Function-based question classification for general QA

Fan Bu, Xingwei Zhu, Yu Hao and **Xiaoyan Zhu**
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Sci. and Tech., Tsinghua University
buf08@mails.tsinghua.edu.cn
etzhu192@hotmail.com
haoyu@mail.tsinghua.edu.cn
zxy-dcs@tsinghua.edu.cn

Abstract

In contrast with the booming increase of internet data, state-of-art QA (question answering) systems, otherwise, concerned data from specific domains or resources such as search engine snippets, online forums and Wikipedia in a somewhat isolated way. Users may welcome a more general QA system for its capability to answer questions of various sources, integrated from existed specialized sub-QA engines. In this framework, question classification is the primary task.

However, the current paradigms of question classification were focused on some specified type of questions, i.e. factoid questions, which are inappropriate for the general QA. In this paper, we propose a new question classification paradigm, which includes a question taxonomy suitable to the general QA and a question classifier based on MLN (Markov logic network), where rule-based methods and statistical methods are unified into a single framework in a fuzzy discriminative learning approach. Experiments show that our method outperforms traditional question classification approaches.

1 Introduction

During a long period of time, researches on question answering are mainly focused on finding short and concise answers from plain text for factoid questions driven by annual trackes such as CLEF, TREC and NTCIR. However, people usually ask more complex questions in real world which cannot be handled by these QA systems tailored to factoid questions.

During recent years, social collaborative applications begin to flourish, such as Wikipedia, Facebook, Yahoo! Answers and etc. A large amount of semi-structured data, which has been accumulated from these services, becomes new sources for question answering. Previous researches show that different sources are suitable for answering different questions. For example, the answers for factoid questions can be extracted from webpages with high accuracy, definition questions can be answered by corresponding articles in wikipedia(Ye et al., 2009) while community question answering services provide comprehensive answers for complex questions(Jeon et al., 2005). It will greatly enhance the overall performance if we can classify questions into several types, distribute each type of questions to suitable sources and trigger corresponding strategy to summarize returned answers.

Question classification (QC) in factoid QA is to provide constraints on answer types that allows further processing to pinpoint and verify the answer (Li and Roth, 2004). Usually, questions are classified into a fine grained content-based taxonomy(e.g. UIUC taxonomy (Li and Roth, 2002)). We cannot use these taxonomies directly. To guide question distribution and answer summarization, questions are classified according to their functions instead of contents.

Motivated by related work on user goal classification(Broder, 2002; Rose and Levinson, 2004) , we propose a function-based question classification category tailored to general QA. The category contain six types, namely **Fact**, **List**, **Reason**, **Solution**, **Definition** and **Navigation**. We will introduced this category in detail in Section 2.

To classify questions effectively, we unify rule-based methods and statistical methods into a single framework. Each question is splited into functional words and content words. We generate strict patterns from functional words and soft patterns from content words. Each strict pattern is a regular expression while each soft pattern is a bi-gram cluster. Given a question, we will evaluate its matching degree to each patterns. The matching degree is either 0 or 1 for strict pattern and between 0 and 1 for soft pattern. Finally, Markov logic network (MLN) (Richardson and Domingos, 2006) is used to combine and evaluate all the patterns.

The classical MLN maximize the probability of an assignment of truth values by evaluating the weights of each formula. However, the real world is full of uncertainty and is unnatural to be represented by a set of boolean values. In this paper, we propose fuzzy discriminative weight learning of Markov logic network. This method takes degrees of confidence of each evidence predicates into account thus can model the matching degrees between questions and soft patterns.

The remainder of this paper is organized as follows: In the next section we review related work on question classification, query classification and Markov logic network. Section 2 gives a detailed introduction to our new taxonomy for general QA. Section 4 introduces fuzzy discriminative weight learning of MLN and our methodology to extract strict and soft patterns. In Section 5 we compare our method with previous methods on Chinese question data from Baidu Zhidao and Sina iAsk. In the last section we conclude this work.

Although we build patterns and do experiments on Chinese questions, our method does not take advantage of the particularity of Chinese language and thus can be easily implemented on other languages.

2 Related Work

Many question taxonomies have been proposed in QA community. Lehnert (1977) developed the system QUALM based on thirteen conceptual categories which are based on a theory of memory representation. On the contrary, the taxonomy proposed by Graesser et al. (1992) has foundations both in theory and in empirical research. Both of these taxonomies are for open-domain question answering.

With the booming of internet, researches on question answering are becoming more practical. Most taxonomies proposed are focused on factoid questions, such as UIUC taxonomy (Li and Roth, 2002). UIUC taxonomy contains 6 coarse classes (**Abbreviation**, **Entity**, **Description**, **Human**, **Location** and **Numeric Value**) and 50 fine classes. All coarse classes are factoid oriented except **Description**. To classify questions effectively, Researchers have proposed features of different levels, such as lexical features, syntactic features (Nguyen et al., 2007; Moschitti et al., 2007) and semantic features (Moschitti et al., 2007; Li and Roth, 2004). Zhang and Lee (2003) compared five machine learning methods and found SVM outperformed the others.

In information retrieval community, researchers have described frameworks for understanding goals of user searches. Generally, web queries are classified into four types: **Navigational**, **Informational**, **Transactional** (Broder, 2002) and **Resource** (Rose and Levinson, 2004). Lee et al. (2005) automatically classify **Navigational** and **Informational** queries based on past user-click behavior and anchor-link distribution. Jansen and Booth (2010) investigate the correspondence between three user intents and eighteen topics. The result shows that user intents distributed unevenly among different topics.

Inspired by Rose and Levinson (2004)'s work in user goals classification, Liu et al. (2008) describe a three-layers cQA oriented question taxonomy and use it to determine the expected best answer types and summarize answers. Other than **Navigational**, **Informational** and **Transactional**, the first layer contains a new **Social** category which represents the questions that do not intend to get an answer but to elicit interaction with other people. **Informational** contains two subcategories **Constant** and **Dynamic**. **Dynamic** is further divided into **Opinion**, **Context-Dependent** and **Open**.

Markov logic network (MLN) (Richardson and Domingos, 2006) is a general model combining first-order logic and probabilistic graphical models in a single representation. Illustratively, MLN is a first-order knowledge base with a weight attached to each formula. The weights can be learnt ei-

TYPE	DESCRIPTION	EXAMPLES
1. Fact	People ask these questions for general facts. The expected answer will be a short phrase.	Who is the president of United States?
2. List	People ask these questions for a list of answers. Each answer will be a single phrase or a phrase with explanations or comments.	List Nobel price winners in 1990s. Which movie star do you like best?
3. Reason	People ask these questions for opinions or explanations. A good answer summary should contain a variety of opinions or comprehensive explanations. Sentence-level summarization can be employed.	Is it good to drink milk while fasting? What do you think of Avatar?
4. Solution	People ask these questions for problem shooting. The sentences in an answer usually have logical order thus the summary task cannot be performed on sentence level.	What should I do during an earthquake? How to make pizzas?
5. Definition	People ask these questions for description of concepts. Usually these information can be found in Wikipedia. If the answer is a too long, we should summarize it into a shorter one.	Who is Lady Gaga? What does the Matrix tell about?
6. Navigation	People ask these questions for finding websites or resources. Sometimes the websites are given by name and the resources are given directly.	Where can I download the beta version of StarCraft 2?

Table 1: Question Taxonomy for general QA

ther generatively (Richardson and Domingos, 2006) or discriminatively (Singla and Domingos, 2005). Huynh and Mooney (2008) applies -norm regularized MLE to select candidate formulas generated by a first-order logic induction system and prevent overfitting. MLN has been introduced to NLP and IE tasks such as semantic parsing (Poon et al., 2009) and entity relation extraction (Zhu et al., 2009).

3 A Question Taxonomy

We suggest a function-based taxonomy tailored to general QA systems by two principles. First, questions can be distributed into suitable QA subsystems according to their types. Second, we can employ suitable answer summarization strategy for each question type. The taxonomy is shown in Tab. 1.

At first glance, classifying questions onto this taxonomy seems a solved problem for English questions because of interrogative words. In most cases, a question starting with "Why" is for reason and "How" is for solution. But it is not always the case for other languages. From table 2 we can see two questions in Chinese share same function word "怎么样" but have different types.

In fact, even in English, only using interrogative words is not enough for function-based question classification. Sometimes the question content is crucial. For example, for question "Who is the current president of U.S. ?", the answer is "Barak Obama" and the type is **Fact**. But for question "Who is Barak Obama?", it will be better if we return the first paragraph from the corresponding Wiki article instead of a short phrase "current president of U.S.". Therefore the question type will be **Definition**.

Compared to Wendy Lehnert's or Arthur Graesser's taxonomy, our taxonomy is more practical on providing useful information for question

Question	怎么样做宫保鸡丁? How to cook Kung Pao Chicken?
Type	**Solution**
Question	大家觉得阿凡达怎么样? What do you think of Avatar?
Type	**Reason**

Table 2: Two Chinese questions share same function words but have different types

extraction and summarization. Compared to ours, The UIUC taxonomy is too much focused on factoid questions. Apart from **Description**, all coarse types in UIUC can be mapped into **Fact**. The cQA taxonomy proposed in Liu et al. (2008) has similar goal with ours. But it is hard to automatically classify questions into that taxonomy, especially for types **Constant**, **Dynamic** and **Social**. Actually the author did not give implementation in the paper as well. To examine reasonableness of our taxonomy, we select and manually annotate 5800 frequent asked questions from Baidu Zhidao (see Section 5.1). The distribution of six types is shown in Fig. 1. 98.5 percent of questions can be categorized into our taxonomy. The proportion of each type is between 7.5% and 23.8%.

The type **Navigation** was originally proposed in IR community and did not cause too much concerns in previous QA researches. But from Fig. 1 we can see that navigational questions take a substantial proportion in cQA data.

Moreover, we can further develop subtypes for each type. For example, most categories in UIUC

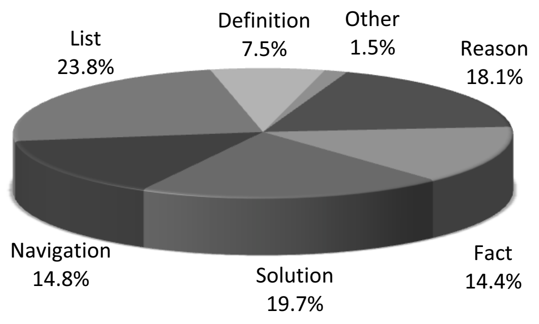

Figure 1: Distribution of six types in Baidu Zhidao data

taxonomy can be regarded as refinement to **Fact** and **Navigation** can be refined into **Resource** and **Website**. We will not have further discussion on this issue.

4 Methodology

Many efforts have been made to take advantage of grammatical , semantic and lexical features in question classification. Zhang and Lee (2003) proposed a SVM based system which used tree kernel to incorporate syntactic features.

In this section, we propose a new question classification methodology which combines rule-based methods and statistical methods by Markov logic network. We do not use semantic and syntactic features for two reasons. First, the questions posted on online communities are casually written which cannot be accurately parsed by NLP tools, especially for Chinese. Second, the semantic and syntactic parsing are time consuming thus unpractical to be used in real systems.

We will briefly introduce MLN and fuzzy discriminative learning in section 4.1. The construction of strict patterns and soft patterns will be shown in 4.2 and 4.3. In section 4.4 we will give details on MLN construction, inference and learning.

4.1 Markov Logic Network

A first-order knowledge base contains a set of formulas constructed from logic operators and symbols for predicates, constants, variables and functions. An *atomic formula* or *atom* is a predicate symbol. Formulas are recursively constructed from atomic formulas using logical operators. The *grounding* of a predicate (formula) is a replacement of all of its arguments (variables) by constants. A *possible world* is an assignment of truth values to all possible groundings of all predicates.

In first-order KB, if a possible world violates even one formula, it has zero probability. Markov logic is a probabilistic extension and softens the hard constraints by assigning a weight to each formula. When a possible world violates one formula in the KB, it is less probable. The higher the weight, the greater the difference in log probability between a world that satisfies the formula and a world does not. Formally, Markov logic network is defined as follows:

Definition 1 *(Richardson & Domingos 2004) A Markov logic network L is a set of pairs (,), where is a formula in first-order logic and is a real number. Together with a finite set of constants C = , it defines a Markov network as follows:*

1. *contains one binary node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground predicate is true, and 0 otherwise.*
2. *contains one feature for each possible grounding of each formula in L. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the associated with in L.*

There is an edge between two nodes of iff the corresponding grounding predicates appear together in at least one grounding of one formula in . An MLN can be regarded as a template for constructing Markov networks. From Definition 1 and the definition of Markov networks, the probability distribution over possible worlds specified by the ground Markov network is given by

MLN weights can be learnt generatively(Richardson and Domingos, 2006) or discriminatively(Singla and Domingos, 2005). In discriminative weight learning, ground atom set is partitioned into a set of evidence atoms and a set of query atoms . The goal is to correctly predict the latter given the former. In this paper, we propose fuzzy discriminative weight learning which can take the prior confidence of each evidence atom into account.

Formally, we denote the ground formula set by . Suppose each evidence atom is given with a prior confidence , we define a confidence function as follows. For each ground atom , if then we have , else . For each ground non-atomic formulas, is defined on standard fuzzy operators, which are

We redefined the *conditional likelihood* of given as

Where is the set of ground formulas involving query atoms, is the set of formulas with at least one grounding involving a query atom and is the sum of confidence of the groundings of the ith formula involving query atoms. The gradient of the conditional log-likelihood (CLL) is

(1)

By fuzzy discriminative learning we can incorporate evidences of different confidence levels into one learning framework. Fuzzy discriminative learning will reduce to traditional discriminative learning when all prior confidences equal to 1.

4.2 Strict Patterns

In our question classification task, we find function words are much more discriminative and less sparse than content words. Therefore, we extract strict patterns from function words and soft patterns from content words. The definition of content and function words may vary with languages. In this paper, nouns, verbs, adjectives, adverbs, numerals and pronouns are regarded as content words and the rest are function words.

The outline of strict pattern extraction is shown in Alg. 1. In line 3, we build template by removing punctuations and replacing each character in each content word by a single dot. In line 4, we generate patterns from the template as follows. First we generate n-grams(n is between 2 and) from

```
Algorithm 1: Strict Pattern Extraction
   Input: Question Set                ,
          Parameters      and
   Output: Pattern Set
 1 Initialize Pattern Set    ;
 2 for each Question    do
 3     String       =ReplaceContentWords(  ,  );
 4     Pattern Set      =GeneratePatterns(      ,  );
 5     for each Pattern   in       do
 6         if   in     then
 7             UpdateTypeFreq(  ,  );
 8         else
 9             Add   to   ;
10 Merge similar patterns in   ;
11 Sort    by Information Gain on type
   frequencies;
12 return top      Patterns in   ;
```

during which each dot is treated as a character of zero length. For coverage concern, if a generated n-gram is not start(end) with dot, we build another n-gram by adding a dot before(behind) and add both and into n-gram set. Then for each n-gram, we replace each consecutive dot sequence by '.*' and the n-gram is transformed into a regular expression. A example is shown in Tab. 3. Although generated without exhaustively enumerating all possible word combinations, these regular expressions can capture most long range dependencies between function words.

Each pattern consists of a regular expression as well as its frequency in each type of questions. Still

Question	在网上可以开通网银吗？ Can I launch online banking services on internet?	
Template	在..可以....吗	
Patterns (=4)	.*以.*吗	.*以.*吗.*
	.*可以.*	.*可以.*吗
	.*可以.*吗.*	.*在.*可以.*
	.*在.*可以.*吗.*	在.*可.*
	在.*可以.*	在.*可以.*吗
	.*在.*可.*	

Table 3: Strict patterns generated from a question

from Alg. 1, in line 5-9, if a pattern in question with type is found in , we just update the frequency of in , else is added to with only freq. equals to 1. In line 10, we merge similar patterns in . two patterns and are similar iff q Q matchP q p matchP q p , in which matchP is defined in Section 4.4.

Since a large number of patterns are generated, it is unpractical to evaluate all of them by Markov logic network. We sort patterns by information gain and only choose top "good" patterns in line 11-12 of Alg. 1. A "good" pattern should be discriminative and of wide coverage. The information gain IG of a pattern is defined as

$$\mathrm{IG}$$

$$- \qquad - \qquad -$$

in which is the number of question types, is the probability of a question having type , (or $^{-}$) is the probability of a question matching(or not matching) pattern . (or $^{-}$) is the probability of a question having type given the condition that the question matches(or does not match) pattern . These probabilities can be approximately calculated by type and pattern frequencies on training data. From the definition we can see that information gain is suitable for pattern selection. The more questions a pattern matches and the more unevenly the matched questions distribute among questions types, the higher IG will be.

4.3 Soft Patterns

Apart from function words, content words are also important in function-based question classification. Content words usually contain topic information which can be a good complement to function words. Previous research on query classification(Jansen and Booth, 2010) shows that user intents distribute unevenly among topics. Moreover, questions given by users may be incomplete and contain not function words. For these questions, we can only predict the question types from topic information.

Compared with function words, content words distribute much more sparsely among questions.

When we represent topic information by content words (or bi-grams), since the training set are small and less frequent words (or bi-grams) are filtered to prevent over-fitting, those features would be too sparse to predict further unseen questions.

To solve this problem, we build soft patterns on question set. Each question is represented by a weighted vector of content bi-grams in which the weight is bi-gram frequency. Cosine similarity is used to compute the similarity between vectors. Then we cluster question vectors using a simple single-pass clustering algorithm(Frakes and Yates, 1992). That is, for each question, we compute its similarity with each centroid of existing cluster. If the similarity with nearest cluster is greater than a minimum similarity threshold , we assign this question to that cluster, else a new cluster is created for this question.

Each cluster is defined as a soft pattern. Unlike strict patterns, a question can match a soft pattern to some extent. In this paper, the degree of matching is defined as the cosine similarity between question and centroid of cluster. Soft patterns are flexible and could alleviate the sparseness of content words. Also, soft patterns can be pre-filtered by information gain described in 4.2 if necessary.

4.4 Implementation

Currently, we model patterns into MLN as follows. The main query predicate is Type(q,t), which is true iff question q has type t. For strict patterns, the evidence predicate MatchP(q,p) is true iff question q is matched by strict pattern p. The confidence of MatchP(q,p) is 1 for each pair of (q,p). For soft patterns, the evidence predicate MatchC(q,c) is true iff the similarity of question q and the cluster c is greater than a minimum similarity requirement . If MatchC(q,c) is false, its confidence is 1, else is the similarity between q and c.

We represent the relationship between patterns and types by a group of formulas below.

MatchP(q,+p) Type(q,+t) Type(q,t')

The "+p, +t" notation signifies that the MLN contains an instance of this formula for each (*pattern*, *type*) pair. For the sake of efficacy, for each pattern-type pair (p,t), if the proportion of type t in questions matching p is less than a minimum requirement , we remove corresponding formula from MLN.

Similarly, we incorporate soft patterns by

MatchC(q,+c) Type(q,+t) Type(q,t')

Our weight learner use -regularization (Huynh and Mooney, 2008) to select formulas and prevent overfitting. A good property of -regularization is its tendency to force parameters to exact zero by strongly penalizing small terms (Lee et al., 2006). After training, we can simply remove the formulas with zero weights.

Formally, to learn weight for each formula, we iteratively solve -norm regularized optimization problem:

where is -norm and parameter controls the penalization of non-zero weights. We implement the Orthant-Wise Limited-memory Quasi-Newton algorithm(Andrew and Gao, 2007) to solve this optimization.

Since we do not model relations among questions, the derived markov network can be broken up into separated subgraphs by questions and the gradient of CLL(Eq. 1) can be computed locally on each subgraph as

(2)

in which and are the evidence and query atoms involving question . Eq. 2 can be computed fast without approximation.

We initialize formula weights to the same positive value . Iteration started from uniform prior can always converge to a better local maximum than gaussian prior in our task.

5 Experiments

5.1 Data Preparation

To the best of our knowledge, there is not general QA system(the system which can potentially answer all kinds of questions utilizing data from heterogeneous sources) released at present. Alteratively, we test our methodology on cQA data based on observation that questions on cQA services are of various length, domain independent and wrote informally(even with grammar mistakes). General QA systems will meet these challenges as well.

In our experiments, both training and test data are from Chinese cQA services Baidu Zhidao and Sina iAsk. To build training set, we randomly select 5800 frequent-asked questions from Baidu Zhidao. A question is frequent-asked if it is lexically similar to at least five other questions. Then we ask 10 native-speakers to annotate these questions according to question title and question description. If an annotator cannot judge type from question title, he can view the question description. If type can be judged from the description, the question title will be replaced by a sentence selected from it. If not, this question will be labeled as **Other**.

Each question is annotated by two people. If a question is labeled different types, another annotator will judge it and make final decision. If this annotator cannot judge the type, this question will also be labeled as **Other**. As a result, disagreements show up on eighteen percents of questions. After the third annotator's judgment, the distribution of each type is shown in Fig. 1.

To examine the generalization capabilities, the test data is composed of 700 questions randomly selected from Baidu Zhidao and 700 questions from Sina iAsk. The annotation process on test data is as same as the one on training data.

5.2 Methods Compared and Results

We compare four methods listed as follows.

SVM with bi-grams. We extract bi-grams from questions on training data as features. After filtering the ones appearing only once, we collect 5700 bi-grams. LIBSVM(Chang and Lin, 2001)is used as the multi-class SVM classifier. All parameters are adjusted to maximize the accuracy on test data. We denote this method as "*SB*";

MLN with bi-grams. To compare MLN and SVM, we treat bi-grams as strict patterns. If a question contain a bi-gram, it matches the corresponding pattern. We set , and . As a result, 5700 bi-grams are represented by 10485 formulas. We denote this method as "*MB*";

MLN with strict patterns and bi-grams. We ask two native-speakers to write strict patterns for each type. The pattern writers can view training data for reference and write any Java-style regular expressions. Then we carefully choose 50 most reliable patterns. To overcome the low coverage, We also use the method described in Sec. 4.2 to automatically extract strict patterns from training set. We first select top 3000 patterns by information gain, merge these patterns with hand-crafted ones and combine similar patterns. Then we represent these patterns by formulas and learn the weight of each formula by MLN. After removing the formula with low weights, we finally retain 2462 patterns represented by 3879 formulas. To incorporate content information, we extract bi-grams from questions with function words removed and remove the ones with frequency lower than two. With bi-grams added, we get 8173 formulas in total. All parameters here are the same as in "*MB*". We denote this method as "*MSB*";

MLN with strict patterns and soft patterns. To incorporate content information, We cluster questions on training data with similarity threshold

and get 2588 clusters(soft patterns) which are represented by 3491 formulas. We these soft patterns with strict patterns extracted in "*MSB*", which add up to 7370 formulas. We set and the other parameters as same as in "*MB*". We denote this method as "*MSS*";

We separate test set into easy set and difficult set. A question is classified into easy set iff it contains function-words. As a result, the easy set contains 1253 questions. We measure the accuracy of these four methods on easy data and the whole test data. The results are shown in Tab 4. From the results we can see that all methods perform better on easy questions and MLN outperforms SVM using same bi-gram features. Although *MSS* is inferior to *MSB* on

	F. num	Easy data	All data
SB	NA	0.724	0.685
MB	10485	0.722	0.692
MSB	8173	**0.754**	0.714
MSS	7370	0.752	**0.717**

Table 4: Experimental results on Chinese cQA data

	F	L	S	R	D	N
Prec.	0.63	0.65	0.83	0.76	0.69	0.55
Recall	0.55	0.74	0.86	0.76	0.44	0.58
	0.59	0.69	0.84	0.76	0.54	0.56

Table 5: Precision, recall and F-score on each type

easy questions, it shows better overall performance and uses less formulas.

We further investigate the performance on each type. The precision, recall and -score of each type by method *MSS* are shown in Tab. 5. From the results we can see that the performance on **Solution** and **Reason** are significantly better than the others. It is because the strict patterns for this two types are simple and effective. A handful of patterns could cover a wide range of questions with high precision. It is difficult to distinguish **Fact** from **List** because strict patterns for these two types are partly overlap each other. Sometimes we need content information to determine whether the answer is unique. Since **List** appears more frequently than **Fact** on training set, MLN tend to misclassify **Fact** to **List** which lead to low recall of the former and low precision of the latter. The recall of **Definition** is very low because many definition questions on test set are short and only consists of content words(e.g. a noun phrase). This shortage could be remedied by building strict patterns on POStagging sequence.

fraction lines, college entrance exam
分数，数线，高考，考分，录取，...
Fact: 56.4% List: 33.3% Solu.: 5.5%
lose weight, summer, fast
减肥，夏天，快速，速减，方法，...
Reas.: 53.8% Solu.: 42.3% List: 3.8%
TV series, interesting, recent
电视，视剧，好看，最近，最新，...
List: 84.0% Fact: 8.0% Navi.: 2.0%
converter, format, 3gp
转换，换器，3gp，mp4，格式，...
Navi.: 75% List: 18.8% Solu.: 6.2%

Table 6: Selected soft patterns on training data

5.3 Case Study on Soft Patterns

To give an intuitive illustration of soft patterns, we show some of them clustered on training data in Tab. 6. For each soft pattern, we list five most frequent bi-grams and its distribution on each type(only top 3 frequent types are listed).

From the results we can see that soft patterns are consistent with our ordinary intuitions. For example, if user ask a questions about “TV series”, he is likely to ask for recommendation of recent TV series and the question have a great chance to be **List**. If user ask questions about “lose weight”, he probably ask something like “How can I lose weight fast?” or “Why my diet does not work?” . Thus the type is likely to be **Solution** or **Reason**.

6 Conclusion and Future Work

We have proposed a new question taxonomy tailored to general QA on heterogeneous sources. This taxonomy provide indispensable information for question distribution and answer summarization. We build strict patterns and soft patterns to represent the information in function words and content words. Also, fuzzy discriminative weight learning is proposed for unifying strict and soft patterns into Markov logic network.

Currently, we have not done anything fancy on the structure of MLN. We just showed that under uniform prior and L1 regularization, the performance of MLN is comparable to SVM. To give full play to the advantages of MLN, future work will focus on fast structure learning. Also, since questions on online communities are classified into categories by topic, we plan to perform joint question type inference on function-based taxonomy as well as topic-based taxonomy by Markov logic. The model will not only capture the relation between patterns and types but also the relation between types in different taxonomy.

Acknowledgment

This work was supported mainly by Canada’s IDRC Research Chair in Information Technology program, Project Number: 104519-006. It is also supported by the Chinese Natural Science Foundation grant No. 60973104.

References

G. Andrew and J. Gao. 2008. *Scalable training of L1-regularized log-linear models*. In Proc. of ICML 2007, pp. 33-40.

A. Broder. 2002. *A taxonomy of Web search*. SIGIR Forum, 36(2), 2002.

C.C. Chang and C.J. Lin. 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

W.B. Frakes and R. Baeza-Yates, editors. 1992. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, 1992.

A.C. Graesser, N.K. Person and J.D. Huber. 1992. *Mechanisms that generate questions*. Questions and Information Systems, pp. 167-187), Hillsdale, N.J.: Erlbaum.

T.N. Huynh and R.J. Mooney. 2008. *Discriminative Structure and Parameter Learning for Markov Logic Networks*. In Proc. of ICML 2008, pp. 416-423.

B.J. Jansen and D. Booth. 2010. *Classifying web queries by topic and user intent*. In Proc. of the 28th international conference on human factors in computing systems, pp. 4285-4290.

J. Jeon, W.B. Croft and J.H. Lee. 2005. *Finding similar questions in large question and answer archives*. In Proc. of ACM CIKM 2005,pp. 76-83.

S. Lee, V. Ganapathi and D. Koller. 2005. *Efficient structure learning of Markov networks using -regularization.*. Advances in Neural Information Processing Systems 18.

U. Lee, Z. Liu and J. Cho. 2005. *Automatic identification of user goals in Web search*. In Proc. of WWW 2005.

W. Lehnert. 1977. *Human and computational question answering*. Cognitive Science, vol. 1, 1977, pp. 47-63.

X. Li and D. Roth. 2002. *Learning question classifiers*. In Proc. of COLING 2002, pp. 556-562.

X. Li and D. Roth. 2004. *Learning question classifiers: the role of semantic information*. Natural Language Engineering.

Y. Liu, S. Li, Y. Cao, C.Y. Lin, D. Han and Y. Yu. 2008. *Understanding and summarizing answers in community-based question answering services*. In Proc. of COLING 2008.

A. Moschitti, S. Quarteroni, R. Basili and S. Manandhar. 2007. *Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classification*. In Proc. of ACL 2007.

M.L. Nguyen, T.T. Nguyen and A. Shimazu. 2007. *Subtree Mining for Question Classification Problem*. In Proc. of IJCAI 2007.

H. Poon and P. Domingos. 2009. *Unsupervised semantic parsing*. In Proc. of EMNLP 2009, pp. 1-10

D.E. Rose and D. Levinson. 2004. *Understanding user goals in web search*. In Proc. of WWW 2004.

M. Richardson and P. Domingos. 2006. *Markov logic networks*. Machine Learning 62:107-136.

P. Singla and P. Domingos. 2005. *Discriminative Training of Markov Logic Networks*. In Proc. of AAAI 2005.

S. Ye, T.S. Chua and J. Lu. 2009. *Summarizing Definition from Wikipedia*. In Proc. of ACL 2009.

D. Zhang and W.S. Lee. 2003. *Question classification using support vector machines*. In Proc. of ACM SIGIR 2003, pp. 26-32.

J. Zhu , Z. Nie, X. Liu, B. Zhang and J.R. Wen. 2009. *StatSnowball: a Statistical Approach to Extracting Entity Relationships*. In Proc. of WWW 2009, pp. 101-110