatcher" and "Hugo Young" are entities that match the question class, namely "Human Individual". The entity "Margaret Thatcher", however, is discarded as it is also contained in the question.

For each question we count the number of occurrences of each entity across all candidate answers and if the most frequently occurring entity occurs more than twice the number of times the second most frequently occurring one, we pick the first as the maximal entity. For those questions where this is not the case, we pick no maximal entity.

We also create four new "words", *max_entity_left*, *max_entity_right*, *entity_left*, and *entity_right*, which are strings that are not contained in the vocabulary, along with associated word vectors which are randomly initialised with entries between -0.05 and 0.05 and are of the same length as the GloVe word embeddings (300). We then add these embeddings to our embedding dictionary and the words to the vocabulary.

Entities are highlighted in the answer candidates by inserting the words *max_entity_left* and *max_entity_right* on either side of maximal entities, and *entity_left* and *entity_right* around other entities. We also include *entity_left* and *entity_right* at the end of the question and the "words" *max_entity_left* and *max_entity_right* at the end of questions that contain maximal entities. We call this method of highlighting *bracketing*.

A second method of highlighting entities in candidate answers is to replace the entity with a word, a method we call *replacing*. To avoid creating two new words for this method, we reuse two of the four words used above: *max_entity_left* and *entity_left*. Table 2 details the modifications made to an example question and candidate answer using each of the above methods.

Method	Question	Answer Candidate
Original	Who is the author of the book, 'The Iron Lady: a biography of Margaret Thatcher'	in 'The Iron Lady,' Young traces the winding staircase of fortune that transformed the younger daughter of a provincial English grocer into the greatest woman political leader since Catherine the Great.
Bracketing	Who is the author of the book, 'The Iron Lady: a biography of Margaret Thatcher' *max_entity_left max_entity_right entity_left entity_right*	in 'The Iron Lady,' *max_entity_left* Young *max_entity_right* traces the winding staircase of fortune that transformed the younger daughter of a provincial English grocer into the greatest woman political leader since *entity_left* Catherine the Great *entity_right*.
Replacing	Who is the author of the book, 'The Iron Lady: a biography of Margaret Thatcher' *max_entity_left entity_left*	in 'The Iron Lady,' *max_entity_left* traces the winding staircase of fortune that transformed the younger daughter of a provincial English grocer into the greatest woman political leader since *entity_left*.

Table 2: Entities *highlighted* in answer candidates using two different methods. The example assumes that the entity "Young" is a maximal entity.

8 Empirical Evaluation

Having described our method of Question Classification, Entity Identification and Entity Highlighting, we next evaluate our method on the task of AS using different Highlighting methods and training data.

The training data commonly used for this task consists of two sets: The first consists of one hundred manually examined questions and corresponding answers candidates and the second, an automatically

generated set consisting of just over 1200 questions. We found the manually inspected, and hence higher quality test set to be too small for use in this task. However, we also found that the automatically generated training set contained several inconsistencies. To prevent noise, we discarded any questions with answer candidates that contained more than one false positive. We similarly discarded questions from the development set. Thus cleaned, we were left with 1164 questions in the training set and 76 and 60 questions in the Raw and Clean versions of the development set respectively. The development data is what the learning model is optimised on before the best performing model is used to evaluate the test data. To ensure that our results are comparable to those published by others, we make no changes to the test data.

Some question classes cannot have entities highlighted, as in the case of "Description" and "Definition". Some other question classes do not have Entity Identification implemented as we found it impossible to identify all possible elements of the class, as in the case of "Vehicles". We call such questions unhighlighted questions and the rest highlighted questions.

For each of the Clean and Raw versions of the data, we run the model on **a)** unhighlighted data, **b)** data highlighted using *bracketing*, **c)** data highlighted using *replacement*, **d)** unhighlighted data and highlighted data using *bracketing* combined, and **e)** unhighlighted data and highlighted data using *replacement* combined. In cases where we split the data into highlighted and unhighlighted sections, the results are combined to find the MRR and MAP scores of the complete data. The model run on data without entities highlighted (as presented in (Rao et al., 2016)) is the baseline. We also calculate the MRR and MAP scores for the baseline for *each* of these variations as we change the training data in each case. We use the same hyper-parameters as those provided in the implementation of the work by Rao et al. (2016). We include the highlighted versions of the training, development and test data for each of the variations above along with details of the hyper-parameters used, the trained models and the output as part of the supplemental material. We present our results in Table 3.

8.1 Result Analysis

The use of question classes embedded in candidate answers *outperforms the current state of the art in literature in every case except one*. This result (Number 10 in Table 3) is an anomaly that we attribute to over-fitting as we perform no hyper-parameter tuning. The strong performance of the baseline on the unhighlighted data (Sr. No. 1) is expected as answer candidates that are descriptive in nature (as is the case for questions belonging to the classes "Description", "Definition", etc., which also do not have entities identified) must necessarily have a larger overlap with the question. Once again, we ascribe the low performance of the corresponding unhighlighted baseline on the Clean Version (Sr. No. 8) to over-fitting.

The highlighting method of *replacement* performs better than *bracketing* except in the case of the anomaly. We believe this is because Named Entities, with their limited frequency, carry little information. Additionally, the replacement of entities that are phrases with a single frequently occurring word could improve sentence representation.

We expected the combination of the model independently trained on unhighlighted and highlighted data to perform better than that trained on the combination. This is not the case for either version of the test data. As in the case of the anomaly (Sr. No. 10), it is impossible to say if this is a result of over-fitting without performing hyper-parameter tuning on all ten of the models we present.

We also experimented with training different models for each of the coarse classes. We did this by extracting subsets of the training, development and test sets belonging to each of the coarse classes ("HUM", "LOC", ...), training the model on the training subset, optimised on the development subset, and testing it on the test subset. We found these results to be surprisingly poor, and believe this to be a result of deep learning models gaining more from increased data rather than homogeneous data. Homogeneity in data might, in fact, lead to overfitting and hence be detrimental. These results are consistent with results in work by Kyashif (2018) who used the QC system described in this work in task-based and common sense QA, where a similar subdivision led to poorer performance.

Data version		Sr. No.	Question Class	Highlight Method	Train #	Test #	Baseline MRR	Baseline MAP	This Work MRR	This Work MAP
RAW	This Work	1	unhighlighted	N.A.	515	13	(0.8065)	(0.8462)	N.A.	N.A.
		2	highlighted	bracketing	649	82	(0.7583)	(0.8179)	(0.8124)	(0.8304)
		3	highlighted	replacement	649	82	(0.7583)	(0.8179)	(0.8174)	(0.8293)
		4	Combining results from 1 & 2						0.8116	0.8326
		5	Combining results from 1 & 3						0.8262	0.8422
		6	combined	bracketing	1164	95	0.7783	0.8386	0.806	0.8316
		7	combined	replacement	1164	95	0.7783	0.8386	**0.8362**	**0.8625**
	Reported Baseline Performance (Rao et al., 2016)					95	0.764	0.827		
	Implementation Best					95	0.7904	0.8223		
	Prior state of the art (Rao et al., 2016)					95	*0.78*	*0.834*		
CLEAN	This Work	8	unhighlighted	N.A.	515	6	(0.6621)	(0.7222)	N.A.	N.A.
		9	highlighted	bracketing	649	62	(0.7449)	(0.7926)	(0.8354)	(0.8679)
		10	highlighted	replacement	649	62	(0.7449)	(0.7926)	(0.6991)	(0.8211)
		11	Combining results from 8 & 9						0.8201	0.855
		12	Combining results from 8 & 10						0.6958	0.8123
		13	combined	bracketing	1164	68	0.7713	0.8368	0.8324	0.862
		14	combined	replacement	1164	68	0.7713	0.8368	**0.8647**	**0.9039**
	Reported Baseline Performance (Rao et al., 2016)					68	0.762	0.854		
	Prior state of the art (Shen et al., 2017)					68	0.822	0.899		

Table 3: Results for each of the different highlighting methods on the Raw and Clean data versions. "Reported Perfomance" indicates the performance reported by Rao et al. (2016) when using the word-level model (Section 7) which is our baseline and the model that we adapt. "Implementation Best" represents the performance of the same model on the implementation that we use, and "Prior state of the art" represents the prior state of the art reported in literature for this dataset. Results in parentheses represent results on a subset of the test data.

As part of this work we release the following datasets[4]:

1. We release 3,500 of the total 5,500 training questions along with the 500 test questions originally released by Li and Roth (2002) manually updated with our classification.
2. We release the manually verified question classes for all 1500 questions in the AS task.
3. From the questions contained in the AS task, we release the list of entities identified for each answer candidate for the complete test set and for a highlighted training set of 649 questions.
4. We also make available an Application Programming Interface (API) to the modified Question Classification system we describe in this work[5].

8.2 Additional Advantages

Unlike other systems, a significant advantage of the system presented in this work is the fact that, not only does the system succeed in Answer Selection but in most cases *can also extract the specific factoid answer*, when one is present. When highlighting is possible and a maximal entity exists (as is the case in 60 of the 68 questions in the clean test set), such a maximal entity is nearly always the answer.

[4]Download from: `www.harishmadabushi.com/research/answer-selection/`
[5]API available at: `www.harishmadabushi.com/research/questionclassification/`

9 Error Analysis

One of the biggest source of errors tends to be incorrect entity tagging. This is especially so in the case of dates where we tag the date and the year independently thus allowing for the tagging of entities in candidate answer that only mention the year. Unfortunately, this often leads to incorrectly identified maximal entities when both the date and the year are present. In hindsight, we believe a better approach to dates would have been to combine sequential entities before entity matching, so as to capture exact dates, such as "13 October 1997" as a single entity rather than two different entities, one consisting of the day and month and the other of the year.

A similar source of errors occurs when processing the names of individuals, where common ways of shortening names cause errors in both the entity identification and the entity matching steps. For example, the ex-CEO of GE, Jack Welch, is referred to as GE-Welch and John Welch in some answer candidates.

When a candidate answer has too many entities of the required type, it is often misclassified. For example, the candidate answer "on its billions of kilometres flight toward Saturn, Cassini is scheduled to loop its path around Venus, the Earth and Jupiter to get the gravity boost needed for closing in onto its destination" for the question "What is Cassini's destination?", contains a large number of planetary objects diluting the signal generated by the entity. We believe the solution to this lies in increasing the granularity of the QC taxonomy.

There are some completely unexpected sources of errors as in the case of the question "Who established the Nobel Prize awards?", where all entities that contain the answer ("Alfred Nobel") are discarded as they are contained in the question. A more sophisticated method of establishing which entities are discarded could reduce errors of this kind.

Finally, there are some questions and answer candidates that are incorrectly tagged in the Answer Selection test set. For example, the question "What years did Sacajawea accompany Lewis and Clark on their expedition?" has, amongst its candidate sentences, the following two sentences: "the coin honors the young woman and teen-age mother who accompanied explorers meriwether lewis and william clark to the pacific ocean in 1805", and "in 1804 , toussaint was hired by lewis and clark , not for his own skills but for those of sacagawea." While the first candidate answer is marked as one containing the answer the second is not marked as such. While this is the only error of this kind, we found several other sentence candidates similarly marked as containing the answer when they, in fact, do not. Despite this, *we make no modifications to the test set*.

10 Conclusions and Future Work

We presented in this paper a method of Answer Selection that first required us to redefine a QC taxonomy provided by Li and Roth (2002), modify the Question Classification system developed by Tayyar Madabushi and Lee (2016) to match this modified taxonomy, create an entity identification method to extract entities belonging to those classes in our taxonomy, and finally, to use different methods of highlighting entities, so this information can be passed on to the Answer Selection model developed by Rao et al. (2016).

Our experiments show that Question Classes provide such a strong signal to the Deep Learning Model that we outperform current state of the art in all variations of our experiments except one. In the best configuration, our MRR and MAP scores outperform the current state of the art by between 3 and 5 points on both the Raw and Clean versions of the TrecQA Answer Selection test set.

We plan to test this method on different datasets including WikiQA and the Stanford Question Answering dataset, while also increasing the number of classes in our taxonomy and the number and kinds of entities identified by the our Entity Identification system.

The relatively small size of this dataset combined with the rather high accuracy achieved by current state of the art systems highlights the need for research in the field of AS through QC to move to other datasets. One challenge in doing so could be the lack of QA datasets with QC annotations. We hope that our release of the QC API as part of this work will help in this regard.

References

Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1987–1990, New York, NY, USA. ACM.

Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, San Diego, California, June. Association for Computational Linguistics.

Hua He, Kevin Gimpel, and Jimmy Lin. 2015a. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1586. Association for Computational Linguistics.

Hua He, Kevin Gimpel, and Jimmy Lin. 2015b. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1586, Lisbon, Portugal, September. Association for Computational Linguistics.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 1011–1019, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ulf Hermjakob. 2001. Parsing and question classification for question answering. In *Proceedings of the Workshop on Open-domain Question Answering - Volume 12*, ODQA '01, pages 1–6, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *CoRR*, abs/1506.06726.

Cody Kwok, Oren Etzioni, and Daniel S. Weld. 2001. Scaling question answering to the web. *ACM Trans. Inf. Syst.*, 19(3):242–262, July.

Denis Kyashif. 2018. Machine comprehension using commonsense knowledge - a dynamic memory network approach. `https://github.com/deniskyashif/sweet-reason`.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 233–242, New York, NY, USA. ACM.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November.

Amit Mishra and Sanjay Kumar Jain. 2016. A survey on question answering systems with classification. *Journal of King Saud University - Computer and Information Sciences*, 28(3):345 – 361.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Marco Pota, Massimo Esposito, and Giuseppe De Pietro, 2016. *A Forward-Selection Algorithm for SVM-Based Question Classification in Cognitive Systems*, pages 587–598. Springer International Publishing, Cham.

Vasin Punyakanok, Dan Roth, and Wen tau Yih. 2004. Natural language inference via dependency tree mapping: An application to question answering. In *IN SUBMISSION*.

Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1913–1916, New York, NY, USA. ACM.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 373–382, New York, NY, USA. ACM.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1179–1189. Association for Computational Linguistics.

Harish Tayyar Madabushi and Mark Lee. 2016. High accuracy rule-based question classification using question syntax and semantics. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1220–1230, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Chen-Tse Tsai, Wen-tau Yih, Chris J.C. Burges, and Scott Wen-tau Yih. 2015. Web-based question answering: Revisiting askmsr. Technical report, Microsoft Research, Redmond, Washington, April.

Nguyen Van-Tu and Le Anh-Cuong. 2016. Improving question classification by feature extraction and selection. *Indian Journal of Science and Technology*, 9(17).

Zhiguo Wang and Abraham Ittycheriah. 2015. Faq-based question answering via word alignment. *CoRR*, abs/1507.02628.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. In *EMNLP-CoNLL*.