Word Substitution in Short Answer Extraction: A WordNet-based Approach

Qingqing Cai, James Gung, Maochen Guan, Gerald Kurlandski, Adam Pease
IPsoft / New York, NY, USA
[qingqing.cai|james.gung|maochen.guan|gerald.kurlandski|adam.pease]@ipsoft.com

Abstract

We describe the implementation of a short answer extraction system. It consists of a simple sentence selection front-end and a two phase approach to answer extraction from a sentence. In the first phase sentence classification is performed with a classifier trained with the passive aggressive algorithm utilizing the UIUC dataset and taxonomy and a feature set including word vectors. This phase outperforms the current best published results on that dataset. In the second phase, a sieve algorithm consisting of a series of increasingly general extraction rules is applied, using WordNet to find word types aligned with the UIUC classifications determined in the first phase. Some very preliminary performance metrics are presented.

1 Introduction

Short Answer Extraction refers to a set of information retrieval techniques that retrieve a short answer to a question from a sentence. For example, if we have the following question and answer sentence

(1) Q: Who was the first president of the United States?
A: George Washington was the first president of the United States.

we want to extract just the phrase "George Washington". But what if we have a mismatch in language between question and answer? What is an appropriate measure for word similarity or substitution in question answering? If we have the question answer pair

(2) "Bob walks to the store."

(3) "Who ambles to the store?"

we probably want to answer "Bob", because "walk" and "amble" are similar and not inconsistent. In isolation, a human would likely judge "walk" and "amble" to be similar, and by many WordNet-based similarity measures they would be judged similar, since "walk" is found as WordNet synsets 201904930, 201912893, 201959776 and 201882170, and "amble" is 201918183, which is a direct hyponym of 201904930.

We can use Resnik's method (Resnik, 1995) to compute similarity. In particular we can use Ted Pedersen's (et al) implementation (Pedersen et al., 2004), which gives the result of `walk#n#4 amble#n#1 9.97400037941652` . Word2Vec (Mikolov et al., 2013a) using their 300-dimensional vectors trained on Google News, also gives a relatively high similarity score for the two words

```
> model.similarity('walk', 'amble')
0.525
```

2 Is Similarity the Right Measure?

But what about if we have

(4) "Bob has an apple."

(5) "Who has a pear?"

We find that this pair is even more similar than "walk" and "amble"

```
> model.similarity('apple', 'pear')
0.645
```

and from Resnik's algorithm

```
Concept #1: apple
Concept #2: pear
apple  pear
apple#n#1  pear#n#1  10.15
```

and yet clearly 4 is not a valid answer to 5. One possibility is that synset subsumption as a measure of word substitution (Kremer et al., 2014; Biemann, 2013)[1] [2] may be the appropriate metric,

[1] https://dkpro-similarity-asl.googlecode.com/files/TWSI2.zip
[2] http://www.anc.org/MASC/coinco.tgz

rather than word similarity.

3 Question Answering

Our approach starts with the user's question and the sentence that is most likely to contain the answer, which is selected with the BM25 algorithm (Jones et al., 2000). Then we identify the incoming question as a particular question type according to the UIUC taxonomy[3]. To this taxonomy we have added the yes/no question type. Then we pass the sentence and the question to a class written specifically to handle a particular UIUC question type. Generally, all the base question types behave differently from one another. Within a base question type, subtypes may be handled generically or with code specially targeted for that subtype. For this paper, we first discuss the approach to question classification, and then to answer extraction with a focus on the question subtypes that are amenable to a WordNet-based approach.

4 Question Classification

This section presents a question classifier with several novel semantic and syntactic features based on extraction of question foci. We use several sources of semantic information for representing features for each question focus. Our model uses a simple margin-based online algorithm. We achieve state-of-the-art performance on both fine-grained and coarse-grained question classification. As the focus of this paper is on WordNet, we leave many details to a future paper and primarily report the features used, the learning algorithm and results, without further justification

4.1 Introduction

Question analysis is a crucial step in many successful question answering systems. Determining the expected answer type for a question can significantly constrain the search space of potential answers. For example, if the expected answer type is *country*, a system can rule out all documents or sentences not containing mentions of countries. Furthermore, accurately choosing the expected answer type is extremely important for systems that use type-specific strategies for answer selection. A system might, for example, have a specific unit for handling *definition* questions or *reason* questions.

[3] http://cogcomp.cs.illinois.edu/Data/QA/QC/definition.html
http://cogcomp.cs.illinois.edu/Data/QA/QC/

In the last decade, many systems have been proposed for question classification (Li and Roth, 2006; Huang et al., 2008; Silva et al., 2011). Li and Roth (Li and Roth, 2002) introduced a two-layered taxonomy of questions along with a dataset of 6000 questions divided into a training set of 5000 and test set of 500. This dataset (henceforth referred to as the UIUC dataset) has since become a standard benchmark for question classification systems.

There have been a number of advances in word representation research. Turian et al. (Turian et al., 2010) demonstrated the usefulness of a number of different methods for representing words, including word embeddings and Brown clusters (Brown et al., 1992), within supervised NLP application such as named entity recognition and shallow parsing. Since then, largely due to advances in neural language models for learning word embeddings, such as WORD2VEC (Mikolov et al., 2013b), word vectors have become essential features in a number of NLP applications.

In this paper, we describe a new model for question classification that takes advantage of recent work in word embedding models, beating the previous state-of-the-art by a significant margin.

4.1.1 Question Focus Extraction

Question foci (also known as *headwords*) have been shown to be an important source of information for question analysis. Therefore, their accurate identification is a crucial component of question classifiers. Unlike past approaches using phrase-structure parses, we use rules based on a dependency parse to extract each focus.

We first extract the question word (how, what, when, where, which, who, whom, whose, or why) or imperative (name, tell, say, or give). This is done by naively choosing the first question word in the sentence, or first imperative word if no question word is found. This approach works well in practice, though a more advanced method may be beneficial in more general domains than the TREC (Voorhees, 1999) questions of the UIUC dataset.

We then define specific rules for each type of question word. For example, *what*/*which* questions are treated differently than *how* questions. In *how* questions, we identify words like *much* and *many* as question foci, while treating the heads of these words (e.g. *feet* or *people*) as a separate type known as **QUANTITY** (as opposed to **FOCUS**. Furthermore, when the focus of a *how* question

is itself the head (e.g. *how much did it cost?* or *how long did he swim?*), we again differentiate the type using a **MUCH** type and a **SPAN** type that includes words like *long* and *short*.

A head chunk such as *type of car* contains two words, *type* and *car*, which both provide potentially useful sources of information about the question type. We refer to words such as *type*, *kind*, and *brand* as **specifiers**. We extract the argument of a specifier (*car*) as well as the specifier itself (*type*) as question foci.

In addition to head words of the question word, we also extract question foci linked to the root of the question when the root verb is an **entailment** word such as *is*, *called*, *named*, or *known*. Thus, for questions like *What is the name of the tallest mountain in the world?*, we extract *name* and *mountain* as question foci. This can result in many question foci in the case of a sentence like *What **relative** of the **racoon** is sometimes known as the **cat-bear**?*

4.1.2 Learning Algorithm

We apply an in-house implementation of the multi-class Passive-Aggressive algorithm (Crammer et al., 2006) to learn our model's parameters. Specifically, we use PA-I, with

$$\tau_t = \min\left\{C, \frac{l_t}{\|x_t\|^2}\right\}$$

for $t = 1, 2, ...$ where C is the aggressiveness parameter, l_t is the loss, and $\|x_t\|^2$ is the squared norm of the feature vector for training example t. The Passive-Aggressive algorithm's name refers to its behavior: when the loss is 0, the parameters are unchanged, but when the loss is positive, the algorithm aggressively forces the loss to return to zero, regardless of step-size. τ (a Lagrange multiplier) is used to used to control the step-size. When C is increased, the algorithm has a more aggressive update.

4.2 Experiments

We replicate the evaluation framework used in (Li and Roth, 2006; Huang et al., 2008; Silva et al., 2011). We use the full, unaltered 5500-question training set from UIUC for training, and evaluate on the 500-question test.

To demonstrate the impact of our model's novel features, we performed a feature ablation test (Table 2) in which we removed groups of features from the full feature set.

Feature Set	**Fine**	**Coarse**
All	**92.0**	**96.2**
-clusters	90.2	96
-vectors	90	95.4
-clusters, vectors	89.8	95.2
-lists	88	94
-clusters, vectors, lists	86.2	92.8
-definition disambiguation	91	94.8
-quantity focus differentiation	90.2	96

Table 2: Feature ablation study: accuracies on coarse and fine-grained labels after removing specific features from the full feature set.

System	Fine	Coarse
Li and Roth 2002	84.2	91.0
Huang et al. 2008	89.2	93.4
Silva et al. 2011	90.8	95.0
Our System	**92.0**	**96.2**

Table 3: System comparison of accuracies for fine (50-class) and coarse (6-class) question labels.

4.3 Discussion

Our model significantly outperforms all previous results for question classification on the UIUC dataset (Table 3). Furthermore, we accomplished this without significant manual feature engineering or rule-writing, using a simple online-learning algorithm to determine the appropriate weights.

5 Answer Extraction

In this section we discuss techniques for short answer extraction once questions have been classified into a particular UIUC type. We employ a "sieve" approach, as in (Lee et al., 2011), that has seen some success in tasks like coreference resolution and is creating a bit of a renaissance in rule-based, as opposed to machine learning, approaches in NLP. We provide in this paper one example of how instead of taking an either/or approach, both methods can be combined into a high performance system. We focus below on the sieves that are specific to question types where we have been able to profitably employ WordNet for finding the right short answer. Preliminary results have been positive employing this approach.

We have two strategies that are used across the base question types: employing semantic role labels and recognizing appositives.

Feature Type	guitar	Cup
Lemma	guitar	cup
Shape	x+	Xx+
Authority List	instrument	sport
Word Vector*	vocals, guitars, bass, harmonica, drums	champions, championship, tournament
Brown Cluster Prefix	0010, 001010, 0010101100, ...	0111, 011101, 0111011000, ...

Table 1: Features used for head words. Each dimension of the corresponding word vector was used as a real-valued feature. *Nearest neighbors of the corresponding word vector are shown.

5.1 Corpus

Our current testing corpus consists of three parts. The first is an open source Q&A test set developed at Carnegie Mellon University (Smith et al., 2008)[4] consisting of roughly 1000 question and answer pairs on Wikipedia articles. The second is a proprietary Q&A test set developed at IPsoft consisting of a growing set of question answer pairs currently numbering roughly 2000 pairs and conducted on short sections of Wikipedia articles. The third test set is TREC-8 (Voorhees, 1999).

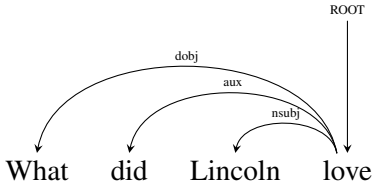
5.2 Semantic Role Labels

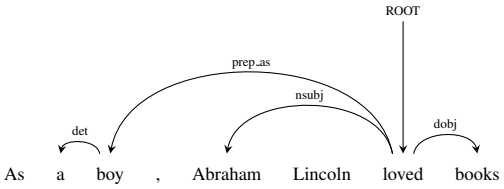
We employ the semantic role labeling of ClearNLP (Choi, 2012)[5]. While the labels are consistent with PropBank (Palmer et al., 2005), ClearNLP fixes the definition of several of the labels (A2-A5) that are left undefined in PropBank. A0 is the "Agent" relation, which is often the subject of the sentence. A1 is the "Patient" or object of the sentence. The remainder can be found in (Choi, 2012).

Let's look at an example and the list the steps followed in the code to analyse the question and answer.

(6) Q: What did Lincoln love?
A: As a boy, Abraham Lincoln loved books.

We have the following dependency graphs among the tokens in each sentence:

(7)


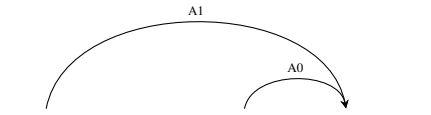
(8)


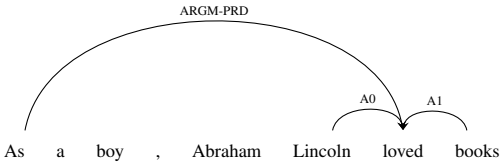
and part of speech labels

(9) What did Lincoln love?
WP VBD NNP VB

(10) As a boy, Abraham Lincoln loved books.
IN DT NN NNP NNP VBD NNS

and semantic role labels

(11)


(12)


1. We collect basic information from the question and answer sentence
   (a) find the question word, e.g. "what", "when", "where", etc. In Example 6 it is "what-1"
   (b) Locate the verb node nearest to the question word. In Example 6 it is "love-4"
   (c) Find the semantic relations in the question. We find an Agent/A0 relationship

[4] download from http://www.cs.cmu.edu/~ark/QA-data/

[5] http://www.clearnlp.com

between Lincoln-3 and the verb love-4. We find a Patient/A1 relationship between the question word What-1 and the verb love-4. (See Examples 11 and 12).

(d) Find semantic relations in the answer sentence. We find an Agent/A0 relationship between Lincoln-6 and the verb loved-7. We find an ARGM-PRD relationship between As-1 and the verb loved-7. We find a Patient/A1 relationship between books-8 and the verb loved-7. (See Examples 11 and 12).

(e) Perform a graph structure match between the question and answer graphs formed by the set of their semantic role labels. Find the parent graph node in the answer that matches as many nodes in the question as possible. In our example, loved-7 is the best match. (See Examples 11 and 12).

2. Collect and score candidate answer nodes. Score each semantic child for best parent found in the previous step, based on part of speech, named entity, dependency relations from Stanford's CoreNLP (Manning et al., 2014), and semantic role label information. We initialize each child to a value of 1.0 and then penalize it by 0.01 for the presence of any out of a set of possible undesirable features, as follows:

   - The candidate's semantic role label starts with "ARGM", meaning that its semantic role is something other than A0-A5. (See Examples 11 and 12). Note that this is only applied in cases where the question type has been identified as "Human" or "Entity"
   - The node's dependency label = "prep*" indicating that it is a prepositional relationship. Note that this is only applied in cases where the question type has been identified as "Human" or "Entity"
   - If the candidate node is the same form (word spelling) as in the question, or its WordNet hyponym
   - If the candidate node is the same root (lemma) as in the question, or its WordNet hyponym
   - If the candidate node is lower case. Note that this is only applied in cases where the question type has been identified as "Human" or "Entity"
   - If the candidate node has a child with a different semantic role label than in the question
   - If the candidate node is an adverb or a Wh- quantifier as marked by its part of speech label

3. Pick the dependency node with highest confidence score as the answer node. In our example we have As-1 = 0.97, Lincoln-6 = 0.96 and books-8 = 0.99.

Note that the step of scoring the answer nodes enumerates a small feature set with hand-set coefficients. We expect in a future phase to enumerate a much larger set of features, and then set the coefficients based on machine learning over our corpus of question-answer pairs. One simple experiment to show the value of semantic role labeling was conducted on a portion of our testing corpus. Using semantic role labels we achieved total of 638 correct answers out of 1460 questions (which was the total number in the IPsoft internal Q&A test set at the time of the test), for a correctness score of 43.7%. Without semantic role labels the result was 462 out of 1460, or 31.6%.

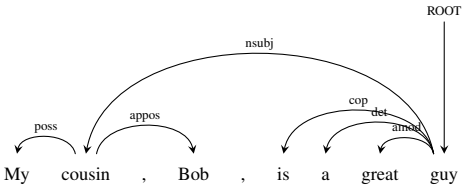
5.3 Appositives

The appositive is a grammatical construction in which one phrase elaborates or restricts another. For example,

(13) My cousin, Bob, is a great guy.

"Bob" further restricts the identity of "My cousin".

(14)



We use the appositive grammatical relation to identify the answers to "What" questions.

5.4 Entity Question Type

Short answer extraction for the Entity question type has some specialized rules for some subtypes, and some rules which are applied generally to all

the other subtypes. We are also exploring using WordNet (Fellbaum, 1998) synsets to get word lists that are members of each Entity subtype (see Table 4). This appears to have a significant effect, since 10 questions are answerable with this approach just addressing two of the 22 Entity subtypes. More work is needed to get comprehensive statistics.

5.4.1 Entity.animal Subtype

1. First try to find an appositive relationship. If there is one, use it as the answer. For example 14, if we ask "Who is a great guy?" we have a simple answer with "Bob" as the appositive. If that fails:

2. try the approach described above in subsection 5.2 and keep the candidate with the highest confidence score

5.4.2 Entity.creative Subtype

1. First try to find an appositive relationship. If there is one, use it as the answer. If that fails:

2. try the approach described above in subsection 5.2 and keep the candidate with the highest confidence score. If that fails:

3. find the first capitalized sequence of words and return it

5.4.3 All Other Entity Subtypes

1. First try to find an appositive relationship. If there is one, use it as the answer. If that fails:

2. try the approach described above in subsection 5.2 and keep the candidate with the highest confidence score

5.5 Example

Take for example the following

(15) Q: What shrubs can be planted that will be safe from deer?
A: Three old-time charmers make the list of shrubs unpalatable to deer: lilac, potentilla, and spiraea. Short Answer: Lilac, potentilla, and spiraea.

Knowing from WordNet that 112310349:{lilac}, and 112659356:{spiraea, spirea} (although not potentilla) are hyponyms of shrub makes it easy to find the right dependency parse subtree for the short answer.

Similarly for

(16) Q: What athletic game did dentist William Beers write a standard book of rules for?
A: In 1860, Beers began to codify the first written rules of the modern game of lacrosse. Short Answer: Lacrosse.

knowing that 100455599:{game} is a hypernym of 100477392:{lacrosse} makes finding the right answer in the sentence easy.

6 UIUC Question Types and Synsets

Table 4 lists all the types and subtypes in the UIUC taxonomy and the WordNet (Fellbaum, 1998) synset numbers that correspond to semantic types for the UIUC types. These are used to get all words that are in the given synsets as well as all words in the synsets that are more specific in the WordNet hyponym hierarchy than those listed. Note that below we prepend to the synset numbers a number for their part of speech. In the current scheme all are nouns, so the first number is always a "1". We only elaborate subtypes of Entity, Human, and Location as the other categories do not use WordNet for matching.

7 Conclusion

Using a WordNet-based word replacement method appears to be better for question answering than using word similarity metrics. In preliminary tests 10 questions in a portion of our corpora are answerable with this approach just addressing two of the 22 Entity subtypes with WordNet based matching. While more experimentation is needed, the results are intuitive and promising. The current approach should be validated and compared against other approaches on current data sets such as (Peñas et al., 2015).

Class	Definition	Synsets
ABBREVIATION	abbreviation	
ENTITY	entities	
animal	animals	100015388
body	organs of body	105297523
color	colors	104956594
creative	inventions, books and other creative pieces	102870092, 103217458, 103129123
currency	currency names	113385913, 113604718
dis.med.	diseases and medicine	114034177, 114778436
event	events	100029378
food	food	100021265
instrument	musical instrument	103800933
lang	languages	106282651
letter	letters like a-z	
other	other entities	
plant	plants	100017222
product	products	100021939
religion	religions	108081668, 105946687
sport	sports	100433216, 100523513, 103414162
substance	elements and substances	100020090
symbol	symbols and signs	
technique	techniques and methods	
term	equivalent terms	
vehicle	vehicles	103100490
word	words with a special property	
DESCRIPTION	description and abstract concepts	
HUMAN	human beings	
group	a group or organization of persons	107950920
ind	an individual	102472293
title	title of a person	
description	description of a person	
LOCATION	locations	
city	cities	108226335, 108524735
country	countries	108168978
mountain	mountains	109359803, 109403734
other	other locations	108630039
state	states	108654360
NUMERIC	numeric values	

Table 4: UIUC class to WordNet synset mappings

References

Chris Biemann. 2013. Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources and Evaluation*, 47(1):97–122.

Peter Brown, Peter Desouza, Robert Mercer, Vincent dellaPietra, and Jenifer Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Jinho D. Choi. 2012. *Optimization of Natural Language Processing Components for Robustness and Scalability*. Ph.D. thesis, University of Colorado at Boulder, Boulder, CO, USA. AAI3549172.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Zhiheng Huang, Marcus Thint, and Zengchang Qin. 2008. Question classification using head words and their hypernyms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 927–936. Association for Computational Linguistics.

K. Sparck Jones, S. Walker, and S.E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments: Part 1. *Information Processing & Management*, 36(6):779 – 808.

Gerhard Kremer, Katrin Erk, Sebastian Pad, and Stefan Thater. 2014. What Substitutes Tell Us – Analysis of an "All-Words" Lexical Substitution Corpus. In *Proceedings of EACL*, Gothenburg, Sweden.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's Multi-pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 28–34, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Xin Li and Dan Roth. 2006. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(03):229–249.

Chris Manning, John Bauer, Mihai Surdeanu, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*. Now Pub.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity: Measuring the Relatedness of Concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anselmo Peñas, Christina Unger, Georgios Paliouras, and Ioannis A. Kakadiaris. 2015. Overview of the CLEF question answering track 2015. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, pages 539–544.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95*, pages 448–453. Morgan Kaufmann.

Joao Silva, Luísa Coheur, Ana Cristina Mendes, and Andreas Wichert. 2011. From symbolic to subsymbolic information in question classification. *Artificial Intelligence Review*, 35(2):137–154.

Noah A. Smith, Michael Heilman, , and Rebecca Hwa. 2008. Question Generation as a Competitive Undergraduate Course Project. In *NSF Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, VA, September.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Ellen M. Voorhees. 1999. Overview of the TREC 2002 Question Answering Track. In *In Proceedings of the 11th Text Retrieval Conference (TREC)*, pages 115–123.