A Principled Decomposition of Pointwise Mutual Information for Intention Template Discovery

Denghao Ma, Kevin Chen-Chuan Chang, Yueguo Chen, Cheng Chen, Chuanfei Xu

Abstract—As dialog systems emerge, *question answering* has become common for users to interact with computers. Towards question understanding, we advocate learning how users ask. As people ask different expressions for the same intention, such intention paraphrase templates, while challenging our understanding, also present a novel opportunity. To exploit this opportunity, this paper proposes the *intention paraphrase template discovery* task: Given some seed questions or templates of an intention, discover new paraphrase templates that describe the intention and are diverse to the seeds enough. As the first exploration of the task, we identify the new quality requirements, i.e., relevance, divergence and popularity, and identify the new challenges, i.e., the paradox of divergent yet relevant paraphrases, and the conflict of popular yet relevant paraphrases. To untangle the paradox of divergent yet relevant paraphrases, in which the traditional bag of *words* falls short, we develop *utility-centric modeling*, which represents a question/template/answer as a bag of *usages* that users engaged (e.g., up-votes), and uses a *utility-flow* graph to interrelate templates, questions and answers. To balance the conflict of popular yet relevant paraphrases, we propose a new and principled decomposition for the well-known Pointwise Mutual Information (PMI), and then develop a Bayesian framework over the utility-flow graph to estimate PMI. The extensive experiments are performed over three large CQA corpora, and show strong performance advantage over the baselines adopted from paraphrase identification methods. We release $885,000$ paraphrase templates with high qualities discovered by our proposed PMI decomposition model, and the data is available in site https://github.com/Para-Questions/Intention_template_discovery.

Index Terms—Question answering, Template Discovery, Pointwise Mutual Information, Bayesian Inference.

1 Introduction

The advent of natural human-computer interaction has rendered *question answering* (QA) a new norm of information service. Companies have long been providing information in the form of "frequently asked questions". Search engines, beyond keyword queries, are making efforts to support *natural language search*. Most recently, dialog systems, such as *virtual assistants* (e.g., Apple Siri) and *chatbots* (e.g., Microsoft Azure Bot), have fundamentally changed how users interact with computers.

To enable QA, as a key hurdle, we must understand the intention of a question, even though users may "paraphrase" their questions in numerously various ways. E.g., for an intention about i^*: "phone overheating", in our survey, users had asked various expressions with different keywords, as Fig. 1 shows, e.g., "... keep ... cool" (q_1), "... get hot ..." (q_2), "... overheat" (q_3), and many more. We consider those question utterances with a similar intention as intention paraphrase templates. Such templates are numerous; on average, there are 89 variants per question on Quora and 654 on Baidu Knows, as our survey (Section 4) showed. The problem was so pronounced that, in 2017, Quora released a "Quora Question Pairs" to detect duplicate pairs, which attracted more than 3000 teams to compete. Therefore, numerous paraphrase templates are used for describing one intention.

To address this hurdle, we advocate learning patterns/templates from users' asking questions. For asking an intention (e.g., i^*), what different paraphrase templates will users use? As textual representations following linguistic convention, such templates can be effectively discovered from what most users had asked, if we have sufficient data— such "past-to-predict-future" is the essence of data mining. The choice of template expressions depends on applications; In this paper, we focus on "surface" textual patterns, which has been shown powerful in many similar applications (e.g., QA [1], web search [2]). To capture common patterns of similar questions (e.g., q_1 and q_4 in Fig. 1), we can define an intention *template* (t_1 or t_2) as, say, a regular expression of words ("keep", "cool") and numbered placeholders (e.g., in t_2: "#1" for "how to" as defined in L_1 and "#2" for "iphone" or "galaxy" as in L_2). Such templates are effective "question models" to predict what intentions and how lexically users may ask.

While useful, the novel opportunity of intention paraphrase template discovery— to find paraphrase templates for a given intention has not been explored. As Sec. 2 will discuss, in NLP, the paraphrase recognition and question duplicate recognition tasks only focus on recognizing whether an input pair of textual units is a paraphrase/duplicate or not, while ignore the first stage— how to get the input textual pairs. Although the two tasks have been deeply explored, few studies have contributed to the first stage. The question retrieval task is to return the most relevant/similar question for an input query, and ignores the "paraphrase" requirement — textual divergence. So the solutions of question retrieval are not suitable for discovering the input textual pairs for the paraphrase recognition and question duplicate recognition tasks.

- *Denghao Ma, Meituan, Beijing, China. E-mail: madenghao@meituan.com*
- *Kevin Chen-Chuan Chang is with Department of Computer Science, University of Illinois at Urbana-Champaign, Illinois, USA. E-mail:kcchang@illinois.edu*
- *Yueguo Chen is with the School of Information, Renmin University of China, Beijing, China. E-mail:chenyueguo@ruc.edu.cn*
- *Cheng Chen is with the School of Information, Renmin University of China, Beijing, China.*
- *Chuanfei Xu is a researcher in Huawei Tech Co., Ltd.*

Fortunately, the intention paraphrase template discovery can be used for constructing the input pairs for the two recognition tasks. We note that no works to date have discovered templates for an intention or measured criteria beyond relevance (e.g., divergence and popularity). How to define such intention template discovery? What criteria should we care? What data sources to discover from? How to find useful templates despite the classic semantic gaps? This paper seeks to find the answers for the above questions.

Problem. We propose the intention paraphrase template discovery task: given some seed questions or templates S (e.g., q_1, q_2) that share a latent intention (i^*), discover new and diverse templates ($t_1, \cdots, t_6$) that users may use to ask i^*.

Criteria and Metrics. What is a good template t for an intention? To gauge how t is relevant and useful, we propose a suite of three metrics. As a paraphrase, t must express a relevant intention (similar to S), and thus the measure of *relevance*. On the other hand, t should be "useful" for question understanding. Thus we ask if t captures a question difficult to recognize— i.e., lexically distinct from seeds— and thus *divergence*. Further, we ask if t is likely to be used, and thus *popularity*. Section 3 will concretely define these metrics.

Data Sources. Where can we discover intention paraphrase templates that meet the triple metrics? As we aim to model user-asked questions, we take a data-driven approach. By our triple criteria, first, to cover relevant questions for any ad-hoc seeds, a data source should be *large scale*. Second, to contain divergent paraphrases of duplicate intents, it should be *redundant* and *diverse*. Third, to gauge popular paraphrases, it should signal how questions were *used*, i.e., *utility-capturing*. Based on these requirements, as Sec. 4 will present, we propose to use the community question answering (CQA) as data sources, e.g., Quora, Yahoo Answers, Baidu, and Amazon.

Challenges. How to satisfy these complementary and potentially contradictory criteria to find good templates? A template that is relevant may not be useful— divergent and popular— and vice versa. There are two-fold challenges.

• *Paradox of Relevance and Divergence.* We must untangle the apparent paradox of semantic relevance and textual divergence. With the well-known semantic gaps, to recognize relevant yet divergent templates, we cannot rely on textual features. We need to break away from the root of the gaps— the conventional bag-of-words used in virtually all existing question matching techniques [3], [4].

Insight: Utility-Centric Modeling. We propose a novel representation of a question x by its usages that users engaged (e.g., clicks, up-votes), which, as x is associated with answers and matched with templates, its usages can be attributed to them accordingly. This attribution enables a uniform bag-of-usages representation for not only questions but also answers and templates, allowing us to interrelate them in a utility-flow graph, which captures how usages distribute among them. The new modeling breaks relevance away from textual similarity and thus untangles the paradox.

• *Conflict of Relevance and Popularity.* We must balance the potential conflict of relevance and popularity. A template (e.g., t_5 and t_6) may be clearly relevant thanks to being specific, which renders it uncommon and unpopular; conversely, a more popular template (e.g., t_1 or t_2) may risk being irrelevant (e.g., matching "how to keep my body cool").

	Question
q_1	How to keep iPhone cool?
q_2	Why does my phone get hot?
q_3	What to do when phone overheats?
q_4	How to keep Galaxy cool?
q_5	LG K8 is very hot?
q_6	The temperature of my phone rises up to 45 degree Celsius, what to do?
...	… … …

	Template
t_1	*How to keep #1 cool?* L1 = {"iphone", "galaxy"}
t_2	*#1 keep #2 cool?* L1 = {"how to"}, L2 = …
t_3	*Why does #1 get hot?* L1 = …
t_4	*What to do when #1 overheats?* L1 = …
t_5	*LG K8 is very hot?*
t_6	*The temperature of #1 rises up to 45 degree Celsius, what to do?* L1 = …
...	… … …

Fig. 1: Questions/templates for intention:"phone overheating".

Insight: PMI Decomposition. We develop a *principled* approach to gauge the quality of a question x w.r.t. latent intention i^* by the pointwise mutual information (PMI) between x and i^*, i.e., how the usages on x and i^* (in the utility-centric modeling) probabilistically depend on each other. Interestingly, we found a new decomposition of PMI as the product of relevance and popularity. This finding not only provides an information theoretic balance for relevance and popularity but also enables the computation of PMI (which cannot be directly computed because the probability distributions of usages over x and i^* are unknown). Accordingly, we develop a Bayesian inference framework with relevance inference and popularity inference to compute PMI. Note that the finding gives another explanation of PMI— which may be of independent interest.

To address the dual challenges, we present our overall framework in Sec. 5. With CQA databases, we construct a tripartite utility-flow graph where nodes are questions, answers and templates; The edges are from the question-answer pairs and question-template pairs. With seeds as input, we develop a Bayesian inference framework to infer relevance and popularity of templates by a principled PMI approach, and generate a ranked list of templates as output.

We conclude our contributions as follows:

• We propose the intention paraphrase template discovery task by proposing new criteria (relevance, popularity and divergence) and new challenges (paradox of relevance and divergence, conflict of relevance and popularity).

• To address the paradox challenge, we propose the utility-centric modeling with the bag of usages and the utility-flow graph interrelating questions, answers and templates.

• To address the conflict challenge, we propose a new decomposition of PMI. i.e., the product of relevance and popularity, and develop a Bayesian inference framework to infer PMI.

• We release $885,000$ paraphrase templates with high qualities discovered by the PMI decomposition model.

• We conduct extensive experiments on Amazon, Yahoo Answer and Baidu Knows test sets. The results show that our model significantly outperforms both baselines and the improved baselines.

2 Related Work

Problems. Our task, i.e., intention paraphrase template discovery, takes an intention (a set of questions or templates) as input, and outputs a ranked list of *templates* with high qualities (relevance, popularity and divergence). The closer related work is paraphrasing and question answering.

• *Paraphrasing.* Research on paraphrasing typically aims at solving three related problems: 1) recognition [5], [6], to rec-

ognize whether an input pair of textual units is a paraphrase or not; 2) generation [7], [8], [9], to generate new paraphrases given an input text; 3) extraction [10], [11], to globally extract paraphrases from a corpus without any input examples.

• *Question Answering.* Question answering also has three related tasks: 1) question search [12], [13], [14], to return the most relevant question of an input query; 2) Duplicate recognition [15], to identify whether a sentence pair is duplicate or not; 3) Question clustering [16], [17], to organize questions into some clusters without any input example.

The intention paraphrase template discovery is different from the paraphrasing tasks and question answering tasks. This is because 1) the paraphrase recognition and question duplicate recognition highlight classifying a given pair, while discovery highlights retrieving and ranking. The discovery can be applied to construct the candidate pairs for recognition; 2) Paraphrase generation highlights generating new paraphrases, while discovery highlights retrieving existing paraphrases from question corpus; 3) Paraphrase extraction and question clustering highlight globally mining which does not require the input seeds, while discovery highlights intention-specific retrieval that requires the seeds; 4) Question search highlights returning the most relevant question, while discovery highlights retrieving all relevant questions. Besides, these existing paraphrasing tasks aim to find relevant and divergent results, even though almost all studies only focus on relevance and unfortunately ignore popularity and divergence. As well the tasks of question answering only focus on the relevance metric. However, the intention paraphrase template discovery highlights multiple qualities, i.e., relevance, popularity and divergence.

Techniques. Since paraphrase recognition is closer to our task than other tasks, we only introduce the techniques of paraphrase recognition. To well recognize the paraphrases, many paraphrase features are designed [5], [7], [10], [18], [19], such as text similarity features, machine translation features, co-click similarity features and entity similarity features. With the development of the neural networks, many end-end deep learning models are proposed [20], [21], [22], [23]. The main deep learning models can be classified into 1) LSTM-based models [20], [21]; 2) CNN-based models [24], [25]; 3) Transformer-based models [22], [26]; 4) the hybrid models [23], [27], [28]. Almost all of these solutions only focus on the relevance estimation. Fortunately, there is one study [5] using the bag of words model to estimate both relevance and divergence. Unfortunately, the solution in [5] cannot address the paradox of relevance and divergence due to the usage of bag-of-words model. Besides, these solutions lack the process of finding candidate pairs, which is also desired in our intention paraphrase template discovery problem. We therefore design an intention-driven ranking framework (see Section 6) so that these solutions can be adapted as our baselines.

Our proposed PMI decomposition model is unsupervised and different from the above methods. Firstly, to address the paradox of relevance and divergence, we propose a new representation model, i.e., bag of usages and a utility-flow graph interrelating the questions, answers and templates. Secondly, to address the conflict of relevance and popularity, we propose a new decomposition of PMI and develop a Bayesian inference framework with popularity and relevance inferences over the utility-flow graph to infer PMI.

Metrics. We are the first to estimate the paraphrase quality from relevance, popularity, divergence and their combination. Relevance and divergence are essential requirements of paraphrases, and popularity captures the usefulness. But existing paraphrasing tasks estimate the quality by only relevance.

3 Problem and Criteria

Data Source. We use a CQA as our data source, as Sec. 4 will propose. A CQA database $D = (Q, A, U)$ contains 1) a set Q of *questions* $\{q_1, \ldots, q_{|Q|}\}$, each of which is a text indicating an *intention*, 2) a set A of *answers* $\{a_1, \ldots, a_{|A|}\}$, each of which replies to one or more questions, and 3) a set U of *usages* $\{u_1, \ldots, u_{|U|}\}$, each of which is a user engagement such as a *click* to view, *vote* to approve, or *follow* to subscribe.

Questions, answers and usages are associated with one another in a CQA. First, each question can be replied by multiple answers, and an answer can also reply multiple questions since similar answers (for different questions) can be merged. In our experiments, we use the solution [29] to merge similar questions and answers during the phase of data processing. We use $a \in \mathcal{A}(q)$ or $q \in \mathcal{Q}_A(a)$ if a replies q. Second, we use $u \in \mathcal{U}(q, a)$ to record a usage u (e.g., a vote) performed on the (q, a) pair (e.g., u voted a as "good answer" for q). For convenience, we write $\mathcal{U}(q)$ for all usages on question q, i.e., $\mathcal{U}(q) = \Sigma_{a \in A}\, \mathcal{U}(q, a)$ and similarly $\mathcal{U}(a)$ for those on answer a, i.e., $\mathcal{U}(a) = \Sigma_{q \in Q}\, \mathcal{U}(q, a)$.

Input. We take some *seeds* (seed questions) S that indicate a latent intention i^*. We will discuss with one seed for simplicity, but the framework generally takes a set of seeds $S = \{q_1, \ldots, q_{|S|}\}$ for their shared intent. In our example (Fig. 1), suppose $S = \{q_1\}$ (for i^*= "phone overheating").

Output. As output, we wish to discover intention templates that share the same intention with S, ranked by some criteria (which we will define). We choose to output templates instead of specific questions since many questions in a CQA are textually similar (or nearly identical), which can be compactly captured by a common template without loss of information.

As a template is to describe a class of questions, we can use any reasonable features or rules, e.g., dependency parses, part of speech (POS) tags, or simply keywords. Our framework is agnostic with regard to template representation. As Sec. 1 mentioned and Fig. 1 shows, we use simple regular expressions to represent templates.

Definition 1 (Template). A *template* $t = (W, L = \{L_1, \ldots, L_{|L|}\})$, where 1) W is a sequence *pattern* $\langle w_1 \cdots w_{|W|} \rangle$, where w_i is a word or a symbol $\#j$ as the j^{th} placeholder and 2) L_j is the j^{th} lexicon, a set of *terms* that can be substituted for $\#j$. A question q = "$x_1 \cdots x_n$" *instantiates* t or t *abstracts* q if, $\forall i$, $w_i = x_i$ when w_i is a word and $x_i \in L_j$ when $w_i = \#j$.

We generate templates (e.g., t_2) from similar question by abstracting alternative terms into a placeholder (e.g., #1 and #2) and recording them in lexicons (L_1 and L_2). We use the BERT+LSTM+CRF model to identify the alternative terms from questions. With such abstraction, a question q can match multiple templates, which we denote $\mathcal{T}(q)$; e.g., $\mathcal{T}(q_1) = \{t_1, t_2\}$. Conversely, a template t can instantiate

many questions, which we denote $\mathcal{Q}_T(t)$, where $\mathcal{Q}_T = \mathcal{T}^{-1}$; e.g., $\mathcal{Q}_T(t_1) = \{q_1, q_4\}$.

Criteria. Our objective is to find intention templates t, which can be instantiated into paraphrase questions of S. As Sec. 1 motivated, good paraphrase questions and thus good templates should satisfy the triple criteria:

• *Divergence.* A useful paraphrase question q should be lexically different from S, since it helps us to predict difficult variations. E.g., w.r.t. q_1, q_3 is more divergent and thus more useful than q_4. In a "noisy" CQA, there often exist a large number of very similar questions (e.g., q_1 and q_4), and we wish to find distinct ones. We denote the divergence of q (w.r.t. seeds S) by $\text{div}(q)$, which can be any reasonable text similarity measures, e.g., cosine similarity or Jaccard coefficient. In our experiment, we use normalized *Kullback-Leibler (KL) divergence*, i.e., $\text{div}(q) \equiv \frac{KL(q\|S)}{\max_{q \in Q} KL(q\|S)}$.

• *Relevance.* A paraphrase question q should be relevant to S, which expresses a similar intention $i^*$and can be replied by similar answers. As users can ask about any intents and natural language is not precise, their questions will exhibit different degrees of similarity that is subjective to interpret. Thus, the semantic relevance of q w.r.t. S is *probabilistic*, which relates q to i^*: *How likely will a user have intention i^*, if she asks q?* We denote this relevance by $\text{rel}(q)$ and define it as the probability from query to intent, i.e., $\text{rel}(q) \equiv p(i^*|q)$. We will concretely model (Sec. 5) and infer it algorithmically.

• *Popularity.* As paraphrases are numerous, we wish to discover those popular questions, which will more likely be asked. Indeed, our CQA survey (Section 4) found that, while a question can have hundreds of synonyms, some are used much more than others. We shall thus measure the specific popularity of q w.r.t. S: *How likely will a user ask q, if she has intention i^*?* We denote this popularity by $\text{pop}(q)$ and define it as the probability from intention to query– which is the *inverse* of relevance– i.e., $\text{pop}(q) \equiv p(q|i^*)$.

As output, each template t can instantiate some questions q ($q \in \mathcal{Q}_T(t)$), and thus its measures are their aggregates.

$$\text{div}(t) \equiv \text{E}[\text{div}(q)] = \Sigma_{q \in \mathcal{Q}_T(t)}\, p(q) \cdot \text{div}(q) \quad (1)$$

$$\text{rel}(t) \equiv p(i^*|q \in \mathcal{Q}_T(t)), \quad (2)$$

$$\text{pop}(t) \equiv p(q \in \mathcal{Q}_T(t)|i^*) \quad (3)$$

where $p(q)$ is estimated with $1/|\,\mathcal{Q}_T(t)|$.

4 Data Source

While learning how users may ask a question for an intention is important, it remains open *where* we can learn– or what *data sources* can fulfill this promise. We have identified several requirements that an ideal data source should have. *CQA* questions are natural language sentences, which is common sense. In this section, we conduct a survey of four CQA databases to verify whether CQA can satisfy the requirements based on 4 large scale CQA databases: Baidu Knows, Amazon, Yahoo Answers and Quora.

• *Large Scale.* We take some popular entity types (e.g., "phone") as queries and use the number of returned questions to evaluate the scale of CQA. Results in Tab. 1 show that Baidu Knows, Yahoo Answers and Quora are in large scale.

	Baidu	Amazon	Yahoo	Quora
Scale: "Phone"	93,975,800	34,639	3,418,624	27,300,000
Scale: "Job"	61,420,832	86	4,919,309	30,600,000
Scale: "Car"	27,906,274	19,674	3,877,098	25,300,000
Scale: "Child"	49,171,676	274	4,374,292	26,700,000
Scale: "Computer"	17,643,646	8,068	4,401,798	24,000,000
Redundancy	654	387	119	89
Utility	68	29	22	36

TABLE 1: CQA characteristics.

Amazon is relatively small because it is a commercial question answering website about products.

• *Redundant.* The redundancy refers to that CQA corpus has a lot of equivalent questions. For each CQA corpus, we randomly pick up 20 CQA questions as queries, and report the average number of synonymous questions of the queries. The results in Tab. 1 illustrate that CQA database contains a lot of redundant questions.

• *Diverse.* To show the divergence of synonymous questions in CQA database, we report the text difference (KL-divergence) between synonymous questions and each query in Fig. 2. The synonymous questions are ordered by their divergence. It is obvious that most synonymous questions have large divergence to the input question. Note that questions of the Yahoo data set have less divergence than those of the others.

• *Utility-capturing.* CQA forums allow users to follow, to vote and to click questions and answers. To investigate the popularity of synonymous questions, we enumerate votes (as utilities) of each synonymous question. The average number of utilities is reported in Tab. 1. Results in Fig. 3 show that synonymous questions have different popularity.

This survey verifies that CQA databases have the large scale, redundant, diverse, and utility-capturing characteristics. So they can be applied as our data to discover paraphrase templates of an intention. We don't use query logs because most of questions in query logs are keywords which cannot satisfy the characteristic of *natural language* questions.

5 Solution

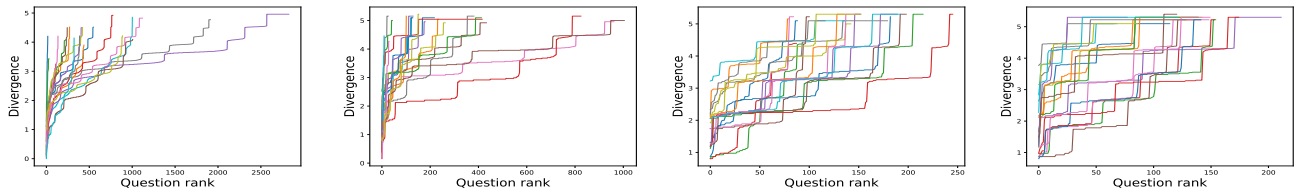
In this section, we present our solution with the utility-centric modeling and PMI decomposition model. The framework of our solution is shown in Fig. 4.

5.1 Utility-Centric Modeling

To address the paradox challenge of relevance and divergence, we observe that a CQA provides rich utility information that can be exploited to define and infer semantics of questions and answers– which frees us from relying on textual features, the root of the paradox. Thus, we will develop a model that models a question q (and other components in a CQA) as a bag-of-usages (BOU) $\mathcal{U}(q)$, instead of the traditional bag-of-words (BOW) which suffers the paradox.

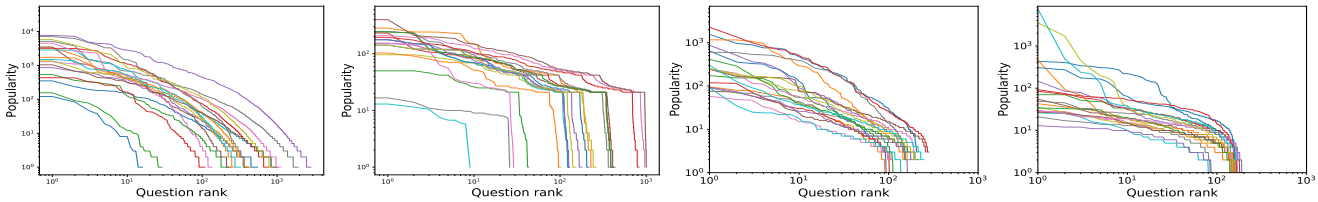
BOU Model. We model a CQA by utility— its usages by users. An intention i, question q, or answer a is simply a bag of usages, $\mathcal{U}(i)$, $\mathcal{U}(q)$, and $\mathcal{U}(a)$ respectively. Each usage u— as performed by some users at some time— is naturally an atomic unit of action with a unique intention i, i.e., $\exists! i,\, u \in \mathcal{U}(i)$. As u is performed on a (q, a) pair, it also appears in exactly one q and a, i.e., $\exists! q,\, u \in \mathcal{U}(q)$ and $\exists! a,\, u \in \mathcal{U}(a)$.

To model the template t, we represent it also as a BOU. Since t abstracts multiple questions, its usages are



(a) Baidu Knows (b) Amazon (c) Quora (d) Yahoo Answers

Fig. 2: Divergence of synonymous questions.



(a) Baidu Knows (b) Amazon (c) Quora (d) Yahoo Answers

Fig. 3: Popularity of synonymous questions.

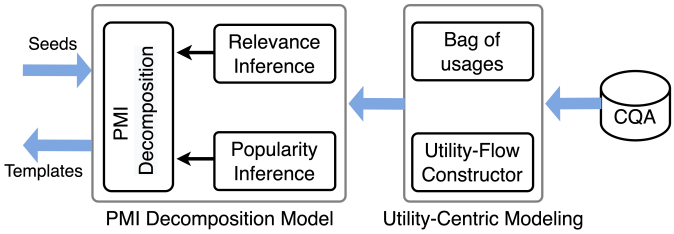


Fig. 4: Overall solution framework.

the weighted aggregates of these questions, i.e., $\mathcal{U}(t) \equiv \Sigma_{q \in \mathcal{Q}_T(t)} w(t,q)\,\mathcal{U}(q)$ where $w(t,q)$ is the weight of $\mathcal{U}(q)$. When a user uses a question q (e.g., q_3 in Fig. 1) by usage u, she chooses q because of its certain textual characteristics (e.g., keywords "do", "overheats") which t may capture (t_4 in this case). So we apply the textual similarity to estimate $w(t,q) = \frac{ts(t,q)}{\sum_{t_1 \in \mathcal{T}(q)} ts(t_1,q)}$, $ts(t,q)$ is the textual similarity of t and q, estimated by the language model [30] in experiments.

As CQA is a corpus of bag-of-usages, to complete our utility-centric modeling, we concretize relevance and popularity (with $\text{rel}(q) \equiv p(i^*|q)$ and $\text{pop}(q) \equiv p(q|i^*)$ defined previously). For $\text{rel}(q)$, *how likely will a usage u have an intention i^*, if it is performed on q?* I.e., how likely will $u \in \mathcal{U}(i^*)$, if $u \in \mathcal{U}(q)$? To simplify notation, let's denote the set of "true" usages as $\mathcal{U}(i^*)$. Since any question q, answer a, or template t are all modeled uniformly as a BOU and let x be such q, a, or t, we can rewrite the relevance of any x as:

$$\text{rel}(x) \equiv p(u \in \mathcal{U}(i^*) | u \in \mathcal{U}(x)). \tag{4}$$

For $\text{pop}(q)$, *how likely will a usage u be performed on q, if it has an intention i^*?* I.e., how likely will $u \in \mathcal{U}(q)$, if $u \in \mathcal{U}(i^*)$? With the uniform BOU representation, we also rewrite $\text{pop}(x)$, for x being q, a, or t, as:

$$\text{pop}(x) \equiv p(u \in \mathcal{U}(x) | u \in \mathcal{U}(i^*)). \tag{5}$$

We note that divergence remains in BOW (defined by KL divergence) and cannot be redefined in BOU, since textual divergence should be measured by BOW features.

To infer $\text{rel}(x)$ and $\text{pop}(x)$, as both are interrelated conditional probabilities (Eq. 4 and 5), we need relate them for various x in the CQA, so that they can be infered using their Bayesian dependencies. We first construct a Utility-flow graph for capturing the dependencies among them.

Utility-flow Graph. Our utility-centric model of CQA has turned it into a corpus of usages u distributed over various "visible" elements x: $q \in Q$, $a \in A$, and $t \in T$. (Usages are also associated with intents $i \in I$, which is latent.) The distribution is recorded in CQA, by the function $\mathcal{U}$ (Section 3), which assigns a usage u to a (q,a) pair (in the $Q \times A$ space) and (q,t) pair (in $Q \times T$). Since usages are intent-bearing— i.e., each usage bears a unique intent— their correspondences between different spaces of elements (Q, A, and T) provide robust signals for their semantic dependencies. I.e., each usage on a pair (x,y) reveals the semantic equivalence– as judged by a user– between x and y. Thus, we construct the utility-flow graph over the CQA, based on $\mathcal{U}$, to capture the usage associations between elements.

We define utility-flow graph $= (\mathcal{N}, \mathcal{E}, \mathcal{W})$ as a tripartite graph consisting of partitions Q, A, and T, with edges representing the flow of utilities between elements. $\mathcal{N} = Q \cup A \cup T$ is a set of nodes in three partitions. $\mathcal{E}$ is a set of edges between partitions Q and A and partitions Q and T, with $\mathcal{W}$ as the weights of those edges. An edge $e(x,y)$ between nodes x and y represents the sharing of usages that are recorded for the (x,y) pair, with a weight $w(x,y) = |\mathcal{U}(x,y)|$, i.e., the number of usages for the pair (x,y).

5.2 PMI Decomposition

To address the conflict challenge of relevance and popularity, our utility-centric modeling enables an information theoretic approach which, as we will see, coherently combines relevance and popularity— based on the classic measure of (point-wise) mutual information.

While a CQA records how each usage u is associated with some q, a, and t, it does not know how u associates with which intent– which is latent. The associations of (q, a) and (q, t) allow us to capture the relations between q, a and q, t which enables our Bayesian inference framework. However, without the latent information of how $u \in \mathcal{U}(i^*)$, though we have how $u \in \mathcal{U}(x)$ ($\forall x$ as q, a, or t), we cannot empirically (taking a CQA as observation) estimate $\text{rel}(x)$ and $\text{pop}(x)$ which need to infer probabilistically.

We observe that a CQA represents a sample of how usages u distribute over intents, questions, templates, and answers. As users visit the CQA, they would use different parts as wish, creating a distribution for each space. That is, for a usage u, for which $i \in I$ will u be performed? And, over which $q \in Q$, $t \in T$, and $a \in A$? The behaviors are not deterministic– they are the probabilistic *distributions* of $p(u \in \mathcal{U}(i))$, $p(u \in \mathcal{U}(q))$, $p(u \in \mathcal{U}(t))$, and $p(u \in \mathcal{U}(a))$, over the *sample spaces* of I, Q, T, and A respectively.

We are interested in how the usages performed at some x ($\forall x$ as q, a, or t) will coincide with the latent intention i^*. Intuitively, if x (say q_2) always coincides with i^*, it means when a usage is on x, it is also on i^*(and thus x is relevant) and vice versa (thus x is popular). More formally, inspecting Eq. 4 and 5, we observe they both measure the coincidence of $u \in \mathcal{U}(x)$ and $u \in \mathcal{U}(i^*)$— but in *opposite* directions.

This observation clearly motivates us to measure the quality of the *coincidence* between outcomes x (as q, a, or t) and y (as i^*). Taking an *information theoretic* approach, we propose to measure this coincidence with *point-wise mutual information* (PMI), which is precisely for this purpose. Specifically, we use the normalized PMI [31], which is $\text{PMI}(x;y) = \ln \frac{p(x,y)}{p(x)\,p(y)}$ normalized by $\frac{1}{p(x,y)}$ for the term inside the logarithm (which is its upper bound when $p(x) = p(y) = 1$):

$$\text{PMI}(x;y) = \ln\left(\frac{p(x,y)}{p(x)\,p(y)} \Big/ \frac{1}{p(x,y)}\right) = \ln p(y|x)\,p(x|y) \quad (6)$$

To measure the coincidence of x and i^* from the usage perspective, we define $\text{PMI}(x;i^*)$ as follows:

$$\begin{aligned}\text{PMI}(x;i^*) &= \text{PMI}(\mathcal{U}(x);\mathcal{U}(i^*)) \\ &= \ln p(\mathcal{U}(i^*)|\,\mathcal{U}(x))\,p(\mathcal{U}(x)|\,\mathcal{U}(i^*)) \\ &= \ln p(u \in \mathcal{U}(i^*)|u \in \mathcal{U}(x))\,p(u \in \mathcal{U}(x)|u \in \mathcal{U}(i^*)) \\ &= \ln \text{rel}(x)\,\text{pop}(x) \propto \text{rel}(x)\,\text{pop}(x) \end{aligned} \quad (7)$$

The derivation of PMI as the product of relevance and popularity has important two-fold implications. First, it addresses our conflict challenge and provides a proper way to combine the conflicting criteria— as their product— with an information theoretic foundation. Second, and conversely, as we will see, it also reveals a way to compute PMI, which is not obvious since we do not know the joint $p(x,y)$ and marginal probabilities $p(x)$, $p(y)$– Eq. 7 helps us to realize PMI as the *decomposition* into relevance and popularity.

Quality Propagation Principle. To guide the propagation of seed qualities (popularity and relevance), we define the quality propagation principle over the utility-flow graph, based on Bayesian inference over edges that connect elements. Since each edge $e(x,y)$ with weight $w(x,y)$ indicates $w(x,y)$ usages are shared by the (x,y) pair, we can model the flow of usages between nodes x and y: What proportion of usages in $\mathcal{U}(x)$ are from those in $\mathcal{U}(y)$, and vice versa? That is, we would like to model the condition probability $p(u \in \mathcal{U}(x)|u \in \mathcal{U}(y))$ and the inverse $p(u \in \mathcal{U}(y)|u \in \mathcal{U}(x))$: With weights $w(x,y)$ as observations, we can model them with maximum likelihood estimation, as follows. Such usage flowing occurs between ($\mathcal{X}$= Q, $\mathcal{Y}$= A) and ($\mathcal{X}$= Q, $\mathcal{Y}$= T).

$$p(u \in \mathcal{U}(x)|u \in \mathcal{U}(y)) = \frac{w(x,y)}{\sum_{x_i \in \mathcal{X}} w(x_i,y)} \quad (8)$$

$$p(u \in \mathcal{U}(y)|u \in \mathcal{U}(x)) = \frac{w(x,y)}{\sum_{y_j \in \mathcal{Y}} w(x,y_j)} \quad (9)$$

The inference will estimate the probabilistic popularity $\text{pop}(x)$ and relevance $\text{rel}(x)$ of each node x. Over the utility-flow graph, because each edge captures some quality flows (as Eq. 8 and 9 formulated), we can "propagate" the estimation of $\text{pop}(x)$ through such dependency with simple Bayesian rules. Since the seeds capture our initial knowledge (i.e., for an initial seed $s \in S$, $\text{rel}(s) = 1$ and $\text{pop}(s) = \frac{|\mathcal{U}(s)|}{\sum_{s' \in S}|\mathcal{U}(s')|}$), intuitively, this inference effectively propagates the seed knowledge throughout the utility-flow graph— like a random walk.

• *Popularity Inference.* While the inference is derived via Bayes theorem, intuitively, it is propagating the initial knowledge (of popularity estimates) from seeds to other nodes x, eventually reaching questions q and templates t. Since the seeds are questions, we first infer the popularity of answer nodes and template nodes. According to Eq. 5, the probabilistic popularity of a template t is estimated as:

$$\begin{aligned}\text{pop}(t) &\equiv^1 p(u \in \mathcal{U}(t)|u \in \mathcal{U}(i^*)) \\ &=^2 \sum_{q \in Q} p(u \in \mathcal{U}(t), u \in \mathcal{U}(q)|u \in \mathcal{U}(i^*)) \\ &=^3 \sum_{q \in \mathcal{Q}_T(t)} p(u \in \mathcal{U}(t), u \in \mathcal{U}(q)|u \in \mathcal{U}(i^*)) \\ &=^4 \sum_{q \in \mathcal{Q}_T(t)} p(u \in \mathcal{U}(t)|u \in \mathcal{U}(q), u \in \mathcal{U}(i^*)) \\ &\qquad \cdot p(u \in \mathcal{U}(q)|u \in \mathcal{U}(i^*)) \\ &=^5 \sum_{q \in \mathcal{Q}_T(t)} p(u \in \mathcal{U}(t)|u \in \mathcal{U}(q)) \cdot \text{pop}(q) \end{aligned} \quad (10)$$

Since the popularity of template nodes on the utility-flow graph are from question nodes, we bring in questions, and expand the popularity to a joint distribution of every question $q \in Q$ in step 2. Step 3 restricts $q \in Q$ to only those questions connected with t, so that $q \in \mathcal{Q}_T(t)$. Step 4 rewrites $p(u \in \mathcal{U}(t), u \in \mathcal{U}(q)|u \in \mathcal{U}(i^*))$ based on Bayes theorem, whose first component can be further reduced as $p(u \in \mathcal{U}(t)|u \in \mathcal{U}(q))$ in step 5, since $u \in \mathcal{U}(t)$ is conditionally independent of $u \in \mathcal{U}(i^*)$ given that $u \in \mathcal{U}(q)$. The probability $p(u \in \mathcal{U}(t)|u \in \mathcal{U}(q))$ is estimated by Eq. 8 or 9. Similarly, the popularity of an answer a can be derived as:

$$\text{pop}(a) = \sum_{q \in \mathcal{Q}_A(a)} p(u \in \mathcal{U}(a)|u \in \mathcal{U}(q)) \cdot \text{pop}(q) \quad (11)$$

where $p(u \in \mathcal{U}(a)|u \in \mathcal{U}(q))$ is estimated by Eq. 8 or 9 too.

On utility-flow graph, each question node receives popularity from its neighbors: answer and template nodes. Its popularity is thus estimated as a mixture model of both sides:

$$\begin{aligned}\text{pop}(q) &\equiv^1 p(u \in \mathcal{U}(q)|u \in \mathcal{U}(i^*)) \\ &\equiv^2 \alpha \sum_{a \in \mathcal{A}(q)} p(u \in \mathcal{U}(q), u \in \mathcal{U}(a)|u \in \mathcal{U}(i^*)) \\ &\quad + (1-\alpha) \sum_{t \in \mathcal{T}(q)} p(u \in \mathcal{U}(q), u \in \mathcal{U}(t)|u \in \mathcal{U}(i^*)) \\ &=^3 \alpha \sum_{a \in \mathcal{A}(q)} p(u \in \mathcal{U}(q)|u \in \mathcal{U}(a)) \cdot \text{pop}(a) \\ &\quad + (1-\alpha) \sum_{t \in \mathcal{T}(q)} p(u \in \mathcal{U}(q)|u \in \mathcal{U}(t)) \cdot \text{pop}(t) \end{aligned} \quad (12)$$

Like the proof of Eq. 10, to bring in neighbors, we expand the popularity to a joint distribution of every $a \in \mathcal{A}(q)$ and $t \in \mathcal{T}(q)$ in Step 2. The parameter α is a weight to tune the two components. Since the inference process of $\sum_{a\in\mathcal{A}(q)} p(u \in \mathcal{U}(q), u \in \mathcal{U}(a)|u \in \mathcal{U}(i^*))$ and $\sum_{t\in\mathcal{T}(q)} p(u \in \mathcal{U}(q), u \in \mathcal{U}(t)|u \in \mathcal{U}(i^*))$ is the same as that of the Eq. 10, we directly give their final inference result in Step 3.

• *Relevance Inference.* Given the seeds with initial relevance, we can propagate the relevance via a candidate-centric propagating strategy where candidates (i.e., q, t and a) determine the propagating quantity. According to Eq. 4, the probabilistic relevance of a template t is estimated as follows:

$$\begin{aligned}\mathrm{rel}(t) &\equiv p(u \in \mathcal{U}(i^*)|u \in \mathcal{U}(t)) \\ &=^2 \sum_{q\in Q} p(u \in \mathcal{U}(i^*), u \in \mathcal{U}(q)|u \in \mathcal{U}(t)) \\ &=^3 \sum_{q\in\mathcal{Q}_T(t)} p(u \in \mathcal{U}(i^*), u \in \mathcal{U}(q)|u \in \mathcal{U}(t)) \\ &=^4 \sum_{q\in\mathcal{Q}_T(t)} p(u \in \mathcal{U}(i^*)|u \in \mathcal{U}(q), u \in \mathcal{U}(t)) \\ &\quad \cdot p(u \in \mathcal{U}(q)|u \in \mathcal{U}(t)) \\ &=^5 \sum_{q\in\mathcal{Q}_T(t)} \mathrm{rel}(q) \cdot p(u \in \mathcal{U}(q)|u \in \mathcal{U}(t))\end{aligned} \tag{13}$$

where the $p(u \in \mathcal{U}(q)|u \in \mathcal{U}(t))$ is estimated by Eq. 8 or 9. Since the derivation of answer relevance is the same as that of template relevance, we give the final inference results of relevance of an answer a as follows:

$$\mathrm{rel}(a) = \sum_{q_i\in\mathcal{Q}_A(a)} \mathrm{rel}(q_i) \cdot \frac{w(a,q_i)}{\sum_{q_j\in\mathcal{Q}_A(a)} w(a,q_j)} \tag{14}$$

We use $r_0(q)$ to denote the initial relevance of q. If q is a seed, $r_0(q) = 1$; otherwise, $r_0(q) = 0$. Besides, q receives relevance from both answer nodes and template nodes, the probabilistic relevance in Eq. 5 is estimated as follows:

$$\begin{aligned}\mathrm{rel}(q)=&\beta_1 r_0(q) + (1-\beta_1)\beta_2 \sum_{a_i\in\mathcal{A}(q)} \mathrm{rel}(a_i) \cdot \frac{w(a_i,q)}{\sum_{a_j\in\mathcal{A}(q)} w(a_j,q)} \\ &+ (1-\beta_1)(1-\beta_2) \sum_{t_i\in\mathcal{T}(q)} \mathrm{rel}(t_i) \cdot \frac{w(t_i,q)}{\sum_{t_j\in\mathcal{T}(q)} w(t_j,q)}\end{aligned} \tag{15}$$

where β_1 and β_2 are to adjust weights of three components.

6 Experiments

6.1 Experimental Setting

Overview of Objectives. We performed extensive experiments to verify the following objectives:

•*Overall Quality.* The overall quality refers to that the combination of relevance, divergence, popularity. Can our proposed PMI decomposition model discover higher quality intention templates than baseline models?

•*Individual Quality.* How do the PMI decomposition model as well as the baselines perform in terms of individual quality (relevance, divergence or popularity)?

•*Improvements of Baselines.* What if the baseline solutions are enhanced by incorporating the popularity inference?

•*Ablation Study.* What is the effectiveness of individual components in the PMI decomposition model?

Denotation	Qualities
P(t)=pop(t)	Popularity
R(t)=rel(t)	Relevance
D(t)=div(t)	Divergence
PD(t)=pop(t) · div(t)	Popularity&Diversity
RD(t)=rel(t) · div(t)	Relevance&Diversity
PR(t)=pop(t) · rel(t)	Popularity&Relevance
PRD(t)=pop(t)·rel(t)·div(t)	Popularity&Relevance&Diversity

TABLE 2: Overview of qualities.

•*Parameter Sensitivity.* How robust is our model with respect to different parameter settings?

Test Sets. Experiments are performed over three large CQA data sources, e.g., Baidu Knows, Yahoo Answers, and Amazon. For each source, we create 100 test cases with 2, 3, 4, 5 or 10 seed questions. To generate ground truths, we create a labeling pool by combining the top-300 results of each model (both baselines and PMI). Three external experts label each template t in the pool with a score of 1 (t is relevant to S) or 0 (t is not relevant to S). The relevance of t (rel(t)) is assigned as the average labeling score. According to the definition of popularity, we estimate the popularity of t as $\mathrm{pop}(t) = \mathrm{rel}(t) \cdot |\mathcal{U}(t)|$. We crawl the utilities (votes) of all questions and record them in CQA database. Based on question utilities, we can get the utilities of a template t: $\mathcal{U}(t) \equiv \Sigma_{q\in\mathcal{Q}_T(t)} \mathcal{U}(q)$. The divergence of t is estimated by Eq. 1, denoted as div(t). Based on the rel(t), pop(t) and div(t), we derive different quality combinations in Tab. 2. Many methods may be used for combining the individual qualities and we only adopt a simple one.

Performance Metrics Since each ground truth template has seven qualities, the discovered results should also have these qualities. According to the work [2], we adopt an accumulated form to measure P, PR, PD and PRD. Taking PRD as an example, PRD of the top-n results is estimated:

$$PRD(n) = \frac{1}{|C|} \sum_{S\in C} \sum_{t\in T_n(S)} \frac{PRD(t)}{\sum_{t_1\in G(S)} PRD(t_1)} \tag{16}$$

where C is a set of test cases, and $G(S)$ is a set of ground truth templates of S. The $T_n(S)$ is the discovered top-n results for the input $S \in C$. $PRD(t)$ is PRD quality of t.

For R, D and RD qualities, we simply adopt the average performance of top-n as a metric. Taking R as an example, the R metric is defined as:

$$R(n) = \frac{1}{|C|} \sum_{S\in C} \frac{\sum_{t\in T_n(S)} R(t)}{n} \tag{17}$$

where $R(t)$ is the relevance of t to the input S. The estimations of $P(t)$, $PR(t)$, $PD(t)$, $PRD(t)$, $R(t)$, $D(t)$ and $RD(t)$ are presented in Tab. 2.

Comparison Models. The intention template discovery task focuses on discovering new intention templates, while both the paraphrase identification and question retrieval tasks focus on classifying a given text pair. So the existing solutions of paraphrase identification and question retrieval can not be directly used for addressing our task. We design a ranking framework for adapting the existing solutions as baselines:

$$score(t,S) = \sum_{s\in S} f(L(s),L(t)) \cdot Rel(W(t),W(s)) \tag{18}$$

where $L(*)$ is the set of entities extracted from template/question $*$. The function $f(L(s),L(t))$ equals 1 if $L(s) \subseteq L(t)$; otherwise it is zero. The $W(*)$ is the term

component of $*$, and $Rel(W(t), W(s))$ is the relevance of $W(t)$ and $W(s)$, measured by using paraphrasing feature functions proposed in [5], [18], [22], [23].

Since our proposed PMI decomposition model is unsupervised, we choose some widely applied supervised and unsupervised solutions as baseline models. The details of the baselines are presented as follows:

• ZZL: The work [5] proposes different paraphrase feature functions over Encarta Logs to recognize the sentence-level paraphrases. The cosine similarity feature and named entity overlapping feature are used for capturing text-level relevance of paraphrases. The unmatched word feature is used for capturing divergence. Features of user clicks and translation similarity are used for capturing the semantic relevance. We ignore the translation similarity feature because it is based on Google search engine and therefore is not frequently available.

• LHZ: The work [18] proposes three general feature functions over query logs to identifies the synonymous templates, i.e., entity distribution similarity, co-click similarity and pseudo-document similarity.

• BERT: BERT [22] is used for estimating the relevance of $Rel(W(t), W(s))$. For the Baidu corpus, the model is assigned with 12-layer, 768-hidden, 12-heads and 110M parameters; For the Yahoo and Amazon corpus, the model is assigned with 24-layer, 1024-hidden, 16-heads, 340M parameters. The pre-trained models are available at site [1].

• MFAE: The work [23] proposes an attention model with multi-fusion asking emphasis (MFAE) to estimate the relevance of two questions. BiLSTM is used for encoding the input questions. The inter-attention and self-attention are used for capturing the inter- and intra-asking emphasis, respectively. Then, the eight-way combinations is used for generating multi-fusion word representation. Finally, the representation is put into a multi-layer perception classifier. The parameter assignment is the same as that in the work [23].

The ZZL and LHZ models need user's click information (such as template-document clicks) of query logs. Without query logs, we take template-answer usages as template-document clicks. To collect template-answer usages, we assume that usages performed on a question-answer pair (q, a) are also performed on a template-answer pair (t, a) satisfying $t \in \mathcal{T}(q)$. To combine multiple features, we exploit the widely used feature fusion way in work [32]:

$$Rel(W(t), W(s)) = \sum_{i}^{m} \theta_i \cdot f_i(W(t), W(s)) \tag{19}$$

where $f_i(W(t), W(s))$ is the i^{th} feature function and θ_i is the weight of $f_i(W(t), W(s))$. We use a five-fold cross-validation method [33] to learn the weight θ_i.

Since BERT and MFAF are supervised methods, we construct the training data with $600k$ samples, based on an intuition: if two questions are replied by the same best answer, they are relevant, and their templates are also relevant and taken as a positive example. Moreover, we randomly sample some questions without common answers and take their templates as negative samples. The proportion of positive and negative samples is $1:1$. Since the baselines do not consider candidate retrieval, we use the embedding-based retrieval model [34] to find candidates for baselines.

1. https://github.com/google-research/bert

To further verify the effectiveness of the Bayesian inference framework in PMI decomposition model, we conduct the ablation study where different solutions are used for estimating PMI. Since $\text{PMI}(x; y) = \ln p(x|y)\, p(y|x)$ where x=t and y=S (Eq. 6), we design four alternative approaches to estimate the two probabilities $p(t|S)$ and $p(S|t)$, instead of the proposed PMI decomposition model:

• PMI based on the bag of words representation (PMI_W). It represents both S and t as bag of word models, and uses the language model for information retrieval [30] to estimate the two probabilities:

$$p(t|S) = \prod_{w \in t} p(w|S)^{n(w,t)} \tag{20}$$

$$p(S|t) = \prod_{w \in S} p(w|t)^{n(w,S)} \tag{21}$$

where the component $p(w|S)$ (or $p(w|t)$) is the probability of generating the word w from S (or t), which is estimated by using the maximum likelihood estimation with Jelinek-Mercer smoothing. The component $n(w, t)$ (or $n(w, S)$) is the number of w occurring in t (or S).

• PMI based on answer representation (PMI_A). We represent both S and t as bag of answers. For the template t, we firstly identify the questions instantiated by t, and take the answers of the instantiated questions as the answers of t. We design answer-based probability estimation models as follows:

$$p(t|S) = \sum_{s \in S} \sum_{a \in \mathcal{A}(s)} p(t|a) \cdot p(a|s) \cdot p(s|S) \tag{22}$$

The $p(t|a)$ is estimated by $\frac{n(t,a)}{\sum_{t_1 \in T} n(t_1,a)}$ where $n(t, a)$ is the frequency of a occurring in answer representation of t. The $p(a|s)$ is estimated by $\frac{n(s,a)}{\sum_{a_1 \in A} n(s,a_1)}$ where $n(s, a)$ is the frequency of a occurring in the answer representation of s. The $p(s|S)$ is estimated by $\frac{1}{|S|}$ where $|S|$ is the size of S.

$$p(S|t) = \sum_{s \in S} \sum_{a \in \mathcal{A}(t)} p(S|s) \cdot p(s|a) \cdot p(a|t) \tag{23}$$

The $p(a|t)$ is estimated by $\frac{n(t,a)}{sum_{a_1 \in A} n(t,a_1)}$, and $p(s|a)$ is estimated by $\frac{n(s,a)}{\sum_{q \in Q} n(q,a)}$ where Q is the whole question set and $n(q, a)$ is the frequency of a occurring in the answer representation of q. We assign $p(S|s)$ as 1, since s is in S.

• PMI over a different graph (PMI_G). We run our models over a different graph whose nodes and edges are the same as those of the utility-flow graph, but the edge weights are estimated by text similarity (Jaccard set similarity) between questions and answers, as well as questions and templates, instead of the number of usages used in PMI.

• PMI with the shortest path inference over the utility-flow graph (PMI_{SPI}). On the utility-flow graph, if the templates are closer to seeds S, the templates are likely more relevant to the intention of S. Based on the intuition, $p(t|S) = p(S|t) = \frac{1}{|S|} \cdot \sum_{s \in S} \frac{1}{hop(t,s)}$ where $hop(t, s)$ is the hop number of the shortest path between t and s.

Both PMI_W and PMI_A are used for verifying the effectiveness of the bag of usages in PMI. The PMI_G is used for verifying the effectiveness of the utility-flow graph in PMI. The PMI_{SPI} is used for verifying the effectiveness of popularity and relevance inferences in PMI.

Test Set	Model	N=10	N=30	N=50	N=100	N=300
Baidu	*BERT*	0.008‡	0.027‡	0.044‡	0.080‡	0.184‡
	MFAE	0.007‡	0.017‡	0.031‡	0.062‡	0.139‡
	LHZ	0.045‡	0.075‡	0.093‡	0.138‡	0.218‡
	ZZL	0.027‡	0.052‡	0.069‡	0.095‡	0.190‡
	PMI	**0.093**	**0.199**	**0.285**	**0.435**	**0.636**
Amazon	*BERT*	0.003‡	0.011‡	0.016‡	0.032‡	0.077‡
	MFAE	0.004‡	0.012‡	0.029‡	0.052‡	0.080‡
	LHZ	0.004‡	0.011‡	0.020‡	0.037‡	0.057‡
	ZZL	0.019‡	0.037‡	0.047‡	0.064‡	0.104‡
	PMI	**0.086**	**0.283**	**0.379**	**0.558**	**0.737**
Yahoo	*BERT*	0.096‡	0.210‡	0.288‡	0.383†	0.538
	MFAE	0.046‡	0.111‡	0.153‡	0.261‡	0.421‡
	LHZ	0.187‡	0.263‡	0.288‡	0.315‡	0.398‡
	ZZL	0.147‡	0.195‡	0.227‡	0.272‡	0.368‡
	PMI	**0.292**	**0.350**	**0.414**	**0.467**	**0.578**
Overall	*BERT*	0.046‡	0.100‡	0.130‡	0.187‡	0.272‡
	MFAE	0.019‡	0.046‡	0.070‡	0.124‡	0.211‡
	LHZ	0.090‡	0.130‡	0.148‡	0.173‡	0.227‡
	ZZL	0.077‡	0.110‡	0.130‡	0.161‡	0.232‡
	PMI	**0.155**	**0.277**	**0.359**	**0.487**	**0.651**

TABLE 3: Overall performance comparisons (PRD).

Model	N=10	N=30	N=50	N=100	N=300
PR quality					
BERT	0.059‡	0.116‡	0.146‡	0.202‡	0.290‡
MFAE	0.022‡	0.057‡	0.084‡	0.144‡	0.238‡
LHZ	0.106‡	0.145‡	0.160‡	0.185‡	0.243‡
ZZL	0.099‡	0.135‡	0.155‡	0.187‡	0.263‡
PMI	0.147	0.268	0.348	0.471	0.630
PD quality					
BERT	0.045‡	0.098‡	0.126‡	0.182‡	0.262‡
MFAE	0.019‡	0.046‡	0.070‡	0.124‡	0.212‡
LHZ	0.090‡	0.129‡	0.146‡	0.171‡	0.225‡
ZZL	0.077‡	0.110‡	0.130‡	0.160‡	0.230‡
PMI	0.160	0.283	0.365	0.492	0.654
RD quality					
BERT	0.202‡	0.195‡	0.177‡	0.164‡	0.141‡
MFAE	0.146‡	0.142‡	0.132‡	0.125‡	0.108‡
LHZ	0.323‡	0.306‡	0.257‡	0.234‡	0.178‡
ZZL	0.258‡	0.229‡	0.180‡	0.159‡	0.130‡
PMI	0.423	0.399	0.363	0.343	0.317

TABLE 4: Comparisons of quality combinations.

Model	Train data	F1-measure	Accuracy
BERT	D_{Base}	0.777	0.713
	D_{PMI}	0.897	0.906
	D_{Ant}	0.710	0.598
	D_{Label}	0.953	0.950
MFAE	D_{Base}	0.715	0.601
	D_{PMI}	0.881	0.878
	D_{Ant}	0.772	0.765
	D_{Label}	0.916	0.910

TABLE 5: An application of the data mined by PMI.

Parameter Setting. For our models, we keep all parameters with default values unless we test the impact of an individual parameter. We set $\alpha = 0.5$ (Eq. 12), $\beta_1 = 0.5$ and $\beta_2 = 0.5$ (Eq. 15). For the baseline models, we adopt the five-fold cross validation method [33] for parameter tuning. To measure the statistical significance, a two-tailed paired t-test is applied. We use ‡ and † to denote the differences at 0.01 and 0.05 levels. The significance is tested against PMI model.

6.2 Experimental Results

6.2.1 Overall Quality Comparisons

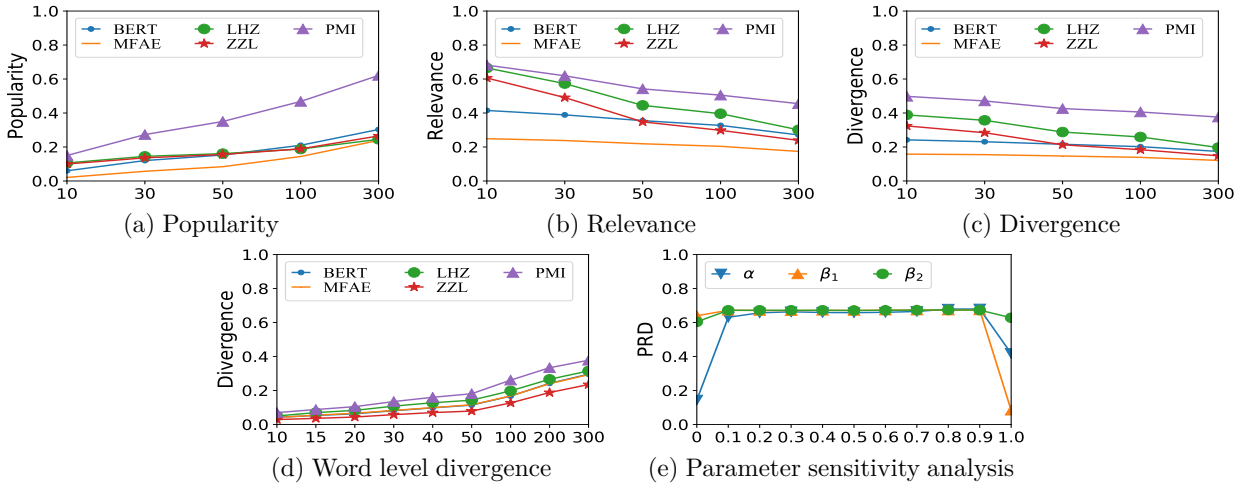
We first report the PRD performance of our model and baselines in Tab. 3 on three individual test sets and their mix (Overall). It can be seen that PMI significantly outperforms all the baselines in all the measures. This is reasonable because PMI captures all the three criteria of intention templates, while baselines can not effectively capture the popularity and divergence. Specifically, the utility-centric modeling in PMI decouples relevance from textual similarity and addresses the paradox of relevance and divergence so that it is able to discover both divergent and relevant results. The divergence and relevance metrics (RD) of all models are reported in Tab. 4 where PMI achieves the best RD metric; The PMI decomposition balances the conflict of relevance and popularity so that it is able to discover both relevant and popular results. We further compare the popularity and relevance metrics (PR) of all models in Tab. 4, and find PMI significantly outperforms all baselines.

All baselines can not effectively address the semantic gap between candidates and input questions. LHZ and ZZL apply answers to bridge the semantic gap, but they suffer from the semantic gap of answers. $BERT$ and $MFAE$ are distant-supervised methods, which are sensitive to the quality of training data. They create training samples based on an assumption: if two questions have the same best answer, their templates are relevant [35]. However, the collected training samples contains some noisy data. The noisy samples are from questions with the same general answers. For example, given two irrelevant questions "My phone is very hot" and "I can not watch TV by my phone" having the same general answer "I also encounter this problem and sympathize with you", the assumption will judge the two questions as relevant. In contrast, supported by the quality propagation principle over the utility-flow graph, PMI model can propagate quality between nodes without direct edges so that PMI can effectively bridge the semantic gap of answers. The quality propagation principle are based on the iterative Bayesian inference which can effectively reduce the effect of noisy data. So PMI can achieve the best overall quality.

As an application of our model PMI, the intention paraphrase templates mined by PMI can be used for constructing a large number of training data for the paraphrase identification task. If two questions can be instantiated by the paraphrase templates of an intention, they are regarded as a question paraphrase pair which is a positive sample. Since the number of mined paraphrase question pairs is very large, we randomly select $300k$ mined pairs as positive samples, and collect $300k$ random question pairs as negative samples. The training data is denoted as D_{PMI}. Besides, we also adopt other approaches to construct training samples. Firstly, we collect $300k$ positive training samples, based on the assumption that if two questions have the same best answer, they are relevant [35], and collect $300k$ random question pairs as negative samples (denoted as D_{Base}). Secondly, we apply the training samples of the paraphrase identification task [2] published by Ant-financial as one training data (denoted as D_{Ant}). Thirdly, we manually label $40k$ question pairs as a training data (denoted as D_{Label}). As the test set of the paraphrase identification task, we manually label 9000 question pairs with relevant or irrelevant annotations.

To test the quality of the four training data, we run the BERT [22] and MFAE [23] for paraphrase identification task and report the results in Tab. 5. It can be seen that using data D_{PMI} achieves significant performance improvement over using D_{Base} and D_{Ant}. This verifies the effectiveness of the data D_{PMI}. Using D_{Ant} can not achieve performance

2. https://dc.cloud.alipay.com/index#/topic/data?id=12



(a) Popularity (b) Relevance (c) Divergence

(d) Word level divergence (e) Parameter sensitivity analysis

Fig. 5: Individual metric comparisons and parameter sensitivity analysis (on the Overall test set).

improvement, because D_{Ant} is financial domain, while the test set is phone domain. This illustrates that cross-domain labeling data can not significantly improve the performance. Using D_{PMI} can not achieve better performance than using D_{Label}, because the data D_{Label} is manually labeled with the highest quality. Overall, our proposed PMI can automatically mine paraphrase question pairs with high quality.

6.2.2 Individual Quality Comparisons

In Fig. 5, we compare individual metrics R, P and D of both our model and baselines. The comparisons further verify the advantage of PMI. It can be seen that the advantage of PMI is more prominent on P (Fig. 5 (a)) and D (Fig. 5 (c)) than that on R (Fig. 5 (b)). This is because the baselines cannot effectively capture the popularity and divergence of candidates, but they can capture the relevance. Compared to baselines, PMI still achieves the best relevance metrics. The reason is that baselines suffer the semantic gap of answers and noisy data of train samples, while PMI can effectively bridge the semantic gap of answers by using quality propagation principle over utility-flow graph and effectively reduce the effect of noisy data by designing iterative Bayesian inferences (popularity and relevance inferences).

To clearly show the advantage of PMI, we present two test cases and top mined results from PMI and LHZ models in Tab. 6 and 7. The score of each quality is ground truth. If the score is larger, the quality is higher. In Tab. 6, the relevance of mined results from PMI is higher than that from LHZ, because LHZ mines two irrelevant results with relevance 0.00. Besides, the results of PMI have higher popularity (P) and divergence (D) qualities than that of LHZ. As well, the result comparison in Tab. 7 also shows that the results of PMI have higher popularity and divergence qualities than those of LHZ.

Since the performance metric D in Fig. 5 (c) is query-level divergence, we also propose a word-level divergence method to further analyze the ability of mining divergent templates. Specifically, we firstly construct the word-level ground truth θ_I by collecting words in all ground truth templates; Secondly, we identify the relevant results from top-N results, and construct word-level representation θ_N by collecting words in the relevant results from top-N results. We define the word-level divergence as $\frac{|\theta_I \cap \theta_N|}{|\theta_I|}$ where $|\theta_I|$ is the word number of θ_I. We present the word-level divergence of all models in Fig. 5 (d). We see that the curve of PMI is higher than that of all baselines, showing that PMI achieves the best word-level divergence.

6.2.3 Improvements of Baselines

As one contribution of this paper, we propose two different approaches to estimate the popularity, i.e., the point-wise mutual information with only the popularity inference (PMI_p) and the number of utilities ($|\mathcal{U}(t)|$). To verify the effectiveness of the two approaches, we improve baselines by incorporating PMI_p or $\mathcal{U}(t)$:

$$Score(t,S) = score(t,S) \cdot PMI_p(t) \tag{24}$$

$$Score(t,S) = score(t,S) \cdot score(t,S) \cdot |\mathcal{U}(t)| \tag{25}$$

where $score(t,S)$ is the score estimated by baseline models, and $PMI_p(t)$ is the popularity score estimated by PMI_p. The improved baselines are denoted as $+PMI_p$ and $+Uti$, respectively. We present the results of improved baselines in Tab. 8. The performance of original models is all improved when popularity measures are introduced in the ranking functions (i.e., Eq. 24 and Eq. 25). This verifies that the popularity inference model PMI_p can effectively capture the popularity of candidates. PMI still outperforms the improved baselines, even though the improved baselines with $+Uti$ actually exploit the ground truth information. This verifies the effectiveness of utility-centric modeling and the popularity inference in PMI.

We also try to improve baselines by incorporating the divergence measure div(t): $Score(t,S) = score(t,S) \cdot \text{div}(t)$ where div(t) is defined in Eq. 1. The results of improved baselines are presented in Tab. 9. It can be seen that incorporating div(t) results in performance degradation. This is because baseline models can not address the paradox of relevance and divergence. Specifically, the baseline models estimate the relevance based on the text similarity, but the divergence is estimated based the text difference. The text similarity conflicts with the text difference.

PMI	PRD	P	D	R	*LHZ*	PRD	P	D	R
# 指纹键失灵的好多有你么	1.29E-03	1.87E-03	0.69	1.00	为什么 # 触屏有点不灵敏	2.37E-06	2.37E-04	0.01	1.00
# 触屏不灵如何进行触屏校准	8.49E-03	1.31E-02	0.65	1.00	# 屏幕失灵	2.20E-03	2.57E-03	0.86	1.00
# 不小心摔一下结果就屏幕失灵怎么办	7.28E-03	1.14E-02	0.64	1.00	# 屏幕失灵该怎么办	2.07E-04	2.77E-04	0.75	1.00
# 触屏校准怎么进不去	4.52E-03	7.35E-03	0.61	1.00	# 触屏出现方块设置	0.00E+00	0.00E+00	0.00	0.00
# 屏幕失灵怎么办	1.52E-02	2.03E-02	0.75	1.00	# 按键失灵了要怎么办	0.00E+00	0.00E+00	0.00	0.00
# 屏幕摔裂了触屏不能用了还能修好吗	6.08E-03	8.08E-03	0.75	1.00	# 屏幕失灵了怎么回事	5.17E-05	6.94E-05	0.75	1.00
# 屏幕触摸没反应怎么办	1.73E-02	2.32E-02	0.75	1.00	# 触屏不灵什么情况	1.47E-05	3.47E-05	0.42	1.00
# 屏幕触摸没反应怎么解决	2.83E-03	3.78E-03	0.75	1.00	# 部分屏幕有些失灵怎么办	4.32E-05	6.94E-05	0.62	1.00
# 触屏不管用怎么强制关机	7.54E-04	1.21E-03	0.62	1.00	# 屏幕失灵有什么办法解决	3.76E-05	6.94E-05	0.54	1.00
# 为什么触屏不灵敏	3.09E-05	1.73E-04	0.18	1.00	# 触屏不灵如何进行校准	8.49E-03	1.31E-02	0.65	1.00

TABLE 6: Example: top results of "为什么华为 mate8 触屏有点不灵敏" from Baidu Knows. # refers to a phone name

PMI	PRD	P	D	R	*LHZ*	PRD	P	D	R
dropped # in water. it now won't # on anyone know how can i fix it ? please help	3.7e-2	5.1e-2	0.74	1.00	i dropped my # into a sink full of water	1.5e-3	2.6e-3	0.58	1.00
i dropped my # in a bucket of water. will it work again	5.0e-2	8.4e-2	0.59	1.00	i dropped my # to water.	2.3e-2	4.7e-2	0.50	1.00
i dropped my # in water almost 24hrs ago and it still won't turn on...any advice	6.7e-2	1.5e-1	0.47	1.00	dropped # in toilet, help	2.2e-3	2.6e-3	0.84	1.00
how do you save a # if it gets dropped in water	2.3e-2	3.2e-2	0.72	1.00	# after being dropped into water	1.7e-3	2.6e-3	0.69	1.00
i dropped me # in water, whats a good way to revive it	3.1e-2	4.1e-2	0.72	1.00	dropped # in water. it now won't # on anyone know how can i fix it ? please help	3.8e-2	5.1e-2	0.74	1.00
# after being dropped into water	2.6e-3	2.6e-3	0.99	1.0	what do i do if i drop my # in water	1.8e-2	3.2e-2	0.57	1.00
i have dropped my # in a bucket of water how do i get it to work	1.5e-2	3.3e-2	0.47	1.00	how do you save a # if it gets dropped in water	2.3e-2	3.2e-2	0.72	1.00
is there a way to make my # work after it's been dropped in the water	2.8e-2	4.7e-2	0.60	1.00	my daughter has dropped my # 3260i in the dog bowl of water. help	4.4e-2	7.4e-2	0.60	1.00

TABLE 7: Example: top results of "i dropped my iphone 4 in water help" from Yahoo Answers. # refers to a phone name

Model	N=10	N=30	N=50	N=100	N=300
BERT	0.046	0.100	0.130	0.187	0.272
$BERT + PMI_p$	0.118	0.185	0.209	0.247	0.294
$BERT + Uti$	0.094	0.161	0.192	0.236	0.299
MFAE	0.019	0.046	0.070	0.124	0.211
$MFAE + PMI_p$	0.104	0.143	0.161	0.194	0.234
$MFAE + Uti$	0.089	0.131	0.157	0.199	0.250
LHZ	0.090	0.130	0.148	0.173	0.227
$LHZ + PMI_p$	0.106	0.143	0.163	0.192	0.230
$LHZ + Uti$	0.112	0.169	0.191	0.220	0.260
ZZL	0.077	0.110	0.130	0.161	0.232
$ZZL + PMI_p$	0.105	0.153	0.178	0.204	0.247
$ZZL + Uti$	0.117	0.164	0.188	0.213	0.266
PMI	0.155	0.277	0.359	0.487	0.651

TABLE 8: Improving models by the popularity measures.

Model	N=10	N=30	N=50	N=100	N=300
BERT	0.046	0.100	0.130	0.187	0.272
$BERT + Div$	0.005	0.026	0.047	0.075	0.174
MFAE	0.019	0.046	0.070	0.124	0.211
$MFAE + Div$	0.004	0.015	0.033	0.059	0.128
LHZ	0.090	0.130	0.148	0.173	0.227
$LHZ + Div$	0.025	0.043	0.057	0.087	0.148
ZZL	0.077	0.110	0.130	0.161	0.232
$ZZL + Div$	0.016	0.035	0.047	0.068	0.134
PMI	0.155	0.277	0.359	0.487	0.651

TABLE 9: Improving models by the divergence measure.

6.2.4 Ablation Study

To verify the effectiveness of components in PMI, we conduct an ablation study. We report the results of PMI_{SPI}, PMI_G, PMI_W, PMI_A and PMI in Tab. 10. Firstly, PMI achieves significant PRD improvement over PMI_W. This illustrates that the bag-of-usages model of PMI is more effective than the bag-of-words model of PMI_W. The BOW model encounters serious semantic gap problem, while the utility-centric modeling can address the semantic gap by propagating usages over the utility-flow graph. PMI outperforms PMI_A because PMI_A encounters the problem of semantic gap in answers. This illustrates 1) the utility-centric model is better than the bag of answers model; 2) the quality propagation principle over the utility-flow graph can bridge the semantic gap of answers. The significant performance gap of PMI and PMI_G verifies the effectiveness of the utility-flow graph. Besides, this

Model	N=10	N=30	N=50	N=100	N=300
PMI_{SPI}	$0.035^{\ddagger}$	$0.054^{\ddagger}$	$0.067^{\ddagger}$	$0.083^{\ddagger}$	$0.095^{\ddagger}$
PMI_G	$0.070^{\ddagger}$	$0.145^{\ddagger}$	$0.178^{\ddagger}$	$0.250^{\ddagger}$	$0.329^{\ddagger}$
PMI_W	$0.032^{\ddagger}$	$0.049^{\ddagger}$	$0.057^{\ddagger}$	$0.076^{\ddagger}$	$0.116^{\ddagger}$
PMI_A	$0.028^{\ddagger}$	$0.061^{\ddagger}$	$0.070^{\ddagger}$	$0.091^{\ddagger}$	$0.145^{\ddagger}$
PMI	0.093	0.199	0.285	0.435	0.636

TABLE 10: Ablation study results over Baidu.

also illustrates that the text similarity between a question and its answer can not accurately capture their relevance, while the utilities between them can capture their relevance more accurately. The performance comparison of PMI and PMI_{SPI} reveals that the iterative Bayesian inference used in PMI is better than the shortest path inference used in PMI_{SPI}. The shortest path inference suffers the effect of noisy data, while the iterative Bayesian inference can effectively reduce the effect of noisy data.

6.2.5 Parameter Sensitivity

We conducted the parameter analysis in Fig. 5 (f). When investigating the impact of one parameter, we set other parameters with default values. Fig. 5 (f) shows the impact of α in Eq. 12 and that of β_1 and β_2 in Eq. 15. When α, β_1 and β_2 vary from 0.1 to 0.9, the achieved PRD does not fluctuate significantly. This shows that our model is not sensitive to parameters α, β_1 and β_2.

7 Conclusion

We propose a new task of intention paraphrase template discovery, identify new metrics and challenges. We propose the utility-centric model and the principled PMI decomposition model to address the challenges. By experiments, we find 1) CQA corpus is a valuable source to mine knowledge; 2) The point-wise mutual information can be interpreted as the product of popularity and relevance; 3) It is non-trivial to use answers to effectively bridge semantic gap of questions, because of the semantic gap of answers; 4) The iterative Bayesian inference on the utility-flow graph can reduce the effect of noise answers on the intention template discovery.

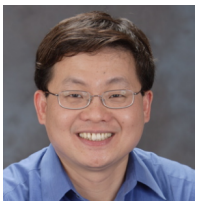
Acknowledgments

Yueguo Chen is supported by National Key Research and Development Program (No. 2020YFB1710004) and the National Science Foundation of China under grants U1711261. Kevin Chen-Chuan Chang is supported by the National Science Foundation IIS 16-19302 and IIS 16-33755, Zhejiang University ZJU Research 083650, Futurewei Technologies HF2017060011 and 094013, UIUC OVCR CCIL Planning Grant 434S34, UIUC CSBS Small Grant 434C8U, and IBM-Illinois Center for Cognitive Computing Systems Research (C3SR)- a research collaboration as part of the IBM Cognitive Horizon Network. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

[1] D. Ravichandran and et al., "Learning surface text patterns for a question answering system," in *ACL , 2002*.
[2] G. Agarwal and et al., "Towards rich query interpretation: walking back and forth for mining query templates," in *WWW 2010*.
[3] X. Cao and et al., "The use of categorization information in language models for question retrieval," in *CIKM, 2009*.
[4] G. Zhou and et al., "Improving question retrieval in community question answering using world knowledge," in *IJCAI, 2013*.
[5] S. Zhao and et al., "Learning question paraphrases for QA from encarta logs," in *IJCAI, 2007*.
[6] Q. Peng and et al., "Predicate-argument based bi-encoder for paraphrase identification," in *ACL 2022*.
[7] K. R. McKeown, "Paraphrasing questions using given and new information," *American Journal of Computational Linguistics, 1983*.
[8] K. Yang and et al., "GCPG: A general framework for controllable paraphrase generation," in *ACL 2022*.
[9] A. Ormazabal and et al., "Principled paraphrase generation with parallel corpora," in *ACL 2022*.
[10] B. Dolan and et al., "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *COLING 2004*.
[11] E. Pavlick and et al., "Domain-specific paraphrase extraction," in *ACL 2015*.
[12] Y. Seonwoo and et al., "Two-step question retrieval for open-domain QA," in *ACL 2022*.
[13] W. Zhang and et al., "Capturing the semantics of key phrases using multiple languages for question retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 888–900, 2016.
[14] G. Zhou and et al., "Modeling and learning distributed word representation with metadata for question retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1226–1239, 2017.
[15] D. Shah and et al., "Adversarial domain adaptation for duplicate question detection," in *EMNLP, 2018*.
[16] Y. Wu and et al., "Mining query subtopics from questions in community question answering," in *AAAI, 2015*.
[17] I. Haponchyk and et al., "Supervised clustering of questions into intents for dialog system applications," in *EMNLP 2018*.
[18] Y. Li and et al., "Unsupervised identification of synonymous query intent templates for attribute intents," in *CIKM, 2013*.
[19] N. P. A. Vo and et al., "Paraphrase identification and semantic similarity in twitter with simple features," in *NAACL 2015*.
[20] L. Dong and et al., "Learning to paraphrase for question answering," in *EMNLP, 2017*.
[21] W. Lan and W. Xu, "Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering," in *COLING 2018*.
[22] J. Devlin and et al., "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT 2019*.
[23] R. Zhang and et al., "What do questions exactly ask? MFAE: duplicate question identification with multi-fusion asking emphasis," in *ICDM 2020*.
[24] W. Yin and et al., "Convolutional neural network for paraphrase identification," in *NAACL HLT 2015*.
[25] H. He and et al., "Multi-perspective sentence similarity modeling with convolutional neural networks," in *EMNLP*.
[26] B. Ko and et al., "Paraphrase bidirectional transformer with multi-task learning," in *BigComp 2020*.
[27] D. R. Kubal and et al., "A hybrid deep learning architecture for paraphrase identification," in *ICCCNT 2018*.
[28] B. Agarwal and et al., "A deep network model for paraphrase detection in short text messages," *Inf. Process. Manag. 2017*.
[29] A. Skabar and et al., "Clustering sentence-level text using a novel fuzzy relational clustering algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 62–75, 2013.
[30] C. Zhai and et al., "A study of smoothing methods for language models applied to ad hoc information retrieval," in *SIGIR, 2001*.
[31] G. Bouma, "Normalized (pointwise) mutual information in collocation extraction," *Biennial GSCL 2009*.
[32] K. Balog and et al., "Query modeling for entity search based on terms, categories, and examples," *TOIS 2011*.
[33] D. Metzler and et al., "Linear feature-based models for information retrieval," *Inf. Retr., 2007*.
[34] T. Kenter and et al., "Short text similarity with word embeddings," in *CIKM 2015*.
[35] J. Jeon and et al., "Finding similar questions in large question and answer archives," in *CIKM, 2005*.

Denghao Ma is currently a senior algorithm researcher of Meituan, responsible for the construction of intention knowledge graph of Meituan. He received his Ph.D degree in computer science from Renmin University of China in 2020. His research interest includes web data mining, knowledge graph construction, information retrieval, question answering.



Kevin Chen-Chuan Chang is a Professor in the Department of Computer Science, University of Illinois at Urbana-Champaign. His research addresses large-scale information access, for search, mining, and integration across structured and unstructured big data including Web data and social media. He also co-founded Cazoodle for deepening vertical data-aware search over the Web.

Yueguo Chen is a PhD Supervisor and Professor in the School of Information, Renmin University of China. He received the BS and Master degree in Mechanical Engineering and Control Engineering from Tsinghua University in 2001 and 2004, and earned Ph.D. degree in Computer Science from National University of Singapore. His research interests include interactive analysis systems of big data, knowledge graph construction, semantic search.

Cheng Chen is a PhD candidate in the school of information, Renmin University of China. His research tries to model and organize user Intent from natural language questions. His research is related to the broad area of modern NLP technologies and data visualization.

Chuanfei Xu received the BE degree from the Shenyang University of Technology in 2007 and the ME and PhD degrees in computer science from Northeastern University in 2009 and 2013. He is currently a researcher in Huawei Tech Co., Ltd. His research interests include search engine and NLP.