Joint Question Clustering and Relevance Prediction for Open Domain Non-Factoid Question Answering

Snigdha Chaturvedi
University of Maryland,
College Park,
Maryland, USA
snigdhac@cs.umd.edu

Vittorio Castelli
IBM T. J. Watson Research Center,
Yorktown Heights,
New York, USA
vittorio@us.ibm.com

Radu Florian
IBM T. J. Watson Research Center,
Yorktown Heights,
New York, USA
raduf@us.ibm.com

Ramesh M. Nallapati
IBM T. J. Watson Research Center,
Yorktown Heights,
New York, USA
nallapati@us.ibm.com

Hema Raghavan
IBM T. J. Watson Research Center,
Yorktown Heights,
New York, USA
hraghav@us.ibm.com

ABSTRACT

Web searches are increasingly formulated as natural language questions, rather than keyword queries. Retrieving answers to such questions requires a degree of understanding of user expectations. An important step in this direction is to automatically infer the type of answer implied by the question, e.g., factoids, statements on a topic, instructions, reviews, etc.

Answer Type taxonomies currently exist for factoid-style questions, but not for open-domain questions. Building taxonomies for non-factoid questions is a harder problem since these questions can come from a very broad semantic space. A few attempts have been made to develop taxonomies for non-factoid questions, but these tend to be too narrow or domain specific. In this paper, we address this problem by modeling the Answer Type as a latent variable that is learned in a data-driven fashion, allowing the model to be more adaptive to new domains and data sets. We propose approaches that detect the relevance of candidate answers to a user question by jointly 'clustering' questions according to the hidden variable, and modeling relevance conditioned on this hidden variable.

In this paper we propose 3 new models: (a) Logistic Regression Mixture (LRM), (b) Glocal Logistic Regression Mixture (G-LRM) and (c) Mixture Glocal Logistic Regression Mixture (MG-LRM) that automatically learn question-clusters and cluster-specific relevance models. All three models perform better than a baseline relevance model that uses explicit Answer Type categories predicted by a supervised Answer-Type classifier, on a newsgroups dataset. Our models also perform better than a baseline relevance model that does not use any answer-type information on a blogs dataset.

WWW'14, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2744-2/14/04.
http://dx.doi.org/10.1145/2566486.2567999 .

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Metrics—*Retrieval models; Clustering*

Keywords

Question Answering, Latent Variable Models, Relevance Prediction, Question Clustering

1. INTRODUCTION

With the emergence of mobile devices and digital personal assistants such as Siri, users have come to expect search engines to answer well formed natural language questions. Recent improvements in many search engines indicate an attempt to address such complex information needs by better understanding users' questions [1, 2, 6] and in turn providing better summaries of retrieved passages and sometimes by directly extracting answers. For example, both Google$^{\mathrm{TM}}$and Bing$^{\mathrm{TM}}$return a factoid-style answer for the question "*Who is the president of the United States*". While these approaches have been successful for a small class of factoid queries or questions (typically on the topics of "people", "weather", "definitions", "movie showtimes", etc.), web-search engines still fail to understand the vast majority of natural language questions. Specifically, search engines still return the standard 10 blue links for *subjective questions* that express complex information needs. Examples of such questions are: "*Is abortion ethical?*" and "*What was the impact on NYC of the stop-and-frisk practice by the NYPD?*" Users resort to social media, newsgroups and community QA systems for these kinds of questions [10]. In this paper we aim to answer subjective questions of this nature by rooting our methods in both Document Retrieval and Question Answering.

Usually Question Answering makes a broad distinction between factoid and non-factoid questions [36]. Factoid questions have fixed, fact-based answers, which can typically be easily categorized into a small, well defined taxonomy. For example, the question "*Who was the first woman killed in the Vietnam War?*" expects a person name as an answer, while "*Which country has the longest life expectancy?*" expects for a country name. Conversely, non-factoid questions such as the one on the 'stop-and-frisk policy' mentioned above are more subjective in nature, and tend to have more than a single correct answer.

The task of finding an answer for factoid questions has received a lot of attention in the past [36]. A popular approach in this domain is to classify the question into an Answer Type class [19], which gives clues about the nature of the answer sought by the question. Answer Type determination is useful because it reduces the search space of answers (one would not consider person names as candidate answers to the question "*Which country has the longest life expectancy?*"), helps in building specific features and enables specialized answer-type-specific relevance prediction models.

For factoid question answering in a limited domain, it is likely that a large set of the questions will fall into a limited set of Answer Type classes. However, there is evidence that the same does not hold for non-factoid questions. Although authors have shown the importance of determining the Answer Types for non-factoid questions (e.g., see Razmara and Kosseim [33] for questions requiring *Lists* as answers), it appears that determining a broad, reusable taxonomy is a difficult task, even in limited domains and with restrictions on how questions are formulated. For example, Verberne *et al.* [35] experiment with two different datasets of *Why* questions and use different taxonomies for each. Designing a new taxonomy for every dataset and then labeling training questions accordingly requires considerable manual efforts and is costly.

We circumvent the difficulty of establishing an Answer Type taxonomy by using a latent variable as proxy for Answer Type or for other categorization criteria that might be used to group questions. We introduced supervised probabilistic framework that groups questions into latent clusters and models the relevance of a candidate answer to the question jointly, using training data annotated only for relevance.

Since our proposed techniques make no assumptions about question categories, they can easily be adapted to new domains. We can interpret our approach as one of joint question-clustering and relevance-prediction, and show that our models result in better predictions via empirical evaluations on two different datasets.

Our contributions can be summarized as follows:

- We introduce a framework that assumes that non-factoid questions, like factoid questions, can belong to numerous categories or clusters which can be useful for relevance prediction. Our assumption is validated in our experiments when our models outperform a cluster agnostic baseline.

- We present three different probabilistic models that identify these latent question clusters in a data driven manner and build cluster specific relevance prediction models.

- We show that our models outperform competitive baselines [28] which use manually labeled data to construct an Answer-Type classifier by at least 5.5%.

- We compare our models to a method that uses an Answer-Type Oracle and show that they perform competitively.

In the next section we describe related work. Section 3 describes our predictive models and their training procedure in details, and is followed by an empirical evaluation in Section 4. Section 5 concludes the paper with a brief summary of the findings and a discussion of future work.

2. RELATED WORK

Factoid and non-factoid questions have mostly been treated differently, and there has been some previous work in identifying whether a question requires a factoid answer, primarily for community question answering. Li *et al.* propose a supervised [26] and a co-training based approach [25] to address this problem. Aikawa *et al.* [3] present a supervised approach using lexical features for the same problem. While these approaches require hand-tagged data for training, Zhou *et al.* [39] utilize social signals in Community Question Answering (CQA) forums to collect training data.

The domain of factoid question answering has attracted a lot of attention in the past. Srihari and Li [34] and Moldovan *et al.* [29] had shown the usefulness of categorizing a question using the question word alone or a combination of question word and answer's semantic class. Hovy *et al.* [19, 23] presented a system that classified a question according to the desired Answer Type followed by a matching of questions and candidate answers using word level and parse tree level information. Following this, numerous efforts focused on two areas: question classification and improving answer pinpointing. Hermjakob [17] enriched question parsing using the Penn treebank corpus with additional question treebank while Li and Roth [27] hierarchically classified questions to a coarse and fine classes using words, POS tags, chunks, named entities, head chunk and semantically related words as features. There have been several other attempts at identifying good sets of features for question classification in factoid question answering such as those using subtree features [24] and head-words and their hypernyms [20]. Moschitti *et al.* [30] exploited Predicate-Argument Structure (PAS) for Question Classification, relevance detection and answer re-ranking. All of these works focused on factoid questions of the kind studied in the TREC QA task [36]. The Watson computer system that competed (and won!) in the game-show Jeopardy also made heavy use of components that classified questions and determined the lexical answer type [22]. However, the format of Jeopardy also largely deals with factoid questions. As Watson is being adapted to other domains like medical [1] these underlying components have to be adapted and re-trained.

The domain of non-factoid question answering is more challenging. Previous research has tried to exploit the idea of question classification for answer determination. However, this has been limited to retrieving answers to specific question types (like "Why" or "How" questions) and/or often designing a new domain specific taxonomy. For example, Oh *et al.* [31] suggest using sentiment features and word class features to retrieve answers to *Why* questions, which are a small subset of general non-factoid questions. Higashinaka and Isozaki [18] build a Japanese Why-Question Answering system using causal relationships extracted in a data-driven manner. On the other hand, Bu *et al.* [8] addressed the general Chinese non-factoid question-answering task by designing a taxonomy of six Answer Type categories. Their system is based on Markov Logic Networks trained on lexical features. Qu *et al.* [32] present a supervised method to automatically classifying questions from the popular CQA website, Yahoo! Answers into manually designed topics arranged in a hierarchical scheme.

The idea of classifying a question into categories before searching for an answer has also been extended to the domain of web query processing. Broder [7] provided a taxonomy for web query classification. Cao *et al.* [9] improve query classification using user's search history. This idea was further extended by Gohar-ian and Mengle [16] using a dynamic window to look for previous queries and using Relationship Net to mine relationships between categories. Beitzel *et al.* [5] use semi-supervised learning to classify queries according to a pre-determined typology. Chen *et al.* [11] and Beitzel *et al.* [4] also propose methods to incorporate unlabeled data to improve query classification.

[1] http://www-01.ibm.com/software/ebusiness/jstart/downloads/MRTAWatsonHIMSS.pdf

Another direction in the domain of web-search has focused on query-rewriting and query clustering in search logs using session information or the query-document click graphs (eg., [21, 38]). These works aim at clustering queries with the same information need. In the current work we are attempting to capture a much broader categorization of the questions. We also don't assume access to click-through information or session information, though those would provide for interesting extensions to our work.

Sometimes a particular question category is further divided into sub categories to build a successful system. Razmara and Kosseim [33] retrieve answers to *List* questions by further dividing this category into nine sub-categories and use word co-occurrences to mine the answer. Similarly, Verberne *et al.* [35] use Rhetorical Structure Theory to mine answers for '*Why*' questions. They experiment with two '*Why*' questions datasets and divide the '*Why*' questions into two and five sub-classes respectively. There have also been several models proposed that implicitly cluster documents (e.g., [37]) for ad-hoc retrieval. Those works usually assume that relevant documents are typically "near" each other in some Euclidean space. Our work is orthogonal to these and looks at clustering questions.

Our work is significantly different from all of the above approaches. Unlike previous approaches, our models do not rely on the existence of a taxonomy. They assume that questions belong to different categories and each category could be modeled more effectively by a different model. These categories are discovered automatedly in a data-driven manner eliminating the need for human-guided taxonomy design or supervised Answer Type classifiers for questions.

3. JOINT QUESTION CLUSTERING AND RELEVANCE PREDICTION MODELS

This section contains a detailed description of our models and their training algorithms.

3.1 Problem Setting

We address the problem of predicting if a given *snippet a* (a short excerpt of a document, e.g., a sentence) provides an answer to a given non-factoid question q. If a answers q, we say that it is a *relevant* snippet, otherwise we say it is *irrelevant*. Our training dataset consists of question and snippet pairs labeled with the corresponding relevance judgments. A relevance judgment r equals to 1 or 0, denoting that the snippet was judged by human annotators to be relevant or irrelevant to the question, respectively. The questions are not assumed to be labeled with any Answer Type categorization and the only supervision given to the model during training is in form of relevance judgment.

3.2 Model Description

We propose three different joint question-clustering and relevance-prediction models that assume that each question belongs to a latent category c (the terms category and cluster are used interchangeably in the rest of the discussion), and that relevance prediction should partly depend on this cluster assignment. Also, during the generative process, each of the three models assumes that the cluster assignment is conditionally independent of the answer snippet given the question. To model the influence of the cluster-assignment on the answer, we introduce a new binary relevance variable r that represents the relevance of the answer to the question and the cluster assignment. As shown in the graphical models in Figure 1, when r is observed, it activates the v-structure between c and a, making them conditionally dependent on each other. For this purpose, for each training instance we design two distinct feature vectors:

For every data instance:

1. Generate the cluster assignment, $c \in \{1...K\}$, given the question, q, using $P(c|q)$ (using weights λ)
2. Generate the relevance prediction, r, given the cluster assignment, c, question, q and answer snippet, a, using $P(r|q,a,c)$ (using cluster specific weights, $\vec{w}_c$, and global weights, $\vec{\omega}$, if applicable)

(a) Logistic Regression Mixture (LRM) and Glocal Logistic Regression Mixture (G-LRM)

For every data instance:

1. Generate the cluster assignment, $c \in \{1 \dots K\}$, given the question, q, using $P(c|q)$ (using weights λ)
2. Generate the global vs local choice, $h \in \{0, 1\}$ with bias π
3. (a) If $h = 0$, generate the relevance prediction, r, given the question, q and answer snippet, a, using $P(r|q,a)$ (using global weights, $\vec{\omega}$)
   (b) If $h = 1$, generate relevance prediction, r, given the cluster assignment, c, question, q and answer snippet, a, using $P(r|q,a,c)$ (using cluster weights, $\vec{w}_c$)

(b) Mixture Glocal Logistic Regression Mixture (MG-LRM)

Figure 2: Generative stories for generating relevance prediction, r, for the three proposed models.

- the clustering feature vector, f_q, used by the clustering components of our models, uses features extracted from only the question
- the relevance feature vector, f_{qa}, used by the relevance prediction components of our models, consists of features extracted from questions and potential answers and both.

All the three proposed models, as well as the baselines described later, use the same set of features described above. Using these definitions and assumptions, the probability of relevance given the question and the answer can be modeled by the following equation:

$$\begin{aligned} P(r|q,a) &= \sum_{c}^{K} P(c|q,a)P(r|q,a,c) \\ &\approx \sum_{c}^{K} P(c|q)P(r|q,a,c) \qquad (1) \end{aligned}$$

All the proposed models parametrize $P(c|q)$ as a log-linear model:

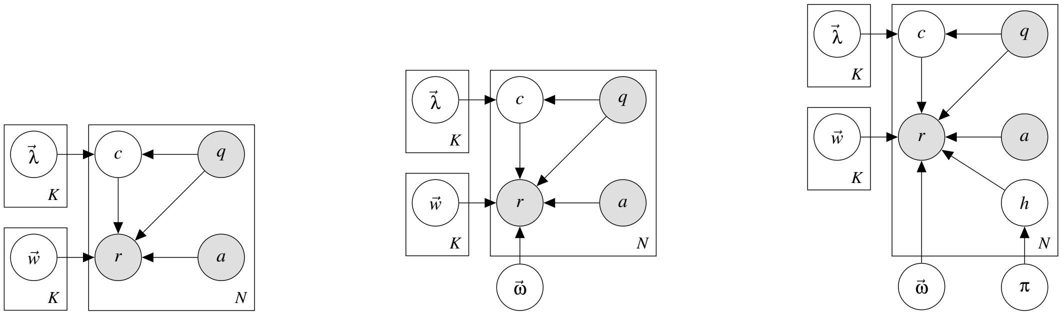
$$P(c|q) = \frac{e^{\vec{\lambda}_c \vec{f}_q}}{\sum_k e^{\vec{\lambda}_k \vec{f}_q}} \qquad (2)$$

where, f_q is the feature vector describing the question q, and $\vec{\lambda}_c$ is the weight vector of the log-linear model for the cluster c. There are K weight vectors, one corresponding to each of the K clusters.

Differences in the three models lie in the ways they use logistic regression to formulate $P(r|q,a,c)$. We now describe the three models in detail.

Logistic Regression Mixture (LRM).

The first model (shown in Figure 1a) assumes that each question cluster has a separate relevance prediction model. Hence, each



(a) Logistic Regression Mixture (LRM)

(b) Glocal Logistic Regression Mixture (G-LRM)

(c) Mixture Glocal Logistic Regression Mixture (MG-LRM)

Figure 1: Plate diagram of the three proposed models. Question, q, potential answer, a and relevance, r are the observed variables while cluster assignment, c and global vs cluster-specific model choice, h, are the latent variables. All other nodes are the model parameters to be learned during training. $\vec{\lambda}$ are the clustering weights used to generate cluster assignment for the question, $\vec{w}$ are the cluster specific relevance prediction weights and $\vec{\omega}$ is the global cluster agnostic relevance prediction weight. π is the bias of the global vs cluster specific model chooser, h.

Model Name	**Objective Function**
LRM	$P(r\|q,a,\mathbf{w},\boldsymbol{\lambda}) = \sum_c^K \frac{e^{\vec{\lambda}_c \vec{f}_q}}{\sum_k e^{\vec{\lambda}_k \vec{f}_q}} \frac{(e^{\vec{w}_c \vec{f}_{qa}})^r}{1+e^{\vec{w}_c \vec{f}_{qa}}} - \alpha \sum_k^K (\|\|\vec{w}_k\|\|^2 + \|\|\vec{\lambda}_k\|\|^2)$
G-LRM	$P(r\|q,a,\vec{\omega},\mathbf{w},\boldsymbol{\lambda}) = \sum_c^K \frac{e^{\vec{\lambda}_c \vec{f}_q}}{\sum_k e^{\vec{\lambda}_k \vec{f}_q}} \frac{(e^{\vec{\omega} \vec{f}_{qa} + \vec{w}_c \vec{f}_{qa}})^r}{1+e^{\vec{\omega} \vec{f}_{qa} + \vec{w}_c \vec{f}_{qa}}} - \alpha \left(\|\|\vec{\omega}\|\|^2 + \sum_k^K (\|\|\vec{w}_k\|\|^2 + \|\|\vec{\lambda}_k\|\|^2) \right)$
MG-LRM	$P(r\|q,a,\vec{\omega},\mathbf{w},\boldsymbol{\lambda}) = \sum_c^K \frac{e^{\vec{\lambda}_c \vec{f}_q}}{\sum_k e^{\vec{\lambda}_k \vec{f}_q}} \left[(1-\pi) \frac{(e^{\vec{\omega} \vec{f}_{qa}})^r}{1+e^{\vec{\omega} \vec{f}_{qa}}} + \pi \frac{(e^{\vec{w}_c \vec{f}_{qa}})^r}{1+e^{\vec{w}_c \vec{f}_{qa}}} \right] - \alpha \left(\|\|\vec{\omega}\|\|^2 + \sum_k^K (\|\|\vec{w}_k\|\|^2 + \|\|\vec{\lambda}_k\|\|^2) \right)$

Table 1: Objective functions used by the proposed models.

$P(r|q,a,c)$ has a separate logistic regression model:

$$P(r|q,a,c) = \frac{(e^{\vec{w}_c \vec{f}_{qa}})^r}{1 + e^{\vec{w}_c \vec{f}_{qa}}} \tag{3}$$

where f_{qa} is the feature vector constructed from the question q, and answer snippet a.

Our models, including the LRM, are inherently conditional since they model $P(r|q,a)$ and not the joint probability $P(r,q,a)$. However, conditioned on the question, q and the answer-snippet a, there is a step-by-step generative process by which they make predictions, r. This process for LRM is shown in Figure 2a. To avoid problems of over-fitting, we regularize the model weights using L2 regularization. The complete optimization objective used by this model is given in Table 1. We learn the values for model parameters, $\vec{w}_c$ and $\vec{\lambda}_c$ during training.

Glocal Logistic Regression Mixture (G-LRM).

The previous model learns separate logistic regression weights for each cluster. However, there can be some features which are cluster independent and are equally important (or unimportant) irrespective of the cluster assignment of the question. A measure of lexical overlap between question and answer snippet could be one example of such a feature. The previous model would need to repeatedly learn independent weights for this feature for each cluster. We reduce this extra effort in our second model, G-LRM, (shown in Figure 1b and Figure 2a) by allowing one cluster independent global weight vector, $\vec{\omega}$, and several cluster specific local weights, $\vec{w}_c$:

$$P(r|q,a,c) = \frac{(e^{\vec{\omega} \vec{f}_{qa} + \vec{w}_c \vec{f}_{qa}})^r}{1 + e^{\vec{\omega} \vec{f}_{qa} + \vec{w}_c \vec{f}_{qa}}} \tag{4}$$

where, as before, f_{qa} is the feature vector constructed using the question q, and answer snippet a and the weights are L2 regularized. The complete optimization objective used by this model is shown in Table 1. In addition to $\vec{w}_c$ and $\vec{\lambda}_c$, this model also learns the value for $\vec{\omega}$ during training.

Mixture Glocal Logistic Regression Mixture (MG-LRM).

In this model we propose another way of incorporating the global and the local weight vectors. We model the relevance probability using a mixture of two logistic regression models: a global cluster independent model $P_g(r|q,a)$, with weight $\vec{\omega}$ and K different local cluster specific models $P_l(r|c,q,a)$, with weights $\vec{w}_c$. The mixture membership is controlled using π which is learned during training.

$$\begin{aligned} P(r|q,a,c) &= [(1-\pi)P_g(r|q,a) + \pi P_l(r|c,q,a)] \\ &= \left[(1-\pi) \frac{(e^{\vec{\omega} \vec{f}_{qa}})^r}{1 + e^{\vec{\omega} \vec{f}_{qa}}} + \pi \frac{(e^{\vec{w}_c \vec{f}_{qa}})^r}{1 + e^{\vec{w}_c \vec{f}_{qa}}} \right] \end{aligned} \tag{5}$$

Table 1 shows the complete equation used by the model, the plate diagram of this model is shown in Figure 1c and the story for making the relevance prediction is shown in Figure 2b. In addition to the latent cluster assignment variable, c, this model assumes another latent variable, h, which is binary in nature and helps in choosing between the global model with $\vec{\omega}$ as weights when $h = 0$ and the cluster specific model which use $\vec{w}_c$ as weight when $h = 1$.

3.3 Training

We train the three models by maximizing the log-likelihood of the data. Since the log likelihood function is non-convex, we use Expectation-Maximization [12] for training. During the E-step we compute the expectations for latent variable assignments using parameter values from the previous iteration and in the M-step, given the expected assignments we maximize the expected log complete likelihood with respect to the model parameters.

Notations.

Our description of EM algorithm uses the following notations: N refers to the total number of training instances and a subscript of n represents the nth training instance; c_n^k is a notation for the nth instance getting assigned to the kth cluster; $<>$ refers to the expected value and R is a general shorthand for regularizer terms. A symbol with$\vec{\ }$represents a vector and one in bold face represents a collection of vectors. The regularizer terms for the three different models are shown in Table 1. Also, $P(r_n|q_n, a_n, c_n^k)$ in the above equations is modeled using Equations 3, 4 and 5 for LRM, G-LRM and MG-LRM respectively.

LRM and G-LRM:

.

E-Step:

$$< c_n^k > \propto \frac{e^{\vec{\lambda}_k \vec{f}_{q_n}}}{\sum_{k'}^K e^{\vec{\lambda}_{k'} \vec{f}_{q_n}}} P(r_n|q_n, a_n, c_n^k)$$

M-Step objective:

$$\begin{aligned} < L >_R &= \sum_n^N \sum_k^K < c_n^k > (\log \frac{e^{\vec{\lambda}_k \vec{f}_{q_n}}}{\sum_{k'}^K e^{\vec{\lambda}_{k'} \vec{f}_{q_n}}} \\ &+ \log P(r_n|q_n, a_n, c_n^k)) - R \end{aligned}$$

MG-LRM:.

For MG-LRM, we have two latent variables: c for cluster assignment and h for the choice between the global relevance model and the cluster-specific relevance model. Using $z_n(c, h)$ to represent a configuration of $c \in 1...K$, and $h \in 0, 1$ assignments for the nth instance and $\delta_a(x)$ for the Kronecker delta function which takes the value of 1 whenever $x = a$ and 0 otherwise, EM could be performed using the following equations:
E-Step:

$$\begin{aligned} < z_n^{k,h} > \;\; \propto \;\; & \frac{e^{\vec{\lambda}_k \vec{f}_{q_n}}}{\sum_{k'}^K e^{\vec{\lambda}_{k'} \vec{f}_{q_n}}} ((1-\pi) P_g(r_n|q_n, a_n))^{\delta_0(h_n)} \\ & (\pi P_l(r_n|q_n, a_n, c_n^k))^{\delta_1(h_n)} \end{aligned}$$

M-Step objective:

$$\begin{aligned} < L >_R = \sum_n^N \sum_k^K \sum_{h \in \{0,1\}} & < z_n^{k,h} > (\log \frac{e^{\vec{\lambda}_k \vec{f}_{q_n}}}{\sum_{k'}^K e^{\vec{\lambda}_{k'} \vec{f}_{q_n}}} \\ & + \delta_h(0) \log(1-\pi) \frac{(e^{\vec{\omega} \vec{f}_{qa}})^r}{1 + e^{\vec{\omega} \vec{f}_{qa}}} \\ & + \delta_h(1) \log \pi \frac{(e^{\vec{w}_c \vec{f}_{qa}})^r}{1 + e^{\vec{w}_c \vec{f}_{qa}}}) - R \end{aligned}$$

Optimization:.

For LRM and MG-LRM, the M-step objective functions are convex in the parameters $\vec{w}_c$ and $\vec{\omega}$ and π (where applicable). Therefore, in the M-step of these models, we have used BFGS [14, 15] to minimize the negative of the objective function.

However, for G-LRM, the M-step objective is bi-convex in $\vec{w}_c$ and $\vec{\omega}$. Consequently, we adopt an alternate optimization strategy where the a single iteration of EM has two steps. In the first step, we compute $< z_n^{k,h} >$ and optimize for the global variable $\vec{\omega}$ using BFGS algorithm, keeping the local variables $\vec{w}_c$ and $\vec{\lambda}_c$ constant. In the next step we recompute $< z_n^{k,h} >$ and optimize the local variables $\vec{w}_c$ and $\vec{\lambda}_c$ using the BFGS algorithm, keeping the global variable $\vec{\omega}$ constant.

3.4 Feature Engineering

Questions and candidate snippets are analyzed by our information extraction pipeline [13], which extracts entity mentions, performs within-document and cross-document coreference, detects relations between entity mentions, compute parse trees, and assigns semantic roles to constituents of the parse tree. The features used for relevance prediction are an extension of those used in the [28]. We have engineered two classes of features – the first set of features are derived from a syntactic and semantic analysis of the question only. The second class of features attempt to capture the relevance of the snippet to the query.

3.4.1 Question Clustering Features

These features aim to capture the category of the question. Several of these features rely on the output of our parser that annotates the question with a Penn-tree-style parse[2].

1. The first word in the question.

2. The part-of-speech (POS) of the first word in the question.

3. The first WH-word or Verb if no WH-word is found. This feature captures words like *Who, List, What, Describe* etc. This feature can be different from the one in (1) as there are some questions where the WH word is not the first word, e.g, in the question *List reasons why married people have affairs.*, the first word is *List* but the first WH-word is *why*.

4. The POS tag for the first WH-word found before.

5. The Noun Phrase to the right of the first WH-word or Verb found before.

6. A binary feature that captures whether a $PP \rightarrow IN\_NP$ structure exists.

7. The PP word in the $PP \rightarrow IN\_NP$ structure if it exists.

8. If the Noun Phrase in the $PP \rightarrow IN\_NP$ is a named entity two additional features are generated (a) a categorical feature indicating the type of the named entity is generated (b) a feature where the form of the PP is lexicalized capturing patterns like "about:PERSON", "against:PEOPLE", "between:ORGANIZATION" and so on.

9. A number of dictionary based features were used. The dictionaries captured "How Words", "Effect Words", "Relationship words" and so on.

3.4.2 Snippet Features

These features are used to capture the quality of a snippet as a possible candidate answer independent of the query.

- Structural features, such as the length of the snippet.

- Grammatical features, capturing aspects of the snippet ranging from the presence of a verb, the presence of a quotation, or the detection of whether the snippet contains a question.

[2] `http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

- Garbage-detection features, that help the models reject, for instance, snippets with excessive profanity.

- Entity mention features, such as features that compute the density of entity mentions in a snippet–for example weather reports or professional team sport result reports have a high density of entity mentions.

- Lexical features, such as dictionary based features that detect whether the snippet contains words that belong to specific dictionaries.

3.4.3 Question and Snippet Features

These features capture similarities between the question and the candidate snippet.

- Text matching features, that compute the overlap score between the query and the snippet using a broad variety different approaches.

- Entity matching features, that match entities in the question with entities in the snippet, and that account for proxies, such as matching "the Obama Administration" with "the U.S. Government".

- Relation matching features, that compare the relations between entities in the question to relations between entities in the snippet and correspondingly produce a match score.

- Event matching features, where an event is represented by: a collection of entity mentions; a collection of verbs that act as "anchors" of the event; a set of relations between the entities and the verbs; and a set of relations between the entities. Event matching features compute similarity scores between events detected in the question and events that appear in the candidate answer.

- Syntactic features, such as the similarity score between syntactic dependencies in the question and in the snippet, and features that match portions of the parse trees of the question and of the candidate snippet.

- Semantic features, computed from the semantic roles associated with constituents of the query and with those of the snippet.

4. EMPIRICAL EVALUATION

In this section we present results for the empirical evaluation of each of the three proposed models.

4.1 Datasets

In our experiments we have used two distinct datasets of non-factoid questions and a pre-retrieved pool of relevant and irrelevant answers, retrieved from a large collection by an IR engine. In both datasets, each question has multiple answer snippets, and each answer snippet has been manually annotated as relevant or irrelevant. $1/6^{th}$ of these questions (along with the corresponding answer snippets) were randomly selected as the test set, and the remaining were used for training and development.

BOLT Dataset.

[3] Participants in the DARPA BOLT task must build systems to answer questions using a corpus of posts collected from threaded Internet discussion forums in 3 languages: Arabic, Chinese and English.

The training and test set for the relevance models were created by us over the course of more than one year. Professional annotators first generate queries by browsing the corpus using a search tool and find questions that have more than 2 relevant answers in the corpus. According to the official BOLT Information Retrieval (IR) task rules, annotators cannot generate questions that (1) require reasoning or calculation over the data to compute the answers; (2) are vague or ambiguous; (3) can be broken into multiple disjoint questions; (4) are multiple choice questions; and (5) are factoid questions—the kinds that have already been well studied in TREC [36]. Any other kind of question is allowed.

The top snippets in response to the queries returned by a system being developed by our consortium for the BOLT IR task are judged for relevance by human annotators. Each annotator judges candidate answers to questions generated by different annotators, to reduce possible biases. This 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. As data is accumulated, the relevance model is intermittently retrained and redeployed to collect more data. The process was iterated for several months, and produced 455 questions and more than 62,000 annotated snippets, out of which about 23.5% are relevant answers. These question-answers pairs are used to train and test the final relevance model used in our question-answering system. The train set consists of 387 questions and 48,000 question-snippet pairs while the test set has 9500 question-snippet pairs consisting of 65 distinct questions. All the 455 questions were also annotated with one of the following ten *Expected Answer Types*: {*Action-of-X-on-Y, Effect-of-X-on-Y, Experience-of-X-with-Y, How-to-do-X, Relationship-of-X-with-Y, Statement-of-X-on-Y, Persons-List, Organizations-List, Locations-List* and *Other-List*}. Note that these Answer Types were not used in any of our proposed models, but were used to train the baseline baseline system and directly fed to the oracle system we use in the evaluation of our algorithms.

TAC Dataset.

This is a publicly available dataset from the Question-Answering track of the TAC-2008 task [4]. TAC-2008 consists of two sets of questions: the *Rigid list* and the *Squishy List*. Out of the two, the *Squishy List* consists of non-factoid questions. There were more than 26,000 question-answer pairs each labeled with the number of relevant 'nuggets' present in the answer snippet. Since we needed binary annotations, all answers with at least one relevant nugget were marked relevant. Additionally, since the skewed ratio of positive to negative class in this dataset made learning difficult, we resampled the relevant classes so that the total number of instances remained the same but the percentage of relevant snippets increased from 14% to 50%. The dataset consisted of 89 distinct questions in this list and they were not annotated with an Answer Type. Out of 89, 71 questions were used for training and the remaining 13 were used during testing. The training set consisted of 22,000 instances of question-snippet pairs while the test set had 4000 pairs.

Select representative questions from the BOLT and TAC datasets are displayed in Table 2.

4.2 Baselines

We have compared our models with following baselines:

LR: This is our primary baseline and is a simple logistic regression

[3] BOLT is the acronym for the DARPA Broad Operational Language Translation program, see http://www.darpa.mil/Our_Work/I2O/Programs/Broad_Operational_Language_Translation_(BOLT).aspx

[4] http://www.nist.gov/tac/2008/qa/

Question	Answer Type
Find statements expressing suspicions about the murder of Rafik al-Hariri.	Statement-on-X
Describe the relationship of former government adviser Lewis Libby to CIA agent Valerie Plame.	Relationship-of-X-with-Y
Why do women like to hide their real age?	Why
What are some experiences people have had as tourists in New York City?	Experience-of-X-with-Y
Is abortion ethical?	Debates
What are the effects of Beijing Olympic games on China?	Effects-of-X-on-Y
How did China react to the 2011 earthquake in Japan?	Action-of-X-regarding-Y
How does one start a garden?	How-to
What reasons do adherents have for accepting intelligent design?	-
Why do people like George Clooney?	-
What complaints are made about Chinaś one-child per family law?	-
What actions by Wolfowitz as President of the World Bank are praised?	-
What was praised in the performance of the New York Philharmonic?	-

Table 2: Above: Representative questions in the BOLT dataset and their corresponding Answer-types. Below: Representative questions in the TAC Squishy questions dataset. Note that Answer-Types annotations are not available in the TAC dataset.

Model	BOLT Data				TAC Data			
	K	P	R	F	K	P	R	F
Oracle	–	63.73	42.33	50.87	–	–	–	–
LR	–	61.33	36.79	45.99	–	24.79	65.66	35.99
LR AnsTypeTruth	–	50.12	44.69	47.25	–	–	–	–
J48 [28]	–	44.60	43.20	43.88	–	–	–	–
LRM	4	59.36	39.03	47.09	9	24.43	71.41	36.40
G-LRM	4	59.97	40.36	48.25	2	24.41	71.80	36.44
G-LRM + Init	7	61.01	42.15	**49.85**	13	24.34	71.80	36.40
MG-LRM	6	59.66	41.75	49.12	18	26.31	58.88	36.37
MG-LRM + Init	7	64.48	39.43	48.93	2	24.53	72.98	**36.72**

Table 3: Performance comparison of the proposed models, LRM, G-LRM and MG-LRM, with the two baselines and the oracle. Their performances are better than the baseline LR and J48 [28] and close to or better than the LR AnsTypeTruth and the oracle which use additional human annotated information. A '+init' represents a model variation where EM was initialized with baseline LRM's vectors.

agnostic to the Answer Type categorization or clustering behavior. Given a question-answer pair it predicts the answer's binary relevance. The model uses the same features as used by the relevance prediction module of the proposed models.

J48 This model, proposed by Luo *et al.* [28], is a decision-tree classifier that, in addition to the standard features used in the LR model , uses the predicted Answer Type of the question as an additional feature. The Answer Type of the question is predicted using an classifier that employs the same features as the question-clustering component $P(c|q)$ in all our models, described in Section 3.4.1. The Answer Type classifier was trained on a set of questions different from those used to train the relevance prediction component.

LR-AnsTypeTruth Since our models produce question-cluster-specific relevance models, we introduce an additional baseline that uses the manual Answer Type annotations for the questions. During training, the questions were divided into distinct subsets using the manually tagged Answer Type annotations, and independent Logistic Regression based relevance models were constructed for each Answer Type. During the testing phase, a test question was classified into one of the Answer Types using the Answer Type classifier described above, and the corresponding relevance model was used to make relevance prediction for the question-answer pair. This is not our primary baseline because this model uses oracle-style information, that is, the manually generated Answer Type annotations which are not available to our proposed models and to the first two comparison systems.

Oracle We further exploit the human Answer-Type annotations to construct an upper bound to our models. As in case of *LR-AnsTypeTruth*, the train questions were segregated into classes and independent Logistic Regression models were constructed for each class. During testing, instead of using the classifier's prediction of the Answer-Type as in *LR-AnsTypeTruth*, the oracle uses the test question's manually tagged Answer-Type annotation to select the corresponding Logistic Regression model to be used for relevance prediction.

4.3 Experimental Results

Since we are more interested in our performance on identifying relevant answers only, we evaluate our models based on the F1 score for the relevant class. Each of our models involves two parameters, namely the number of clusters (K) and the regularization coefficients, α. The optimal values of these parameters were selected using 5-fold Cross Validation on the training set. Table 3 presents the performances of the proposed models and the baselines on the two test sets. For all models and baselines, model selection was done using 5-fold cross validation on the respective training sets. The three proposed models were trained with random initialization for EM. Additionally, training for G-LRM and MG-LRM were also initialized with the baseline LR's weight vectors and this is represented by a '+ Init' in the table. There is a '–' in the K column for the baselines and the oracle because these models are cluster independent. Also, as the TAC dataset lacks the manual Answer-Type annotations, the table does not include results for the

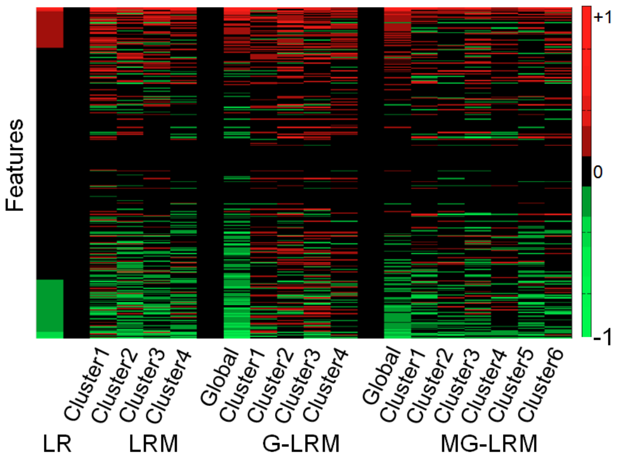
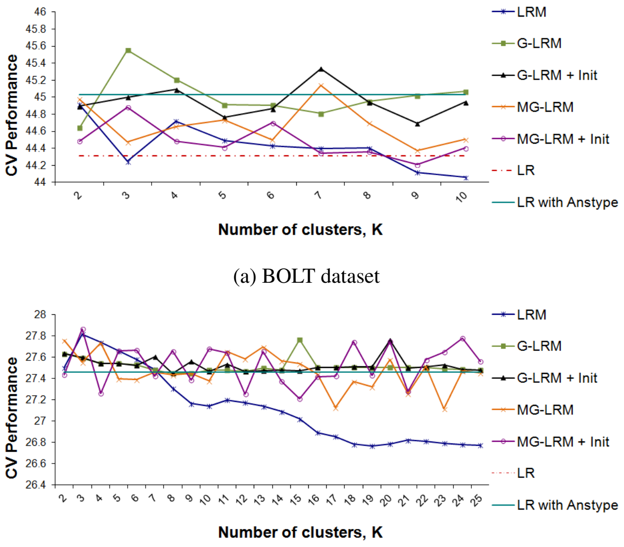


Figure 3: Global and cluster weights visualization for the various models. Each colored column represents a weight vector and each row is a feature. Bright red colors indicate high positive weights, bright green color represents high negative weights, and black denotes weights close to zero. Black columns were used as visual cues to better separate the weights of the four models. Features are sorted in decreasing weights as learnt by the logistic regression model. The cluster weights for a particular model look different from each other and from the LR weights and the global weight of the model (if applicable). On the other hand, for G-LRM and MG-LRM, LR weight vector is more similar to the global weight vector than to any of the cluster specific weights of the same model.



(a) BOLT dataset

(b) TAC dataset

Figure 4: Cross validation performances of the various models with increasing number of clusters, K, for the two datasets. There is little variation in models' performances with increasing K values

LR ANsTypeTruth and the Oracle models, since they need these annotations.

The table shows that all the three models, with or without random initialization, outperform the baseline *J48* and *LR* models. Interestingly, for the BOLT dataset, their performance is also mostly better than that of *LR AnsTypeTruth* and comparable to the Oracle even though they do not use the additional Answer Type annotations.

A possible explanation for our models outperforming the baseline model which uses a manually designed Answer Type taxonomy is that while designing the taxonomy, humans experts analyze only a small sample of the hundreds of questions present in the dataset. This sample might not be sufficiently representative of the dataset leading to incomplete or sub-optimal taxonomies. The order in which the questions are seen by the taxonomy designers might also possibly bias the taxonomy design process. On the other hand, our models have a bird's eye view of the data and discover the clusters in a data driven manner and hence, the clustering (equivalent to taxonomy) might be statistically more robust, leading to better performances.

The improvements on the TAC dataset are consistent, but not as pronounced as on the BOLT dataset. This could be due to the nature of the TAC dataset – the questions not being very diverse, thus reducing the usefulness of the cluster specific models – or to limitations of the clustering features, f_q , used by the clustering components, which were developed while working on the BOLT program and might fail to capture important characteristics of the TAC Squishy questions.

4.4 Visual Exploration of Clusters

Our models are based on the assumption that questions belong to different clusters and that each cluster uses features differently to make relevance predictions: features important for the cluster-specific relevance model in a cluster might be irrelevant for the model of a different cluster. We qualitatively evaluate our assumption using a visualization approach. Figure 3 shows a heat map of all feature weights on the BOLT dataset for the baseline Logistic Regression and the three proposed models (after model selection and with random EM initializations).

The leftmost column represents the weight vector learnt by the baseline Logistic Regression (LR). The next four columns show the four weight vectors for the LRM model, corresponding to the number of clusters, $K = 4$, automatically determined by the model selection procedure. We see that the four columns look different from each other and from LR weights. We also notice bright green color on top, which indicates that the clusters are assigning high positive weights to features that, in general, were weighted negatively by the LR model. Similarly, the bottom portion of these four columns contains shades of red color representing features that were positively weighted by the clusters but not by LR. This indicates that the LRM model is indeed learning different weight vectors for different clusters which, in turn, are different from the cluster-agnostic Logistic Regression model.

Similarly, the next 5 columns represent the global weight vector, ω, and the four cluster specific weights, w_c learnt by the G-LRM. We see that the global vector looks similar to LR's weight vector since both of them are cluster-agnostic. On the other hand, the local cluster weights, w_c, look very different from each other and from the global and LR's weights. Similar behavior can be observed for the global and six local weights learnt by the MG-LRM.

We conclude that the weights learnt by the models are dissimilar to those of the baseline logistic regression model. Also, as expected, the global weights look more similar to the logistic regression model's weight than the cluster-specific weights.

4.5 Choice of number of clusters

Our models assume that the questions in the dataset can be grouped into K distinct clusters and that each cluster has a distinct relevance prediction model as well. Hence, it is reasonable to assume that

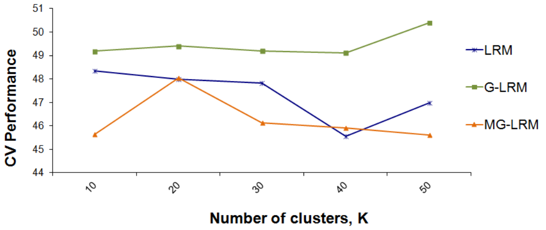


Figure 5: Cross validation performances of the three proposed models don't vary considerably even for high values of number of cluster, K, for the BOLT dataset

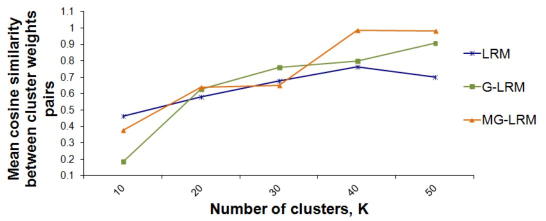


Figure 6: Mean cosine similarity between pairs of cluster weights increases with number of clusters indicating that forcing the models to learn high number of clusters results in highly similar clusters

there is an optimal value for K, below or above which the model under-fits or over-fits the data respectively.

Interestingly, our experiments with measuring our models' performances as a function of K did not exhibit any such pattern. Figure 4 shows that there is no single peak in model performances and also, the variation among performances for different values of K is not huge. In fact, as depicted by Figure 5, the performances of the three proposed models do not vary considerably even for values of K as high as 40 or 50.

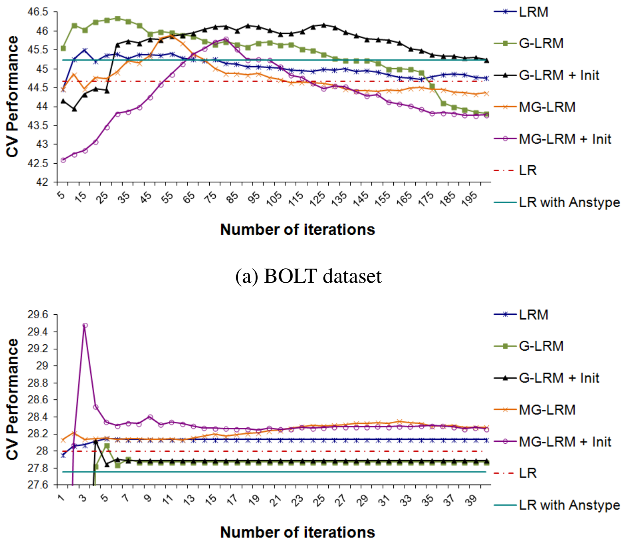
In order to investigate this unexpected behavior further, we plot the average cosine similarity between the weights-vectors of all pairs of clusters as a function of K in Figure 6. The plot illustrates that as K grows, the cluster weight vectors become increasingly similar on average. This indicates that if the model is forced to learn high number of clusters, it learns highly correlated clusters.

We conclude that the models are not very sensitive to the choice of the number of clusters. On the one hand, this is a positive result: the models do not require a fine tuning of K. On the other hand, this can make it difficult to assign semantic meaning to the clusters.

4.6 EM Convergence and Sensitivity

Our models are trained using Expectation Maximization which often requires several iterations to converge. Hence, as a practical consideration, it is important to analyze the sensitivity of the models' performances to the number of EM iterations.

Figure 7 shows a plot of the mean-cross-validation performances of the tuned models as a function of number of iterations. For most models, the performance increases as the model learns good weights and then stabilizes at a slightly lower value, which can be attributed to the opposing effects of over-fitting and of the stabilizing effect of the regularization coefficients. In particular, in Figure 7a we see that for MG-LRM, the peak appears at a higher number of iterations than the other models. We conjecture that it is due to the higher complexity of this model: for example, if the initialization favored the global component (low initial π value), especially likely in MG-LRM+Init, then several iterations would be needed to increase π enough to learn good cluster weights.



(a) BOLT dataset

(b) TAC dataset

Figure 7: Cross validation performance vs. training time for the two datasets The performances improve as the models get trained, achieves a peak and then stabilizes.

In conclusion, our models are not sensitive to the setting of number of iterations, as long as we run them beyond a minimum number, which is around 75 iterations for the BOLT dataset and 5 iterations for the TAC dataset.

4.7 Regularization Coefficients

The proposed models have two parameters that need to be specified, namely the regularization coefficient, α (see Table 1) and the number of clusters, K. In the rest of this section we study the effect of these parameters on the models' performances.

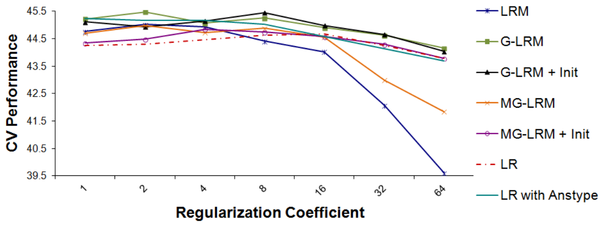
Figure 8 shows the variation in performances of the models with changing values of the regularization coefficients for the two datasets. Since the proposed models have another parameter i.e. K, the number of clusters, for the purpose of obtaining a plot, we experimented with several values of K for a particular value of regularization coefficient and used the median for the plot.

The figure shows that the performances of the various models are lower for very high and very low values of the regularization coefficient. This is as expected, and represent regions of the model underfitting and overfitting the data respectively.

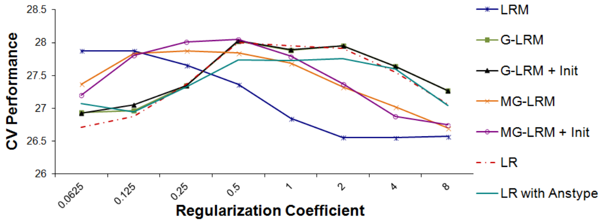
5. CONCLUSION AND FUTURE WORK

In this paper we have addressed the problem of detecting if a given snippet answers a given non-factoid question. The gist of our approach is the assumption that, like factoid questions, non-factoid questions belong to distinct categories related to the Answer Type they elicit, and that relevance prediction models can benefit from using these categories. We recognize that building a universal, domain-independent taxonomy of general questions would be extremely difficult, and that constructing taxonomies for individual specific domains would be expensive.

To address this challenge, we propose three probabilistic models for joint question-clustering and relevance-prediction, that identify



(a) BOLT dataset



(b) TAC dataset

Figure 8: Cross validation performances of the various models with increasing values of regularization coefficients, α, for the two datasets. The performances are lower for very low or high values of α because of under-fitting and over-fitting respectively.

these categories in a data-driven manner and build category-specific models. In a nutshell, these models differ in how explicitly they capture the category-specific posterior probability of relevance and how they account for the contribution of category-agnostic features. Since they treat categories as latent variables, our models do not require an explicit taxonomy of question type, a characteristic that makes them applicable irrespectively of the domain.

Our experiments, conducted on two different datasets of non-factoid questions and candidate answers extracted from social media, reveal that our approaches are better than baseline models built on the same features that do not account for the latent categories. Their performances also approach that of oracle which uses manually labeled question categories as input.

In our future work we intend to experiment with model-selection criteria to automatically select the number of clusters, and to introduce a penalty in the objective function that discourages learning highly similar clusters. We also intend to experiment with different regularization criteria that favor sparsity in the learned weight vector; we conjecture that it might be possible to explore the underlying semantics of the clusters that have a small number of features with high weights. Also, the present models are not personalized. In future, it would be interesting to learn user-specific clusters of questions using the user's search behavior and preferences to provide a more personalized experience. The data-driven approach of our models make them an ideal candidate to adapt to a large number of users with varied preferences.

Acknowledgments

The authors would like to thank Dr. Salim Roukos for his valuable suggestions and for providing the necessary resources for developing labeled data for relevance as well as for answer-types.

6. REFERENCES

[1] Bing feature update: Searching for a good deal? new natural language capabilities in bing shopping understand prices. http://www.bing.com/blogs/site_blogs/b/search/archive/2011/03/01/bing-feature-update-searching-for-a-good-deal-new-natural-language-capabilities-in-bing-shopping-understand-prices.aspx.

[2] Meet hummingbird: Google just revamped search to answer your long questions better. http://www.forbes.com/sites/roberthof/2013/09/26/google-just-revamped-search-to-handle-your-long-questions.

[3] N. Aikawa, T. Sakai, and H. Yamana. Community QA Question Classification: Is the Asker Looking for Subjective Answers or Not? *IPSJ Online Transactions*, 4:160–168, 2011.

[4] S. M. Beitzel, E. C. Jensen, O. Frieder, D. Grossman, D. D. Lewis, A. Chowdhury, and A. Kolcz. Automatic web query classification using labeled and unlabeled training data. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 581–582, New York, NY, USA, 2005. ACM.

[5] S. M. Beitzel, E. C. Jensen, O. Frieder, D. D. Lewis, A. Chowdhury, and A. Kolcz. Improving automatic query classification via semi-supervised learning. In *Proceedings of the 5th IEEE International Conference on Data Mining*, ICDM '05, pages 42–49, Washington, DC, USA, 2005. IEEE Computer Society.

[6] M. S. Bernstein, J. Teevan, S. Dumais, D. Liebling, and E. Horvitz. Direct answers for search queries in the long tail. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 237–246, New York, NY, USA, 2012. ACM.

[7] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, Sept. 2002.

[8] F. Bu, X. Zhu, Y. Hao, and X. Zhu. Function-based question classification for general qa. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1119–1128, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[9] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-aware query classification. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 3–10, New York, NY, USA, 2009. ACM.

[10] L. Chen, D. Zhang, and L. Mark. Understanding user intent in community question answering. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 823–828, New York, NY, USA, 2012. ACM.

[11] M. Chen, J.-T. Sun, X. Ni, and Y. Chen. Improving context-aware query classification via adaptive self-training. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 115–124, New York, NY, USA, 2011. ACM.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977.

[13] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. A statistical model for multilingual entity detection and tracking. In D. M. Susan Dumais and S. Roukos, editors, *HLT-NAACL 2004:*

Main Proceedings, pages 1–8, Boston, MA, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[14] P. E. Gill and W. Murray. *Minimization Subject to Bounds on the Variables*. NPL Report NAC72, 1976.

[15] P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1981.

[16] N. Goharian and S. S. Mengle. Context aware query classification using dynamic query window and relationship net. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 723–724, New York, NY, USA, 2010. ACM.

[17] U. Hermjakob. 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, 2001. Association for Computational Linguistics.

[18] R. Higashinaka and H. Isozaki. Corpus-based question answering for why-questions. In *In Proceedings of IJCNLP*, pages 418–425, 2008.

[19] E. Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, and D. Ravichandran. 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, 2001. Association for Computational Linguistics.

[20] Z. Huang, M. Thint, and Z. Qin. Question classification using head words and their hypernyms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 927–936, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[21] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 387–396, New York, NY, USA, 2006. ACM.

[22] A. Lally, J. M. Prager, M. C. McCord, B. Boguraev, S. Patwardhan, J. Fan, P. Fodor, and J. Chu-Carroll. Question analysis: How watson reads a clue. *IBM Journal of Research and Development*, 56(3):2, 2012.

[23] E. H. Laurie, L. Gerber, U. Hermjakob, M. Junk, and C. yew Lin. Question answering in webclopedia. In *In Proceedings of the Ninth Text REtrieval Conference (TREC-9*, pages 655–664, 2000.

[24] M. Le Nguyen, T. T. Nguyen, and A. Shimazu. Subtree mining for question classification problem. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 1695–1700, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[25] B. Li, Y. Liu, and E. Agichtein. Cocqa: Co-training over questions and answers with an application to predicting question subjectivity orientation. In *EMNLP*, pages 937–946. ACL, 2008.

[26] B. Li, Y. Liu, A. Ram, E. V. Garcia, and E. Agichtein. Exploring question subjectivity prediction in community qa. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 735–736, New York, NY, USA, 2008. ACM.

[27] X. Li and D. Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[28] X. Luo, H. Raghavan, V. Castelli, S. Maskey, and R. Florian. Finding What Matters in Questions. In *Proceedings of NAACL-HLT*, pages 878–887, 2013.

[29] D. Moldovan, S. Harabagiu, A. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, V. Rus, and I. Background. The structure and performance of an open-domain question answering system. In *In Proceedings of the Conference of the Association for Computational Linguistics (ACL-2000*, pages 563–570, 2000.

[30] A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. Exploiting syntactic and shallow semantic kernels for question answer classification. In *In Proc. of ACL-07*, pages 776–783, 2007.

[31] J.-H. Oh, K. Torisawa, C. Hashimoto, T. Kawada, S. De Saeger, J. Kazama, and Y. Wang. Why question answering using sentiment analysis and word classes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 368–378, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[32] B. Qu, G. Cong, C. Li, A. Sun, and H. Chen. An evaluation of classification models for question topic categorization. *Journal of the American Society for Information Science and Technology*, 63(5):889–903, 2012.

[33] M. Razmara and L. Kosseim. Answering list questions using co-occurrence and clustering. In *LREC*, 2008.

[34] R. Srihari and W. Li. A question answering system supported by information extraction. In *Proceedings of the sixth conference on Applied natural language processing*, ANLC '00, pages 166–172, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[35] S. Verberne, L. Boves, N. Oostdijk, and P.-A. Coppen. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 735–736, New York, NY, USA, 2007. ACM.

[36] E. M. Voorhees. Overview of the TREC 2004 question answering track. In *TREC*, 2004.

[37] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM.

[38] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Clustering user queries of a search engine. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 162–168, New York, NY, USA, 2001. ACM.

[39] T. C. Zhou, X. Si, E. Y. Chang, I. King, and M. R. Lyu. A data-driven approach to question subjectivity identification in community question answering, 2012.