Using the Web as Corpus for Un-supervised Learning in Question Answering

Yi-Chia Wang[1], Jian-Cheng Wu[2], Tyne Liang[1] and Jason S. Chang[2]

1. Dep. of Computer and Information Science, National Chiao Tung University, Taiwan, R.O.C.
2. Dep. of Computer Science, National Tsing Hua University, Taiwan, R.O.C.

rhyme.cis92g@nctu.edu.tw

Abstract. In this paper we propose a method for unsupervised learning of relation between terms in questions and answer passages by using the Web as corpus. The method involves automatic acquisition of relevant answer passages from the Web for a set of questions and answers, as well as alignment of wh-phrases and keywords in questions with phrases in the answer passages. At run time, wh-phrases and keywords are transformed to a sequence of expanded query terms in order to bias the underlying search engine to give higher rank to relevant passages. Evaluation on a set of questions shows that our prototype improves the performance of a question answering system by increasing the precision rate of top ranking passages returned by the search engine.

1. Introduction

It was noted that people have submitted longer and longer queries to the Web search engines. Recently, users have started to submit natural language queries instead of a list of keywords. It has encouraged many researchers to develop question answering systems which specifically aim at natural language questions, such as AskJeeves (www.ask.com) and START (www.ai.mit.edu/projects/infolab/).

For typical question answering systems, document/passage retrieval is the most significant subtask. In this step, the QA system breaks a natural language question into a set of keywords, uses keywords to query a search engine, and returns documents or messages that are related to the queries for further processing. However, the keywords in questions usually are not very effective in retrieving relevant passages. Consider the question "*Who invented glasses with two foci?*" Typically, we will send the keywords "*invented glasses two foci*" to a search engine to retrieve documents or passages. Submitting such keywords to AltaVista, we got irrelevant information about astronomy or physics rather than the inventor "*Benjamin Franklin*" of bifocal glasses. Intuitively, if we include the phrase "*inventor of*" or "*bifocal*" in the query sent to the search engine (*SE*), we are likely to retrieve passages with the answer.

We present the system *Atlas* (Automatic Transform Learning by Aligning Sentences of question and answer), which automatically learns the transforms from wh-phrases and keywords to n-grams in relevant passages by using the Web as corpus. The transformed query should be more likely to retrieve passages that contain the answer. For instance, consider the natural language question "*Who invented the light bulb?*" Using the keywords in the question directly, we end up with the keyword query, "*invented light bulb*," for a search engine such as Google. We observed that such a query has room for improvement in terms of bringing in more instances of the relevant answer. Our experiment indicates that the proposed method will determine the best transforms for the wh-phrase "*who invented*" including "*inventor of*", "*was invented*", and "*invented by*". On the other hand, the best transforms discovered for the keyword "*bulb*" include "*light bulb*" and "*electric light*." Intuitively, these transforms used together will convert the question into an expanded query for Google, "*("was invented" || "invented by") ("electric light" || "light bulb")*" which is more effective in retrieving relevant sentences in the top ranking summaries returned by the search engine, such as "*The light bulb was invented by an illuminated scientist called Thomas Edison in 1879!*". One indicator of effective query is the precision rate at *R* document retrieved (P_R), the percentage of first *R* top ranking Web pages (or summaries) which contain the answer. Another indicator is the mean reciprocal rank (MRR) of the first relevant document (or summary) returned. If the *r*-th document (summary) returned is the first one with the answer then the reciprocal rank is 1/*r*. Our goal in this study is exploration of methods that will automatically learn the transforms that convert natural language questions to queries with high average P_R or MRR.

The rest of the paper is organized as follows. In Section 2, we survey the related work. In Section 3, we describe our method for unsupervised learning of transforms for question and answer pairs which are automatically acquired from the Web and how we use the aligned result for effective query expansion in the QA

system. The experiment and evaluation results are given in Section 4. In the last section, we conclude with discussion and future work.

2. Related Work

Extensive work on question answering has been reported in the many literature (Buchholz et al., 2001; Harabagiu et al., 2001; John et al., 2002; Shen et al., 2003). In this study, we focus on learning the transforms that can be used to convert questions into effective queries in order to retrieve relevant passages.

Hovy et al. (2000) utilized hypernyms and synonyms in WordNet to expand queries for increasing recall. However, blindly expanding a word to its synonyms sometimes causes undesirable effects. As for hypernyms, it is difficult to determine how many hypernyms a word should be expanded. In contrast to this approach, our method learns query transforms specific to a word or phrase based on real-life questions and answer passages.

In a recent study most closely related to our method, Agichtein et al. (2004) described the *Tritus* system that learns transforms of wh-phrases such as "*what is*" to "*refers to*" by using FAQ data automatically. Our method learns transforms for wh-phrases as well as keywords from the web. Tritus system uses heuristic rules and thresholds for term and document frequency to learn transforms, while we rely on a mathematical model method for statistical machine translation. Shen, Lin and Chen (2003) proposed a method that is similar to the *Tritus* system for the why question.

Recently, Echihabi and Marcu (2003) presented a noisy channel approach to question answering. Their method also involves collecting answer passages from the web and aligning words across a question and relevant answer passages. However, they require full parsing of the sentences and complicated decision of making a "cut" in the parse tree to determine whether to align word, syntactic, or semantic categories. Our simple method is also based on alignment but it does not require full parsing and perform alignment at the surface levels of words and n-grams.

In contrast to previous work on query expansion for question answering, we propose a method that learns query transforms for all phrases in a natural language question automatically on the Web.

3. Method for Learning Question to Query Transforms

In this section, we present an unsupervised method for QA which automatically learns transforms from wh-phrases and keywords to answer n-grams by using the Web as corpus.

3.1 Problem Statement

Given a set of natural language questions *Q*s and answer terms *A*s, we obtain a collection of passages that contain the answer *A* to the question *Q* via some search engine *SE*. From the collection of answer passages *AP*s, our goal is to discover a set of transforms *T* that can be applied to wh-phrases and keywords in *Q* in the hope that the transformed queries are more effective in retrieving passages containing *A*.

3.2 Procedure for Learning Transforms

This subsection illustrates the procedure for learning transforms *T* from wh-phrases and unigrams in *Q* into bigrams in *AP*. The reason why we decide to use bigrams in *AP* is that bigram contains more information than unigram and is more effective in retrieving relevant passages. On the other hand, we break *Q*s into unigrams following the standard approach in IR.

(1)	Automatically collect pairs of *Q* and *AP* from the Web for training. (Section 3.2.1)
(2)	Select frequent wh-phrases. (Section 3.2.2)
(3)	Apply the alignment technique to the collected material. (Section 3.2.3)

Fig.1. Procedure for learning transforms

3.2.1 Collecting Training Material from the Web

In the first step of the learning process (see Figure 1), we retrieve a set of (*Q*, *A*, *AP*) pairs from the Web for training purpose where *Q* stands for a natural language question, and *AP* is a passage containing keywords in *Q* and the answer term *A*. The data gathering process is described as follows:

1. For each (*Q, A*) pair in the given collection, we extract keywords *K* of *Q*, say, $k_1, k_2, \ldots, k_n$.
2. Submit ($k_1, k_2, \ldots, k_n, A$), as a query to *SE*.
3. Download the top *M* summaries that are returned by *SE*.
4. Retain only those summaries containing *A*. See Table 1 for details.

Table 1. An example of converting a question (*Q*) with its answer (*A*) to *SE* query and retrieving answer passages (*AP*)

(*Q*, *A*)	***AP***
What is the capital of Pakistan? Answer:(*Islamabad*)	Bungalow For Rent in *Islamabad*, Capital Pakistan. Beautiful Big House For …
($k_1, k_2, \ldots, k_n, A$)	*Islamabad* is the capital of Pakistan. Current time, …
capital, Pakistan, Islamabad	…the airport which serves Pakistan's capital *Islamabad*, …

3.2.2 Selecting Frequent Wh-phrases

In the second step, we produce a set of high frequency phrases that characterize different question categories. We follow the method proposed by Agichtein et al. (2004). The method simply involves computing the frequency of all n-grams in *Q*s and filters out those with small counts. We will treat the wh-phrases (*QP*s) as a token in the subsequent steps. However, we differ from their approach in that we are not limited to n-grams of function words. For instance, we derived "*in what year*", "*who wrote*", etc. More examples of wh-phrases are listed in Table 2.

Table 2. An example of wh-phrases that are used

Wh-words	**Wh-phrases *QP*s**
What	"what is the", "in what year", "what was", …
Who	"who was the", "who wrote", …
Which	"which country", "with which", …
⋮	⋮

3.2.3 Learning Question to Query Transforms

In the third step, we use word alignment techniques originally developed for statistical machine translation to find out relation between wh-phrases or keywords in *Q* and n-grams in *AP*. We use the Competitive Linking Algorithm proposed by Melamed (1997) to align (*Q*, *AP*) pair. We proceed as follows:

1. Perform Part of Speech (POS) tagging on both *Q* and *AP* in the collection. (See Table 3 and 4)
2. Replace all instances of *A* with the tag <ANS> in *AP*. For example, the answer "*Islamabad*" in *AP* for the question "*What is the capital of Pakistan?*" is replaced with <ANS>. (See Table 4.) The purpose of <ANS> is to avoid data sparseness while counting bigrams in the following step.
3. Segment *Q* into unigrams or *QP*s and eliminate unigrams with low counts. We denote the remaining unigrams as $q_1, q_2, ..., q_n$. (See Table 5)
4. Segment *AP* into bigrams and eliminate bigrams with small term frequency (*tf*) or very large document frequency(*df*). We denote the remaining bigrams $a_1, a_2, ..., a_m$. (See Table 6)
5. For all *i*, *j*, calculate log likelihood ratio (LLR) of q_i and a_j. (See Table 7)
6. Eliminate candidates with a LLR value lower than 7.88. (See Table 8)
7. Sort list of (q_i, a_j) by decreasing LLR value. (See Table 8)
8. Go down the list and select a pair if it does not conflict with previous selection.
9. Stop when running out of pairs in the list.
10. Produce the list of aligned pairs for all *Q*s and *AP*s.
11. Select top *N* bigrams, $a_1, a_2, ..., a_r$, for every wh-phrase or unigram q_i in alignment pairs. (See Table 9)

Table 3. Part of Speech of *Q*

***Q* word**	**Lemma**	**Position**	**POS**
What is the	what be the	1	*
capital	capital	2	nn
of	of	3	in
Pakistan	Pakistan	4	np
?	?	5	.

Table 4. Part of Speech of *AP*

***AP* word**	**Lemma**	**Position**	**POS**
Most	most	1	rbt
of	of	2	in
Pakistan	Pakistan	3	np
rainfall	rainfall	4	nn
is	be	5	bez
scarce	scarce	6	jj
.	.	7	.
Islamabad	<ANS>	8	np
,	,	9	,
the	the	10	at
capital	capital	11	nn
of	of	12	in
Pakistan	Pakistan	13	np
since	since	14	in
1963	1963	15	cd
,	,	16	,
and	and	17	cc
Rawalpindi	Rawalpindi	18	np
,	,	19	,
are	be	20	ber
both	both	21	abx
located	locate	22	vbn
on	on	23	rp
the	the	24	at
Pothowar	Pothowar	25	np
Plain	Plain	26	nn

Table 5. Wh-phrases and unigrams in *Q*

Question words	Number of occurrence
what is the	14,571
capital	7,513
of	29,673
Pakistan	135

Table 6. Bigrams in *AP*

N-gram in AP	Number of occurrence
most of	368
of Pakistan	54
Pakistan rainfall	1
rainfall be	4
be scarce	2
scarce .	1
. <ANS>	8574
<ANS> ,	16665
, the	10227
the capital	1690
capital of	1669
of Pakistan	54
Pakistan since	1
since 1963	3
1963 ,	58
, and	9994
and Rawalpindi	3
Rawalpindi ,	5
, be	3718
be both	77
both locate	2
locate on	174
on the	4868
the Pothowar	2
Pothowar Plain	2

Note: The entries in the shaded area are eliminated for their low counts

Table 7. Combination of q_i and a_j

q_i	a_j	Number of co-occurrence	LLR
what is the	most of	82	754.72
capital	most of	34	293.2
Of	most of	127	1118.59
Pakistan	most of	1	1.27
what is the	of Pakistan	47	614.3
Of	of Pakistan	49	580.78
capital	of Pakistan	43	602.61
Pakistan	of Pakistan	44	990.37
…	…	…	…

Table 8. Alignment results

q_i	a_j	LLR
of	, and	27227.9
capital	capital of	21194.2
what is the	, is	7443.56
Pakistan	of Pakistan	990.37

Table 9. Examples of transforms selected from alignment results for N=3

Wh-phrase and Keyword in *Q*	Bigram in *AP*	Alignment counts
what is the	is the	1,254
what is the	in the	503
what is the	, the	242
capital	capital of	545
capital	capital city	241
capital	state capital	236
Pakistan	Pakistan ,	39
Pakistan	of Pakistan	21
Pakistan	in Pakistan	6

3.3 Runtime Transformation of Questions

At run time, *Q* is broken into wh-phrases and keywords which are converted to a sequence of query terms according to transforms based on the alignment results described in Section 3.2 in order to give higher ranks to passages that contain the answer for specific *SE*. See Table 10 for example of the conversion process of the question "*Who invented light bulb?*"

Table 10. An example of transformation from question into query

Question		
Who invented light bulb?		
Wh-phrase	**Keywords**	
Who invented	light	bulb
Transform wh-phrase and keywords		
was invented invented by	electric light light bulb	electric light light bulb
Expanded query		
Boolean query: ((was invented)OR(invented by))AND((electric light)OR(light bulb))		
Equivalent Google query: ("was invented" || "invented by") ("electric light" || "light bulb")		

4. Experiments and Evaluation

4.1 Training Data Set

Our data training data set were collected from http://www.quiz-zone.co.uk.We use 3,581 distinct (*Q*, *A*) pairs for automatically retrieving *AP* from the search engine Google. For each *Q*, top 100 summaries returned by Google are downloaded. See Table 11 for details of the training corpus.

Table 11. The training corpus

Training data set	Distinct (*Q*, *A*)	Distinct (*Q*, *AP*)
Quiz-Zone	3,581	99,697

4.2 Alignment Results

We choose the top 2 (N=2) bigrams for each *QP* or keyword in alignment results. Table 12 lists examples of *QP* or keyword and its two corresponding transformed bigrams.

Table 12. Parts of alignment results

QP* or Keyword in *Q	**Bigram in *AP***	**Alignment count**
invent	be invent	175
invent	invent by	43
who wrote	be bear	94
who wrote	he write	87
capital	capital of	545
capital	capital city	241

4.3 Evaluation Results

We used a test set of ten questions which are set aside from the training corpus. Table 13 shows the keyword queries and the expanded queries based on the transforms learned from the Web. We evaluated the expanded query by the mean reciprocal rank (MRR) and the precision rate at ten summaries returned by Google. For comparison, we also evaluated Google without applying query transforms. During experiment, the ten batches of returned summaries for the ten questions were evaluated by two human judges. As we can see in Table 14, using keywords from the natural language questions directly to query Google resulted in an MMR value of 0.48. However, when using expanded queries provided by the Atlas system, we had an MMR of 0.70, a statistically significant improvement. The average precision rate was improved slightly from 40% to 47%. The experimental results show that the Atlas system used in conjunction with the search engine Google outperforms the underlying search engine itself.

Table 13. Test questions

Q	**Keyword query for Google (GO)**	**Expanded query for Google (AT+GO)**
What is the capital of Pakistan?	capital Pakistan	("capital +of" || "capital city") Pakistan
What became the 50th state of the America?	became 50th state America	("+to become" || "leader +of") "50th state" "United State"
Who had a hit in 1994 with "Zombie"?	hit 1994 "Zombie"	("number one" || "hit +in") 1994 "Zombie"
In which year did Coronation Street begin?	year Coronation Street begin	("was found" ||"was born") Coronation Street ("began +in" || "began on")
In "The Simpsons", what is the name of Ned Flanders wife?	"The Simpsons" name Ned Flanders wife	"The Simpsons" ("name +is" || "name +of") Ned Flanders wife
In mythology, who supported the heavens on his shoulders?	Mythology supported heavens shoulders	"+in Greek" "+of +his" supported heavens shoulders
Which Saint's day is on March 1st?	Saint day March 1st	"+is +a" Saint "St" March 1st
What is the largest city in Switzerland?	largest city Switzerland	("largest country" || "second largest") Switzerland
Who directed the Oscar-winning film "The English Patient"?	directed Oscar-winning film "The English Patient"	("directed +by" || "+and directed") Oscar-winning film "The English Patient"
Which country was once ruled by Tsars?	country once ruled Tsars	("country +is" || "country +in") "ruled +by" Tsars

Table 14. Evaluation results

Performances	MRR	Precision (%)
AT+GO (*Atlas* expanded query for Google)	0.70	47
GO (Direct keyword query for Google)	0.48	40

5. Conclusions and Future Work

We show that our method clearly provide means for learning transformation from a natural language question to a query by applying statistical word alignment technique. The method involves automatically acquiring relevant passages from the Web for a set of questions and answers, aligning phases across from questions to answer passages in order to create phrase transforms that involve wh-words as well as content words. Evaluation on a set of questions shows that our prototype in conjunction with a search engine outperforms the underlying search engine used alone.

Many future directions present themselves. For example, the patterns learned from answer passages acquired on the Web can be extended to include longer and more effective n-grams to further booster the MMR value or average precision rate. Additionally, an interesting direction to explore is creating phrase transforms that contain the answer extraction patterns. These answer extraction patterns can be learned for different types of answers. Yet another direction of research would be to provide confidence factors for ranking the likelihood of many candidate answers extracted using patterns.

In summary, we have introduced a method for learning query transforms that improves the ability to retrieve passages with answers using the Web as corpus. The method involves finding query transformations based on techniques borrowed from training a noisy channel in machine translation study. We have implemented and thoroughly evaluated the method as applied to a set of more than 4,000 questions. We have shown that the method can be used with a search engine as an effective component in a question answering system.

References

[1] Abdessamad Echihabi, Daniel Marcu. A Noisy-Channel Approach to Question Answering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp.16-23, July 2003.

[2] Buchholz, Sabine. Using grammatical relations, answer frequencies and the World Wide Web for question answering. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

[3] Eugene Agichtein, Steve Lawrence, Luis Gravano. Learning to find answers to questions on the Web. In *ACM Transactions on Internet Technology (TOIT)*, Volume 4, Issue 2, pp.129-162, 2004.

[4] Harabagiu, S., D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Buneascu, R. Gîrju, V. Rus and P. Morarescu. FALCON: Boosting Knowledge for Answer Engines. In *Proceedings of the 9th Text Retrieval Conference (TREC-9)*, pp.479–488.

[5] Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, CY. Question answering in Webclopedia. In *Proceedings of the TREC-9 Question Answering Track*, pp.655-672, 2000.

[6] John M. Prager, Jennifer Chu-Carroll, Krysztof Czuba. Use of WordNet Hypernyms for Answering What-Is Question. In *Proceedings of the TREC-2002 Conference (TREC 2002)*.

[7] Melamed, I. Dan. A Word-to-Word Model of Translational Equivalence. In *Proceedings of the 35st Annual Meeting of the Association for Computational Linguistics*, pp.490-497, 1997.

[8] **沈天佐, 林川傑, 陳信希. 以網際網路內容為基礎之問答系統 "Why" 問句研究**. In *Proceedings of Rocling 2003*, pp.211-229, 2003.