Finding What Matters in Questions

Xiaoqiang Luo, Hema Raghavan, Vittorio Castelli, Sameer Maskey and Radu Florian
IBM T.J. Watson Research Center
1101 Kitchawan Road, Yorktown Heights, NY 10598
{xiaoluo,hraghav,vittorio,smaskey,raduf}@us.ibm.com

Abstract

In natural language question answering (QA) systems, questions often contain terms and phrases that are critically important for retrieving or finding answers from documents. We present a learnable system that can extract and rank these terms and phrases (dubbed *mandatory matching phrases* or MMPs), and demonstrate their utility in a QA system on Internet discussion forum data sets. The system relies on deep syntactic and semantic analysis of questions only and is independent of relevant documents. Our proposed model can predict MMPs with high accuracy. When used in a QA system features derived from the MMP model improve performance significantly over a state-of-the-art baseline. The final QA system was the best performing system in the DARPA BOLT-IR evaluation.

1 Introduction

In most question answering (QA) systems and search engines term-weights are assigned in a context independent fashion using simple TF-IDF like models (Robertson and Walker, 1994; Ponte and Croft, 1998). Even the more recent advances in information retrieval techniques for query term weighting (Bendersky et al., 2010; Bendersky, 2011) typically rely on bag-of-words models and corpus statistics, such as inverse-document-frequency (IDF), to assign weights to terms in questions. While such solutions may work for keyword queries of the type common on search engines such as Google, they do not exploit syntactic and semantic information when it comes to well formed natural language questions. In this paper we propose a new model that identifies important terms and phrases in a natural language question, providing better query analysis that ultimately leads to significant improvements in a QA system.

To motivate the work presented here, consider the query "How does one apply for a New York day care license?". A bag-of-words model would likely assign a high score to "**New licenses** for **day care** centers in **York** county, PA" because of high word overlap, but it does not answer the question, and also the state is wrong. A matching component that uses the phrases "New York," "day care," and "license" is likely to do better. However, a better matching component will understand that *in the context of this query* all three phrases "New York," "day care" and "license" are important, and that "New York" needs to modify "day care." A snippet that does not *contain*[1] these important phrases, is unlikely an answer. We call these important phrases *mandatory matching phrases* (MMPs).

In this paper, we explore deep syntactic and semantic analyses of questions to determine and rank MMPs. Unlike existing work (Zhao and Callan, 2010; Bendersky et al., 2010; Bendersky, 2011), where term/concept weights are learned from a set of questions and judged documents based on *corpus-based statistics*, we annotate *questions* and build a trainable system to select and score MMPs. This model relies heavily on existing syntactic parsers and semantic-oriented named-entity recognizers, but does not need question answer pairs. This is espe-

[1] "contain" here means semantic equivalence or entailment, not necessarily the exact words or phrases.

Proceedings of NAACL-HLT 2013, pages 878–887,
Atlanta, Georgia, 9–14 June 2013.

cially attractive at the initial system-building stage when no or little answer data is available.

The main contributions of this paper are: firstly, we propose a framework to select and rank important question phrases (MMPs) for question answering in Section 3. This framework seamlessly incorporates lexical, syntactic and semantic information, resulting in an MMP prediction F-measure as high as 88.6%. Secondly, we show that features derived from identified MMPs improve significantly a relevance classification model, in Section 4.2. Thirdly, we show that using the improved relevance model into our QA system results in a statistically significant 5 point improvement in F-measure, in Section 5. This finding is further corroborated by the results on the official 2012 BOLT IR (IR, 2012) task where the combined system yielded the best performance in the evaluation.

2 Related Work

Popular information retrieval systems like BM25 (Robertson and Walker, 1994) and language models (Ponte and Croft, 1998) use unsupervised techniques based on corpus statistics for term weighting. Many of these techniques are variants of the one proposed by (Luhn, 1958). Recently, several researchers have studied approaches for term weighting using supervised learning techniques. However, much of this research has focused on information retrieval task rather than on question answering problems of the nature addressed in this paper. (Bendersky and Croft, 2008) restricted themselves to predicting key noun phrases, which is perhaps sufficient for a retrieval task. However, for questions like "Find comments about how American hedge funds legally avoid taxes," the verb "avoid" is perhaps as important as the noun phrase "American hedge funds" and "taxes". Works like that of (Lease et al., 2009) and (Zhao and Callan, 2010) predict importance at the word level. While word level importance is perhaps sufficient for an IR task, predicting the importance of phrases, especially those derived from a parse tree, gives a much richer representation that might also be useful for better question understanding and thus generate more relevant answers. Both (Lease et al., 2009; Zhao and Callan, 2010) propose supervised methods that learn from a large set of queries and relevance judgments on their answers. While this is possible in a TREC Ad-hoc-retrieval-like task, such a large training corpus of question-answer pairs is unavailable for most scenarios. (Monz, 2007) learns term weights for the IR component of a question answering task. His work unlike ours does not aim to find the answers to the questions.

Most QA systems in the literature have dealt with answering factoid questions, where the answer is a noun phrase in response to questions of the form "Who," "Where," "When." Most systems have a question analysis component that represents the question as syntactic relations in a parse or as deep semantic relations in a handcrafted ontology (Hermjakob et al., 2000; Chu-carroll et al., 2003; Moldovan et al., 2003). In addition certain systems (Bunescu and Huang, 2010) aim to find the "focus" of the question, that is, the noun-phrases in the question that would co-refer with answers. Additionally, much past work has focused on finding the lexical answer type (Pinchak, 2006; Li and Roth, 2002). Since these papers considered a small number of answer types, rules over the detected relations and answer types could be applied to find the relevant answer. However, since our system answers non-factoid questions that can have answer of arbitrary types, we want to use as few rules as possible. The MMPs therefore become a critical component of our system, both for question analysis and for relevance detection.

3 Question Data and MMP Model

To train the MMP model, we first create a set of questions and label their MMPs. The labeled data is then used to train a statistical model to predict MMPs for new questions as discussed next.

3.1 Question Corpus

We use a subset of the DARPA BOLT corpus (see Section 5.1) containing forum postings in English. Four annotators use a search tool to explore this document collection. They can perform keyword searches and retrieve forum threads from which they generate questions. The program participants decided a basic set of question types that are out-of-scope of the current research agenda. Accordingly,

annotators cannot generate questions (1) that require reasoning or calculation over the data to compute the answers; (2) that are vague or ambiguous; (3) that can be broken into multiple disjoint questions; (4) that are multiple choice questions; (5) that are factoid questions—the kinds that have already been well studied in TREC (Voorhees, 2004). Any other kind of question is allowed. Two other annotators, who have neither browsed the corpus nor generated the questions, mark selected spans of the questions into one of two categories—*MMP-Must* and *MMP-maybe*. The annotation tool allows arbitrary spans to be highlighted and the annotators are instructed to select spans corresponding to the smallest semantic units. The phrases that are very likely to appear contiguously in a relevant answer are marked as *MMP-Must*. Annotators can mark multiple spans per question, but not overlapping spans. We generated 201 annotated questions using this process.

Figure 1 contains an example, where "American," "hedge fund," and "legally avoid taxes" are required elements to find answers and are thus marked as *MMP-Must*s (signified by enclosing rectangles). We purposely annotate MMPs at the word level and not in the parse tree, because this requires minimal linguistic knowledge. We do, however, employ an automatic procedure to attach MMPs to parse tree nodes when generating MMP training instances.

3.2 MMP Training

Questions annotated in Section 3.1 are first processed by an information extraction (IE) pipeline consisting of syntactic parsing, mention detection and coreference resolution (Florian et al., 2004; Luo et al., 2004; Luo and Zitouni, 2005). After IE, we have access to the syntactic structure represented by a parse tree and semantic information represented by coreferenced mentions (including those of named entities).

To take advantage of the availability of the syntactic and semantic information, we first attach the MMP annotations to parse tree nodes of a question, and, if necessary, we augment the parse tree.

There are several reasons why we want to embed the MMPs into a parse tree. First, many constituents in parse trees correspond to important phrases we want to capture, especially proper names. Second, after an MMP is attached to a tree node, the problem

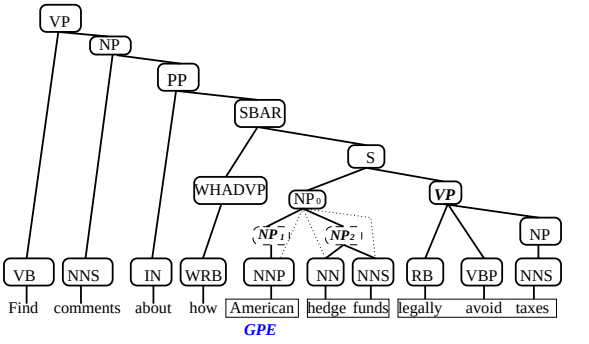

Figure 1: MMPs are aligned with tree nodes: MMPs are shown in rectangular boxes along with their aligned nodes (with slanted labels); augmented parse tree nodes (i.e., NP1, NP2) in dashed nodes. Dotted edges under NP0 are the structure before the tree is augmented.

of predicting MMPs reduces to classifying parse tree nodes, and syntactic information can be naturally built into the MMP classifier. Lastly, and more importantly, associating MMPs with tree nodes opens the door to explore features derived from the syntactic parse tree. For instance, it is easy to read bilexical dependencies from a parse tree (provided that head information is propagated); with MMPs aligned with the parse tree, bilexical dependencies can be ranked by examining whether or not an MMP phrase is a head or a dependent. This way, not only are the dependencies in a question captured, but MMP scores or ranks can be propagated to dependencies as well. We will discuss more how MMP features are computed in Section 4.2.2.

Annotators can mark MMPs that are not perfectly aligned with a tree node. Hence, care has to be taken when generating MMP training instances. As an example, In Figure 1, "American" and "hedge funds" are marked as two separate MMPs, but the Penn-Tree-style parse tree has a flat "NP0" constituent spanning directly on "American hedge fund," illustrated in Figure 1 as dotted edges.

To anchor MMPs in the parse tree, we *augment* it by combining the IE output and the MMP annotation. In the aforementioned example, "American" is a named mention with the entity type GPE (geopolitical entity) and there is no non-terminal node spanning it: so, a new node "NP1" is created; "hedge funds" is marked as an MMP: so, a second node ("NP2") is created to anchor it.

A training instance for building the MMP model is defined as a span along with an MMP label. For instance, "hedge funds" in Figure 1 will generate a positive training instance as ⟨(5,6), +1⟩, where (5,6) is the span of "hedge funds" in the question sentence, and +1 signifies that it is a positive training instance. For the purpose of this paper we use only binary labels, mapping all MMP-Must to $+1$ and MMP-Skip and MMP-Maybe to -1.

Formally, we use the following procedure to generate training instances:

Algorithm 1 Pseudo code to generate MMP training instances.

Input: An input question tree with detected mentions and marked MMPs
Output: A list of MMP training instances
1: Foreach mention m in the question
2: if no node spans m, and m does not cross bracket
3: Find lowest node N dominating m
4: Insert a child node of N that spans exactly m
5: Foreach mention p in marked MMPs
6: Find lowest non-terminal N_p dominating p
7: Generate a positive training example for N_p
8: Mark N_p as visited
9: Recursively generate instances for N_p's children
10: Generate a negative training instance for all unvisited nodes in Step 5-9

Steps 1 to 4 augment the question tree by creating a node for each named mention, provided that no existing node spans exactly the mention and the mention does not cross-bracket tree constituents. Steps 5 to 8 generate positive training instances for marked MMPs; step 9 recursively generates positive training instances [2] for tree nodes dominated by N_p, where N_p is the lowest non-terminal node dominating the marked MMP p.

After MMP training instances are generated we design and compute features for each instance, and use them to train a classifier.

3.3 MMP Features and Classifier

We compute four types of features that will be used in a statistical classifier. These features are designed to characterize a phrase from the lexical, syntactic, semantic and corpus-level aspect. The weights associated with these features are automatically learned from training data.

We will use "(NP1 American)" in Figure 1 as the running example below.

Lexical Features: Lexical features are motivated by the observation that spellings in English sometimes offer important cues about word significance. For example, an all-capitalized word often signifies an acronym; an all-digit word in a question is likely a year, etc. We compute the following lexical features for a candidate MMP:

CaseFeatures: is the first word of an MMP upper-case? Is it all capital letters? Does it contain numeric letters? For "(NP American)" in Figure 1, the upper-case feature fires.

CommonQWord: Does the MMP contain question words, including "What," "When," "Who," etc.

Syntactic Features: The second group of features are computed from syntactic parse trees after annotated MMPs are aligned with question parse-trees as described previously.

PhraseLabel: this feature returns the phrasal label of the MMP. For "(NP American)" in Figure 1, the feature value is "NP." This captures that an NP is more likely an MMP than, say, an ADVP.

NPUnique: this Boolean feature fires if a phrase is the only NP in a question, indicating that this constituent probably should be matched. For "(NP American)," the feature value would be false.

PosOfPTN: these features characterize the position of the parse tree node to which an MMP is anchored. They compute: (1) the position of the left-most word of the node; (2) whether the left-most word is the beginning of the question; (3) the depth of the anchoring node, defined as the length of the path to the root node. For "(NP American)" in Figure 1, the features state that it is the 5th word in the sentence; it is not the first word of the sentence; and the depth of the node is 6 (where root has depth 0).

PhrLenToQLenRatio: This feature computes the number of words in an MMP, and its relative ratio to the sentence length. This feature controls the length of MMPs at decoding time, since most of MMPs are short.

Semantic Features (NETypes): The third group of features are computed from named entities and aim to capture semantic information. The feature tests if

[2] One exception to this step is that if a node spans a single stop word, then a negative training instance is generated.

a phrase is or contains a named entity, and, if this is the case, the value is the entity type. For "(NP American)" in Figure 1, the feature value would be "GPE."

Corpus-based Features (AvgCorpusIDF): This group of features computes the average of the IDFs of the words in this phrase. From the corpus IDF, we also compute the ratio between the number of stop words and the total number of words in the MMP, and use it as another feature.

3.4 MMP Classification Results

We now show that we can reliably predict MMPs of questions. We split our set of 201 annotated questions into a training set consisting of 174 questions and a test set with the remaining 27 questions. We use the procedure and features described in Section 3 to train a logistic regression binary classifier using WEKA. Then, the trained MMP classifier is applied to the test set question trees. Since the class bias is quite skewed (only 16% of the phrases are marked as MMP-Must) we also use re-sampling at training time to balance the prior probability of the two classes. At testing time, a parser and a mention detection algorithm (Florian et al., 2004; Luo et al., 2004; Luo and Zitouni, 2005) are run on each question. The detected mentions are then used to augment the question parse trees. The MMP classifier achieves an 88.6% F-measure (cf. Table 1, with 91.6% precision). This is a respectable number, considering the limited amount of training data. We experimented with decision trees and bagging as well but found logistic regression to work the best.

Feature	P	R	F1
AvgCorpusIDF	0.849	0.634	0.725
+NPUnique	0.868	0.634	0.732
+NETypes	0.867	0.662	0.750
+PhraseLabel	0.890	0.705	0.783
+CaseFeatures	0.829	0.820	0.824
+PosOfPTN	0.911	0.852	0.880
+PhrLenToQLenRatio	0.915	0.855	0.883
+commonQWord	0.916	0.858	0.886

Table 1: The performances of the MMP classifier while incrementally adding features.

The examples in Table 2 illustrate the top three MMPs produced by the model on two questions. These results are encouraging: in the first example the word AIDS is clearly the most "important" word, but IDF alone is not adequate to place it in the top since AIDS is also a common verb (words are lower-cased before IDF look-up). Similarly, in the third example, the phrase "the causes" has a much higher MMP score than the phrase "the concerns" (MMP score of 0.109), even though the words "concerns" has a slightly higher IDF, 2.80, than the word "causes"(2.68). However, in this question, understanding that the word "causes" is critical to the meaning of the question is critical and is captured by the MMP model.

We analyzed feature importance for MMP classification by incrementally adding each feature group to the model. The result is tabulated in Table 1. Not surprisingly, syntactical (i.e., "NPUnique," "PhraseLabel" and "PosOfPTN") and semantic features (i.e., "NETypes") are complementary to the corpus-based statistics features (i.e., average IDF). Lexical features also improve recall: the addition of "CaseFeatures" boosts the F-measure by 4 points. At first sight, it is surprising that the feature group "PosOfPTN," which characterize the position of a candidate MMP relative to the sentence and relative to the parse tree, has such a large impact—it improves the F-measure by 5.6 points. However, a cursory browsing of the training questions reveals that most MMPs are short and concentrate towards the end of the sentence. So this feature group helps by directing the model to predict MMPs at the end of the sentence and to prefer short phrases versus long ones.

4 Relevance Model with MMPs

We now validate our second hypothesis that MMPs are effective for open domain question answering. We demonstrate this through the improvement in performance on relevance prediction. More specifically, given a natural language question, the task is one of finding relevant sentences in posts on online forums. The relevance prediction component is critical for question answering as has been seen in TREC(Ittycheriah and Roukos, 2001) and more recently in the Jeopardy challenge(Gondek et al., 2012). The improved relevance model further improves our question answering system as seen in Section 5.

Question	Top 3 MMPs	MMP-score	Top words by IDF
List statistics about changes in the demographics of AIDS.	1: AIDS 2: changes 3: the demographics	0.955 0.525 0.349	demographics AIDS statistics
What are the concerns about the causes of autism?	1: autism 2: the causes 3: the causes of autism	0.989 0.422 0.362	autism concerns causes

Table 2: Example questions and the top-3 phrases ranked by the MMP model.

4.1 Data for Relevance Model

The data to train and test the relevance model is obtained as follows. First, a rudimentary version (i.e., key word search) of a QA system using Lucene is built. The Lucene index comprised of a large number of threads in online forums released to the participants of the BOLT-IR task(IR, 2012) for development of our systems. The corpus is described in more detail in Sec. 5. Top snippets returned by the search engine are judged for relevancy by our annotators. The initial (small) batch of data is used to train a relevance model which is deployed in the system. The new model is in turn used to create more answers for new questions. When more data is collected, the relevance model is retrained and redeployed to collect more data. The process is iterated for several months, and at the end of this process, a total of 390 training questions are created and about 28,915 snippets are judged by human annotators, out of which about 6,528 are relevant answers. These question-answers pairs are used to train the final relevance model used in our question-answering system. A separate held-out test set of 59 questions is created and its system output is also judged by humans. This data set is our test set.

4.2 Relevance Prediction

A key component in our question-answering system is the snippet relevance model, which is used to compute the probability that a snippet is relevant to a question. The relevance model is a conditional distribution $P(r|q, s; D)$, where r is a binary random variable indicating if the candidate snippet s is relevant to the question q. D is the document where the snippet s is found.

In our question answering system, MMPs extracted from questions are used to compute the features for the relevance model. To test their effectiveness, we conduct a controlled experiment by comparing the system with MMP features with 2 baselines: (1) a system without MMP features; (2) a baseline with each word as an MMP and the word's IDF as the MMP score.

4.2.1 Baseline Features

We list the features used in our baseline system, where no MMP feature is used. The features can be categorized into the following types. **(1) Text Match Features**: One set of features are the cosine scores between different representations of the query and the snippet. In one version the query and snippet words are used as is; in another version the query and snippet are stemmed using porter stemmer; in yet another the words are morphed to their roots by a table extracted from WordNet. We also compute the inclusion scores (the proportion of query words found in the snippet) and other word overlap features. **(2) Answer Type Features:** The top 3 predictions of a statistical classifier trained to predict answer categories were used as features. **(3) Mention Match Features** compute whether a named entity in the query occurs in the snippet. The matching takes into consideration the results from within and cross document coreference resolution components for nominal and pronominal mentions. **(4) Event match features** use several hand-crafted dictionaries containing terms exclusive to various types of events like "violence", "legal", "election". Accordingly a set of features that take a value of "1" if both the query and snippet contain the same event type were designed. **(5) Snippet Statistics:** Several features based on snippet length, the position of the snippet in the post etc were created.

4.2.2 Features Derived from MMP

The MMPs extracted from questions are used to compute features in the following ways.

As MMPs are aligned with a question's syntactic tree, they can be used to find answers by matching a question constituent with that of a candidate snippet. The MMP model also returns a score for each phrase, which can be used to compute the degree to which a question matches a candidate snippet.

In this section, we use $s = w_1^n$ to denote a snippet with words $w_1, w_2, \cdots, w_n$, and m to denote a phrase from the MMP model along with a score $M(m)$. The features are listed below:

HardMatch: Let $I(m \in s)$ be a 1 or 0 function indicating if a snippet contains the MMP m, then the hard match score is computed as:

$$HM(q,s) = \frac{\sum_{m \in q} M(m) I(m \in s)}{\sum_{m \in q} M(m)}.$$

SoftLMMatch: The SoftLMMatch score is a language-model (LM) based score, similar to that used in (Bendersky and Croft, 2008), except that MMPs play the role of concepts. The snippet-side language model score $LM(v|s)$ is computed as:

$$LM(v|s) = \frac{\sum_{i=1}^{n} I(w_i = v) + 0.05}{n + 0.05|V|},$$

where w_i is the i^{th} in snippet s; $I(w_i = v)$ is an indicator function, taking value 1 if w_i is v and 0 otherwise; $|V|$ is the vocabulary size.

The soft match score between a question q and a snippet s is then:

$$SM(q,s) = \frac{\sum_{m \in q} \left(M(m) \prod_{w \in m} LM(w|s) \right)}{\sum_{m \in q} M(m)},$$

where $m \in q$ denotes all MMPs in question q, and similarly, $w \in m$ signifying words in m.

MMPInclScore: An MMP m's inclusion score is:

$$IS(m,s) = \frac{\sum_{w \in m} I(l(w,s) > \delta) IDF(w)}{\sum_{w \in m} IDF(w)},$$

where $w \in m$ are the words in m; $I(\cdot)$ is the indicator function taking value 1 when the argument is true and 0 otherwise; δ is a constant threshold; $IDF(w)$ is the IDF of word w. $l(w,s)$ is the similarity of word w to the snippet s as: $l(w,s) = max_{v \in s} JW(w,v)$, where $JW(w,v)$ is the Jaro Winkler similarity score between words w and v.

The MMP weighted inclusion score between the question q and snippet s is computed as:

$$IS(q,s) = \frac{\sum_{m \in q} M(m) IS(m,s)}{\sum_{m \in q} M(m)}$$

MMPRankDep: This feature, $RD(q,s)$ first tests if there exists a matched bilexcial dependency between q and s; if yes, it further tests if the head or dependent in the matched dependency is the head of any MMP.

Let $m_{(i)}$ be the i^{th} ranked MMP; let $\langle w_h, w_d | q \rangle$ and $\langle u_h, u_d | s \rangle$ be bilexical dependencies from q and s, respectively, where w_h and u_h are the heads and w_d and u_d are the dependents; let $EQ(w,u)$ be a function testing if the question word w and snippet word u are a match. In our implementation, $EQ(w,u)$ is true if either w and u are exactly the same, or their morphs are the same, or they head the same entity, or their synset in WordNet overlap. With these notations, $RD(q,s)$ is true if and only if

$$EQ(w_h, u_h) \wedge EQ(w_d, u_d) \wedge w_h \in m_{(i)} \wedge w_d \in m_{(j)}$$

is true for some $\langle w_h, w_d | q \rangle$, for some $\langle u_h, u_d | s \rangle$ and for some i and j.

$EQ(w_h, u_h) \wedge EQ(w_d, u_d)$ requires that the question dependency $\langle w_h, w_d | q \rangle$ and the snippet dependency $\langle u_h, u_d | s \rangle$ match; $w_h \in m_{(i)} \wedge w_d \in m_{(j)}$ requires that the head word and dependent word are in the i^{th}-rank and j^{th} rank MMP, respectively. Therefore, $RD(q,s)$ is a dependency feature enhanced with MMPs.

To test the effectiveness of the MMP features, we trained 3 snippet classifiers on the data described in Section 4.1: one baseline system without MMP features (henceforth "no-MMP"); a second baseline with words as MMPs and their IDFs as the scores in the MMP model(henceforth "IDF-as-MMP"); the third system uses the MMPs generated by the model from Section 3 and all MMP features described in this section. We used two types of classifiers: decision tree (DTree) and logistic regression (Logit).

The classification results on a set of 59 questions disjoint from the training set are shown in Table 3. The numbers in the table are F-measure on answer snippets (or positive snippets). Within a machine

Model	Learner	
	DTree	Logit
noMMP	0.426	0.458
IDF-as-MMP	0.413	0.455
MMP	0.451	**0.470**

Table 3: F-measure for Relevance Prediction.

learning method, the model with MMP features is always the best. Between the two classifiers, the logistic regression models are consistently better than the decision tree ones. The results show that MMP features are very helpful to the relevance model.

5 End-to-End System Results

The question-answering system is used in the 2012 BOLT IR evaluation (IR, 2012). The task is to answer questions against a corpus of posts collected from Internet discussion forums in 3 languages: Arabic, Chinese and English. There are 499K, 449K and 262K threads in each of these languages. The Arabic and Chinese posts were first translated into English before being processed. We now describe our experiments on the set of 59 questions developed internally and demonstrate the effectiveness of an MMP based relevance model in the end-to-end system. In the next subsection we discuss our performance in the BOLT-IR evaluation done by NIST for DARPA.

We now briefly describe the question-answering system we developed for the DARPA BOLT IR task, where we applied the MMP classifier and its features. Users submit questions to the system in natural language; the BOLT program mandates that these questions comply with the restrictions described in Section 3.1. Questions are analyzed by a query preprocessing stage that includes our MMP extraction classifier. The preprocessed queries are converted to search queries. These are sent to an Indri-based search engine (Strohman et al., 2005), which returns candidate passages, typically spanning numerous sentences. Each sentence of the retrieved passages is analyzed by a relevance detection module, consisting of a statistical classifier that uses, among others, features computed from the MMPs extracted from the questions. Sentences or spans that are deemed relevant to the question by the relevance detection module are further grouped into equivalence classes that provide different information about the answers. The system generates a single answer for each equivalence class, since elements of the same class are redundant with respect to each other. The elements of each equivalence class are converted into citations that support the corresponding answer.

The ultimate goal of the MMP model is to improve the performance of our question-answering system. To test the effectiveness of the MMP model, we contrast the model trained in Section3 with an IDF baseline, where each non-stop word in a question is an MMP and its score is the corpus IDF. The IDF baseline is what a typical question answering system would do in absence of deep question analysis. To have a fair comparison, the two systems are tested on the same set of 59 questions as the relevance model.

The results of the IDF baseline and MMP system are tabulated in Table 4. Note that the recalls are less than 1.0 because (1) annotated snippets come from both systems; (2) the annotation is done for all snippets in a window surrounding system snippets.

As can be seen from Table 4, the MMP system is about 5 points better than the baseline system. The precision is notably better by 2 points, and the recall is far better (by 7.7%) than that of the baseline. We also compute the question-level F-measures and conduct a Wilcoxon signed-rank test for paired samples. The test indicates that the MMP system is better than the baseline system at $p < 0.00066$. Therefore, the MMP system has a clear advantage over the baseline system.

System	Prec	Recall	F1
baseline	.4228	.3679	.3935
MMP	.4425	.4452	.4438

Table 4: End-to-End system result on 59 questions.

5.1 BOLT Evaluation Results

The BOLT evaluation consists of 146 questions, mostly event- or topic- related, e.g., “What are people saying about the ending of NASA’s space shuttle program?”. A system answer, if correct, is mapped manually to a facet, which is one semantic unit that answers the question. For each question, facets are collected across all participants’ submission. A

facet-based F-measure is computed for each participating site. The recall from which the official F-measure is computed is weighted by snippet citations (a citation is a reference to the original document that supports the correct facet). In other words, a snippet with more citations leads to a higher recall than one with less citations. The performances of 4 participating sites are listed in Table 5. Note that the F-measure is weighted and is not necessarily a number between the precision and the recall.

	Facet Metric		
Site	Precision	Recall	(Weighted) F
SITE 1	0.2713	0.1595	0.1713
SITE 2	0.1500	0.1316	0.1109
SITE 3	0.1935	0.2481	0.1734
Ours	0.2729	0.2195	0.2046

Table 5: Official BOLT 2012 IR evaluation results.

.

Among 4 participating sites, our system has the highest performance. SITE 1 has about the same level of precision, with lower recall, while SITE 3 has the best recall, but lower precision. The results validate that the MMP question analysis technique presented in this paper is quite effective.

6 Conclusions

We propose a framework to select and rank mandatory matching phrases (MMP) for question answering. The framework makes full use of the lexical, syntactic and semantic information in a question and does not require answer data.

The proposed MMP framework is tested at 3 levels in a full QA system and is shown to be very effective to improve its performance: first, we show that it is possible to reliably predict MMPs from questions alone: the MMP classifier can achieve an F-measure as high as 88.6%; second, phrases proposed by the MMP model are incorporated into a snippet relevance model and we show that it improves its performance; third, the MMP framework is used in an question answering system which achieved the best performance in the official 2012 BOLT IR (IR, 2012) evaluation.

Acknowledgments

This work was partially supported by the Defense Advanced Research Projects Agency under contract No. HR0011-12-C-0015. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the U.S. government and no official endorsement should be inferred.

References

Michael Bendersky and W. Bruce Croft. 2008. Discovering key concepts in verbose queries. *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval - SIGIR '08*, page 491.

Michael Bendersky, Donald Metzler, and W. Bruce Croft. 2010. Learning concept importance using a weighted dependence model. *Proceedings of the third ACM international conference on Web search and data mining - WSDM '10*, page 31.

Michael Bendersky. 2011. Parameterized concept weighting in verbose queries. *Proceedings of the 34th annual international ACM SIGIR conference on research and development in information retrieval*.

Razvan Bunescu and Yunfeng Huang. 2010. Towards a general model of answer typing: Question focus identification. In *Proceedings of the 11th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*.

Jennifer Chu-carroll, John Prager, Christopher Welty, Krzysztof Czuba, and David Ferrucci. 2003. A multi-strategy and multi-source approach to question answering. In *In Proceedings of Text REtrieval Conference*.

R Florian, H Hassan, A Ittycheriah, H Jing, N Kambhatla, X Luo, N Nicolov, and S Roukos. 2004. A statistical model for multilingual entity detection and tracking. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 1–8, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

D. C. Gondek, A. Lally, A. Kalyanpur, J. W. Murdock, P. A. Duboue, L. Zhang, Y. Pan, Z. M. Qiu, and C. Welty. 2012. A framework for merging and ranking of answers in DeepQA. *IBM Journal of Research and Development*, 56(3.4):14:1 –14:12, may-june.

Ulf Hermjakob, Eduard H. Hovy, and Chin yew Lin. 2000. Knowledge-based question answering. In *In Proceedings of the 6th World Multiconference on Systems, Cybernetics and Informatics (SCI-2002*, pages 772–781.

BOLT IR. 2012. Broad operational language translation (BOLT). www.darpa.mil/Our_Work/I2O/Programs/Broad_Operational_Language_Translat%ion_(BOLT).aspx. [Online; accessed 10-Dec-2012].

Abraham Ittycheriah and Salim Roukos. 2001. IBM's statistical question answering system - TREC-11. In *Proceedings of the Text REtrieval Conference*.

Matthew Lease, James Allan, and W. Bruce Croft. 2009. *Advances in Information Retrieval*, volume 5478 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, April.

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.

H. P. Luhn. 1958. A business intelligence system. *IBM J. Res. Dev.*, 2(4):314–319, October.

Xiaoqiang Luo and Imed Zitouni. 2005. Multilingual coreference resolution with syntactic features. In *Proc. of Human Language Technology (HLT)/Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. of ACL*.

Dan Moldovan, Christine Clark, Sanda Harabagiu, and Steve Maiorano. 2003. Cogex: a logic prover for question answering. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 87–93.

Christof Monz. 2007. Model tree learning for query term weighting in question answering. In *Proceedings of the 29th European conference on IR research*, ECIR'07, pages 589–596, Berlin, Heidelberg. Springer-Verlag.

Christopher Pinchak. 2006. A probabilistic answer type model. In *In EACL*, pages 393–400.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA. ACM.

S. E. Robertson and S. Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval*, SIGIR '94, pages 232–241, New York, NY, USA. Springer-Verlag New York, Inc.

Trevor Strohman, Donald Metzler, Howard Turtle, and W. Bruce Croft. 2005. Indri: a language-model based search engine for complex queries. Technical report, in Proceedings of the International Conference on Intelligent Analysis.

Ellen M. Voorhees. 2004. Overview of the TREC 2004 question answering track. In *TREC*.

Le Zhao and Jamie Callan. 2010. Term necessity prediction. *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*, page 259.