A User-Centered Concept Mining System for Query and Document Understanding at Tencent

Bang Liu[1*], Weidong Guo[2*], Di Niu[1], Chaoyue Wang[2]
Shunnan Xu[2], Jinghong Lin[2], Kunfeng Lai[2], Yu Xu[2]
[1]University of Alberta, Edmonton, AB, Canada
[2]Platform and Content Group, Tencent, Shenzhen, China

ABSTRACT

Concepts embody the knowledge of the world and facilitate the cognitive processes of human beings. Mining concepts from web documents and constructing the corresponding taxonomy are core research problems in text understanding and support many downstream tasks such as query analysis, knowledge base construction, recommendation, and search. However, we argue that most prior studies extract formal and overly general concepts from Wikipedia or static web pages, which are not representing the user perspective. In this paper, we describe our experience of implementing and deploying *ConcepT* in Tencent QQ Browser. It discovers user-centered concepts at the right granularity conforming to user interests, by mining a large amount of user queries and interactive search click logs. The extracted concepts have the proper granularity, are consistent with user language styles and are dynamically updated. We further present our techniques to tag documents with user-centered concepts and to construct a *topic-concept-instance* taxonomy, which has helped to improve search as well as news feeds recommendation in Tencent QQ Browser. We performed extensive offline evaluation to demonstrate that our approach could extract concepts of higher quality compared to several other existing methods. Our system has been deployed in Tencent QQ Browser. Results from online A/B testing involving a large number of real users suggest that the Impression Efficiency of feeds users increased by 6.01% after incorporating the user-centered concepts into the recommendation framework of Tencent QQ Browser.

CCS CONCEPTS

• **Information systems → Information retrieval query processing**; **Query intent**; • **Applied computing → Document analysis**.

KEYWORDS

Concept Mining; Concept Tagging; Taxonomy Construction; Query Understanding; Document Understanding

ACM Reference Format:
Bang Liu[1*], Weidong Guo[2*], Di Niu[1], Chaoyue Wang[2] and Shunnan Xu[2], Jinghong Lin[2], Kunfeng Lai[2], Yu Xu[2]. 2019. A User-Centered Concept Mining System for Query and Document Understanding at Tencent. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330727

1 INTRODUCTION

The capability of *conceptualization* is a critical ability in natural language understanding and is an important distinguishing factor that separates a human being from the current dominating machine intelligence based on vectorization. For example, by observing the words "Honda Civic" and "Hyundai Elantra", a human can immediately link them with "fuel-efficient cars" or "economy cars", and quickly come up with similar items like "Nissan Versa" and probably "Ford Focus". When one observes the seemingly uncorrelated words "beer", "diaper" and "Walmart", one can extrapolate that the article is most likely discussing topics like marketing, business intelligence or even data science, instead of talking about the actual department store "Walmart". The importance of concepts is best emphasized by the statement in Gregory Murphy's famous book *The Big Book of Concepts* that "Concepts embody our knowledge of the kinds of things there are in the world. Tying our past experiences to our present interactions with the environment, they enable us to recognize and understand new objects and events."

In order to enable machines to extract concepts from text, a large amount of effort has been devoted to knowledge base or taxonomy construction, typically represented by DBPedia [8] and YAGO [19] which construct taxonomies from Wikipedia categories, and Probase [22] which extracts concepts from free text in web documents. However, we argue that these methods for concept extraction and taxonomy construction are still limited as compared to how a human interacts with the world and learns to conceptualize, and may not possess the proper granularity that represents human interests. For example, "Toyota 4Runner" is a "Toyota SUV" and "F150" is a "truck". However, it would be more helpful if we can infer that a user searching for these items may be more interested in "cars with high chassis" or "off-road ability" rather than another Toyota SUV like "RAV4"—these concepts are rare in existing knowledge bases or taxonomies. Similarly, if an article talks about the movies "the Great Gatsby", "Wuthering Heights" and "Jane Eyre", it is also hard to infer that the article is actually about "book-to-film adaptations". The fundamental reason is that taxonomies such as DBPedia [8] and Probase [22], although maintaining structured knowledge about the world, are not designed to conceptualize from the *user's perspective* or to infer the user intention. Neither can they

[*] These authors contributed equally to this work.

KDD '19, August 4–8, 2019, Anchorage, AK, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330727

exhaust all the complex connections between different instances, concepts and topics that are discussed in different documents. Undoubtedly, the ability for machines to conceptualize just as a user would do—to extract trending and user-centered concept terms that are constantly evolving and are expressed in user language—is critical to boosting the intelligence of recommender systems and search engines.

In this paper, we propose *ConcepT*, a concept mining system at Tencent that aims to discover concepts at the right granularity conforming to user interests. Different from prior work, *ConcepT* is not based on mining web pages only, but mining from huge amounts of query logs and search click graphs, thus being able to understand user intention by capturing their interaction with the content. We present our design of *ConcepT* and our experience of deploying it in Tencent QQ Browser, which has the largest market share in Chinese mobile browser market with more than 110 millions daily active users. *ConcepT* serves as the core taxonomy system in Tencent QQ Browser to discover both time-invariant and trending concepts.

ConcepT can significantly boost the performance of both searching and content recommendation, through the taxonomy constructed from the discovered user-centered concepts as well as a concept tagging mechanism for both short queries and long documents that accurately depict user intention and document coverage. Up to today, *ConcepT* has extracted more than 200, 000 high-quality user-centered concepts from daily query logs and user click graphs in QQ Browser, while still growing at a rate of 11, 000 new concepts found per day. Although our system is implemented and deployed for processing Chinese query and documents, the proposed techniques in *ConcepT* can easily be adapted to other languages.

Mining user-centered concepts from query logs and search click graphs has brought about a number of new challenges. First, most existing taxonomy construction approaches such as Probase [22] extract concepts based on Hearst patterns, like "such as", "especially", etc. However, Hearst patterns have limited extraction power, since high-quality patterns are often missing in short text like queries and informal user language. Moreover, existing methods extract concepts from web pages and documents that are usually written by experts in the *writer perspective*. However, search queries are often informal and may not observe the syntax of a written language [6]. Hence, it is hard if not impossible to mine "user perspective" concepts based on predefined syntax patterns.

There are also many studies on keyphrase extraction [10, 12, 16]. They measure the importance or quality of all the N-grams in a document or text corpus, and choose keyphrases from them according to the calculated scores. As a result, such methods can only extract continuous text chunks, whereas a concept may be discontinuous or may not even be explicitly mentioned in a query or a document. Another concern is that most of such N-gram keyphrase extraction algorithms yield poor performance on short text snippets such as queries. In addition, deep learning models, such as sequence-to-sequence, can also be used to generate or extract concepts. However, deep learning models usually rely on large amounts of high-quality training data. For user-centered concept mining, manually labeling such a dataset from scratch is extremely costly and time consuming.

Furthermore, many concepts in user queries are related to recent trending events whereas the concepts in existing taxonomies are mostly stable and time-invariant. A user may search for "Films for New Year (贺岁大片)" or "New Japanese Animation in April (四月新番)" in Tencent QQ Browser. The semantics of such concepts are evolving over time, since apparently we have different new animations or films in different years. Therefore, in contrast to existing taxonomies which mostly maintain long-term stable knowledge, it will be challenging yet beneficial if we can also extract time-varying concepts and dynamically update the taxonomies constructed.

We make the following novel contributions in the design of *ConcepT*:

First, we extract candidate user-centered concepts from vast query logs by two unsupervised strategies: 1) bootstrapping based on pattern-concept duality: a small number of predefined string patterns can be used to find new concepts while the found concepts can in turn be used to expand the pool of such patterns; 2) query-title alignment: an important concept in a query would repeat itself in the document title clicked by the user that has input the query.

Second, we further train a supervised sequence labeling Conditional Random Field (CRF) and a discriminator based on the initial seed concept set obtained, to generalize concept extraction and control the concept quality. These methods are complementary to each other and are best suited for different cases. Evaluation based on a labeled test dataset has shown that our proposed concept discovery procedure significantly outperforms a number of existing schemes.

Third, we propose effective strategies to tag documents with potentially complex concepts to depict document coverage, mainly by combining two methods: 1) matching key instances in a document with their concepts if their *isA* relationships exist in the corresponding constructed taxonomy; 2) using a probabilistic inference framework to estimate the probability of a concept provided that an instance is observed in its context. Note that the second method can handle the case when the concept words do not even appear in the document. For example, we may associate an article containing "celery", "whole wheat bread" and "tomato" with the concept "diet for weight loss" that a lot of users are interested in, even if the document does not have exact wording for "weight loss" but has context words such as "fibre", "healthy", and "hydrated".

Last but not least, we have constructed and maintained a three-layered *topic-concept-instance* taxonomy, by identifying the *isA* relationships among instances, concepts and topics based on machine learning methods, including deep neural networks and probabilistic models. Such a user-centered taxonomy significantly helps with query and document understanding at varying granularities.

We have evaluated the performance of *ConcepT*, and observed that it can improve both searching and recommendation results, through both offline experiments and a large-scale online A/B test on more than 800, 000 real users conducted in the QQ Browser mobile app. The experimental results reveal that our proposed methods can extract concepts more accurately from Internet user queries in contrast to a variety of existing approaches. Moreover, by performing query conceptualization based on the extracted concepts and the correspondingly constructed taxonomy, we can improve the results of search engine according to a pilot user experience study in our experiments. Finally, *ConcepT* also leads to a higher Impression Efficiency as well as user duration in the real world according to the large-scale online A/B test on the recommender system in feeds stream (text digest content recommended to users in a stream as they scroll down in the mobile app). The results suggest that the

Impression Efficiency of the users increases by 6.01% when *ConcepT* system is incorporated for feeds stream recommendation.

2 USER-CENTERED CONCEPT MINING

Our objective of user-centered concept mining is to derive a word/phrase from a given user query which can best characterize this query and its related click logs at the proper granularity.

Denote a user query by $q = w_1^q w_2^q \cdots w_{|q|}^q$, which is a sequence of words. Let Q be the set of all queries. Denote a document title by $t = w_1^t w_2^t \cdots w_{|t|}^t$, another sequence of words. Given a user query q and its corresponding top-ranked clicked titles $T^q = \{t_1^q, t_2^q, \cdots, t_{|T^q|}^q\}$ from query logs, we aim to extract a concept phrase $\mathbf{c} = w_1^c w_2^c \cdots w_{|\mathbf{c}|}^c$ that represents the main semantics or the intention of the query. Each word $w_i^c \in \mathbf{c}$ belongs to either the query q or one of the corresponding clicked titles $t_j^q \in T^q$.

An overview of the detailed steps of user-centered concept mining from queries and query logs in *ConcepT* is shown in Fig. 1, which mainly consists of three approaches: pattern-concept bootstrapping, query-title alignment, as well as supervised sequence labeling. All the extracted concepts are further filtered by a discriminator. We utilize bootstrapping and query-title alignment to automatically accumulate an initial seed set of *query-concept* pairs, which can help to train sequence labeling and the discriminator, to extract a larger amount of concepts more accurately.

Bootstrapping by Pattern-Concept Duality. We first extract an initial set of seed concepts by applying the bootstrapping idea [1] only to the set of user queries Q (without the clicked titles). Bootstrapping exploits *Pattern-Concept Duality*, which is:

- Given a set of patterns, we can extract a set of concepts from queries following these patterns.
- Given a set of queries with extracted concepts, we can learn a set of patterns.

Fig. 1 (a) illustrates how bootstrapping is performed on queries Q. First, we manually define a small set of patterns which can be used to accurately extract concept phrases from queries with high confidence. For example, "Top 10 XXX (十大XXX)" is a pattern (with original Chinese expression in parenthesis) that can be used to extract seed concepts. Based on this pattern, we can extract concepts: "fuel-efficient cars (省油的汽车)" and "gaming phones (游戏手机)" from the queries "Top 10 fuel-efficient cars (十大省油的汽车)" and "Top 10 gaming phones (十大游戏手机)", respectively.

We can in turn retrieve more queries that contain these extracted concepts and derive new patterns from these queries. For example, a query "Which gaming phones have the best performance? (哪款游戏手机性能好?)" also contains the concept "gaming phones (游戏手机)". Based on this query, a new pattern "Which XXX have the best performance? (哪款XXX性能好?)" is found.

We also need to shortlist and control the quality of the patterns found in each round. Intuitively speaking, a pattern is valuable if it can be used to accurately extract a portion of existing concepts as well as to discover new concepts from queries. However, if the pattern is too general and appears in a lot of queries, it may introduce noise. For example, a pattern "Is XXX good? (XXX好不好?)" underlies a lot of queries including "Is the fuel-efficient car good? (省油的车好不好?)" and "Is running everyday good (每天跑步好不好?)", whereas "running everyday (每天跑步)" does not serve as a sufficiently important concept in our system. Therefore, given a new pattern $\mathbf{p}$ found in a certain round, let n_s be the number of concepts in the existing seed concept set that can be extracted from query set Q by $\mathbf{p}$. Let n_e be the number of new concepts that can be extracted by $\mathbf{p}$ from Q. We will keep the pattern $\mathbf{p}$ if it satisfies: 1) $\alpha < \frac{n_s}{n_e} < \beta$, and 2) $n_s > \delta$, where $\alpha, \beta,$ and δ are predefined thresholds. (We set $\alpha = 0.6,\ \beta = 0.8,$ and $\delta = 2$ in our system.)

Concept mining by query-title alignment. Although bootstrapping helps to discover new patterns and concepts from the query set Q in an iterative manner, such a pattern-based method has limited extraction power. Since there are a limited number of high-quality syntax patterns in queries, the recall rate of concept extraction has been sacrificed for precision. Therefore, we further propose to extract concepts from both a query and its top clicked link titles in the query log.

The intuition is that a concept in a query will also be mentioned in the clicked titles associated with the query, yet possibly in a more detailed manner. For example, "The last Hong Kong zombie movie (香港|最后|一|部|僵尸|电影)" or "Hong Kong zombie comedy movie (香港|搞笑|僵尸|电影)" convey more specific concepts of the query "Hong Kong zombie movie (香港|僵尸|电影)" that leads to the click of these titles. Therefore, we propose to find such concepts based on the alignment of queries with their corresponding clicked titles. The steps are listed in the following:

(1) Given a query q, we retrieve the top clicked titles $T^q = \{t_1^q, t_2^q, \cdots, t_{|T^q|}^q\}$ from the query logs of q, i.e., T^q consists of document titles that are clicked by users for more than N times during the past D days ($N = 5$ and $D = 30$ in our system).

(2) For query q and each title $t \in T^q$, we enumerate all the N-grams in them.

(3) Let N-gram $g_{in}^q = w_i^q w_{i+1}^q \cdots w_{i+n-1}^q$ denote a text chunk of length n starting from position i of query q, and $g_{jm}^t = w_j^t w_{j+1}^t \cdots w_{j+m-1}^t$ denote a text chunk of length m starting from position j of title t. For each pair of such N-grams, $< g_{in}^q, g_{jm}^t >$, we identify g_{jm}^t as a candidate concept if: i) g_{jm}^t contains all the words of g_{in}^q in the same order; ii) $w_i^q = w_j^t$, and $w_{i+n-1}^q = w_{j+m-1}^t$.

Query-title alignment extends concept extraction from query set alone to concept discovery based on the query logs, thus incorporating some information of the user's interaction into the system.

Supervised sequence labeling. The above unsupervised methods are still limited in their generalizability. We further perform supervised learning and train a Conditional Random Field (CRF) to label the sequence of concept words in a query or a title, where the training dataset stems from the results of the bootstrapping and query-title alignment process mentioned above, combined with human reviews as detailed in the Appendix. Specifically, each word is represented by its tag features, e.g., Part-of-Speech or Named Entity Recognition tags, and the contextual features, e.g., the tag features of its previous word and succeeding word, the combination pattern of tags of contextual words and the word itself. These features are fed into a CRF to yield a sequence of labels, identifying the concept chunk, as shown in Fig. 1.

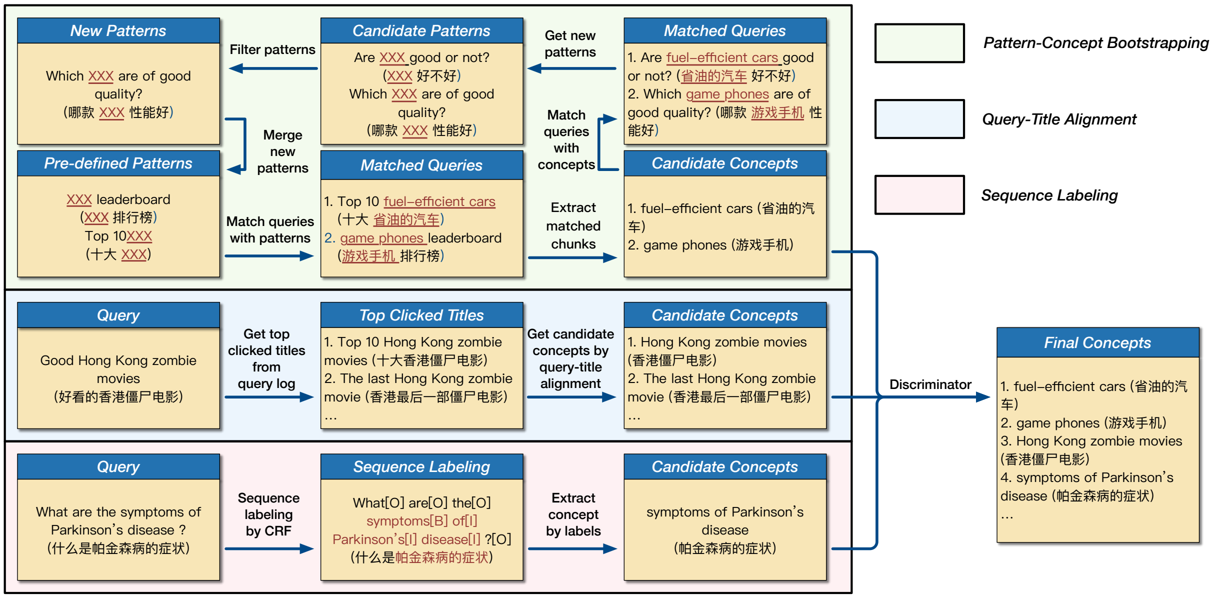

New Patterns
Which XXX are of good quality?
(哪款 XXX 性能好)
Filter patterns
Candidate Patterns
Are XXX good or not?
(XXX 好不好)
Which XXX are of good quality?
(哪款 XXX 性能好)
Get new patterns
Matched Queries
1. Are fuel–efficient cars good or not? (省油的汽车 好不好)
2. Which game phones are of good quality? (哪款 游戏手机 性能好)
Merge new patterns
Match queries with concepts
Pre-defined Patterns
XXX leaderboard
(XXX 排行榜)
Top 10XXX
(十大 XXX)
Match queries with patterns
Matched Queries
1. Top 10 fuel–efficient cars
(十大 省油的汽车)
2. game phones leaderboard
(游戏手机 排行榜)
Extract matched chunks
Candidate Concepts
1. fuel–efficient cars (省油的汽车)
2. game phones (游戏手机)
Query
Good Hong Kong zombie movies
(好看的香港僵尸电影)
Get top clicked titles from query log
Top Clicked Titles
1. Top 10 Hong Kong zombie movies (十大香港僵尸电影)
2. The last Hong Kong zombie movie (香港最后一部僵尸电影)
...
Get candidate concepts by query-title alignment
Candidate Concepts
1. Hong Kong zombie movies (香港僵尸电影)
2. The last Hong Kong zombie movie (香港最后一部僵尸电影)
...
Discriminator
Final Concepts
1. fuel–efficient cars (省油的汽车)
2. game phones (游戏手机)
3. Hong Kong zombie movies (香港僵尸电影)
4. symptoms of Parkinson's disease (帕金森病的症状)
...
Query
What are the symptoms of Parkinson's disease ?
(什么是帕金森病的症状)
Sequence labeling by CRF
Sequence Labeling
What[O] are[O] the[O] symptoms[B] of[I] Parkinson's[I] disease[I] ?[O]
(什么是帕金森病的症状)
Extract concept by labels
Candidate Concepts
symptoms of Parkinson's disease
(帕金森病的症状)
Pattern-Concept Bootstrapping
Query-Title Alignment
Sequence Labeling

Figure 1: The overall process of concept mining from user queries and query logs.

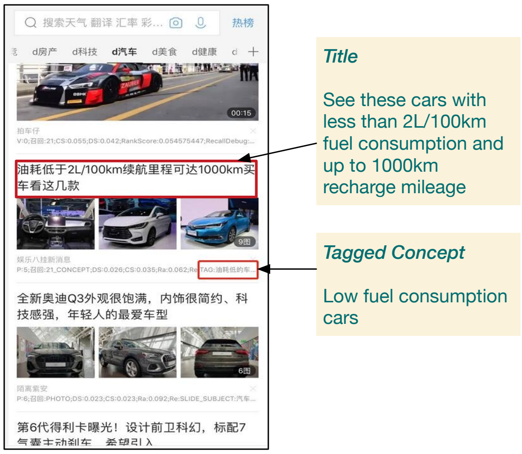
The above approaches for concept mining are complementary to each other. Our experience shows that CRF can better extract concepts from short text when they have clear boundary with surrounding non-concept words, e.g., "What cars are fuel-efficient (省油的汽车有哪些)". However, when the concept is split into multiple pieces, e.g., "What gifts should we prepare for birthdays of parents? (父母过生日准备什么礼物?)", the query-title alignment approach can better capture the concept that is scattered in a query.

A Discriminator for quality control. Given the concepts extracted by above various strategies, we need to evaluate their value. For example, in Fig. 1, the concept "The last Hong Kong zombie movie (香港|最后|一|部|僵尸|电影)" is too fine-grained and maybe only a small amount of users are interested in searching it. Therefore, we further train a classifier to determine whether each discovered concept is worth keeping.

We represent each candidate concept by a variety of its features such as whether this concept has ever appeared as a query, how many times it has been searched and so on (more details in Appendix). With these features serving as the input, we train a classifier, combining Gradient Boosting Decision Tree (GBDT) and Logistic Regression, to decide whether to accept the candidate concept in the final list or not. The training dataset for the discriminator is manually created. We manually check a found concept to see whether it is good (positive) or not sufficiently good (negative). Our experience reveals that we only need 300 samples to train such a discriminator. Therefore, the creation of the dataset incurs minimum overhead.

3 DOCUMENT TAGGING AND TAXONOMY CONSTRUCTION

In this section, we describe our strategies for tagging each document with pertinent user-centered concepts to depict its coverage. Based on document tagging, we further construct a 3-layered *topic-concept-instance* taxonomy which helps with feeds recommendation in Tencent QQ Browser.



拍车仔
V:0;召回:21;CS:0.055;DS:0.042;RankScore:0.054575447;RecallDebug:...

油耗低于2L/100km续航里程可达1000km买车看这几款

娱乐八挂新消息
P:5;召回:21_CONCEPT;DS:0.026;CS:0.035;Ra:0.062;Re TAG:油耗低的车...

全新奥迪Q3外观很饱满，内饰很简约、科技感强，年轻人的最爱车型

陌离紫安
P:6;召回:PHOTO;DS:0.023;CS:0.023;Ra:0.092;Re:SLIDE_SUBJECT:汽车...

第6代得利卡曝光！设计前卫科幻，标配7气囊主动刹车，希望引入

Title

See these cars with less than 2L/100km fuel consumption and up to 1000km recharge mileage

Tagged Concept

Low fuel consumption cars

Figure 2: Example of concept tagging for documents in the feeds stream of Tencent QQ Browser.

3.1 Concept Tagging for Documents

While the extracted concepts can characterize the implicit intention of user queries, they can also be used to describe the main topics of a document. Fig. 2 shows an example of concept tagging in Tencent QQ Browser based on the *ConcepT* system. Suppose that a document titled "See these cars with less than 2L/100km fuel consumption and up to 1000km recharge mileage" can be tagged with the concept "low fuel-consumption cars", even though the title never explicitly mentions these concept words. Such concept tags for documents, if available, can help improve search and recommendation performance. Therefore, we propose to perform concept tagging for documents.

Given a document d and a set of concepts $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{|\mathbf{C}|}\}$, our problem is selecting a subset of concepts $\mathbf{C}^d = \{\mathbf{c}^d_1, \mathbf{c}^d_2, \cdots, \mathbf{c}^d_{|\mathbf{C}^d|}\}$ from $\mathbf{C}$ that are most related to the content of d. Fig. 3 presents the

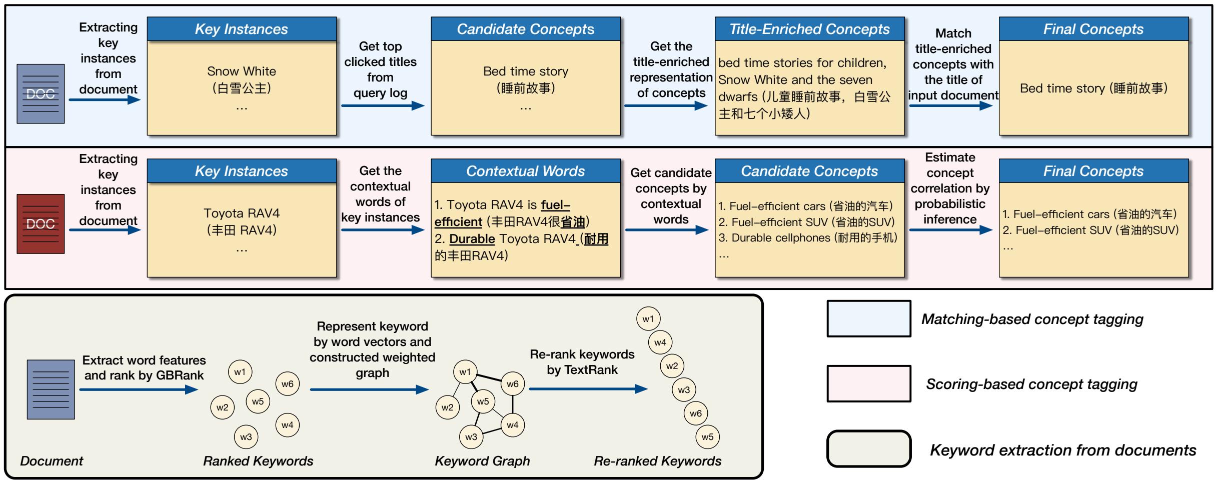

DOC
Extracting key instances from document
Key Instances
Snow White
(白雪公主)
...
Get top clicked titles from query log
Candidate Concepts
Bed time story
(睡前故事)
...
Get the title-enriched representation of concepts
Title-Enriched Concepts
bed time stories for children, Snow White and the seven dwarfs (儿童睡前故事，白雪公主和七个小矮人)
Match title-enriched concepts with the title of input document
Final Concepts
Bed time story (睡前故事)
DOC
Extracting key instances from document
Key Instances
Toyota RAV4
(丰田 RAV4)
...
Get the contextual words of key instances
Contextual Words
1. Toyota RAV4 is fuel–efficient (丰田RAV4很省油)
2. Durable Toyota RAV4 (耐用的丰田RAV4)
Get candidate concepts by contextual words
Candidate Concepts
1. Fuel–efficient cars (省油的汽车)
2. Fuel–efficient SUV (省油的SUV)
3. Durable cellphones (耐用的手机)
...
Estimate concept correlation by probabilistic inference
Final Concepts
1. Fuel–efficient cars (省油的汽车)
2. Fuel–efficient SUV (省油的SUV)
...
Extract word features and rank by GBRank
Represent keyword by word vectors and constructed weighted graph
Re-rank keywords by TextRank
w1
w6
w2
w5
w4
w3
w1
w6
w2
w5
w4
w3
w1
w4
w2
w3
w6
w5
Document
Ranked Keywords
Keyword Graph
Re-ranked Keywords
Matching-based concept tagging
Scoring-based concept tagging
Keyword extraction from documents

Figure 3: The overall procedures of concept tagging for documents. We combine both a matching-based approach with a scoring-based approach to handle different situations.

procedures of concept tagging for documents. To link appropriate concepts with a document, we propose a probabilistic inference-based approach, together with a matching-based method to handle different situations.

Specifically, our approach estimates the correlation between a concept and a document through the key instances contained in the document. When no direct *isA* relationship can be found between the key instances in the document and the concepts, we use probabilistic inference as a general approach to identify relevant concepts by utilizing the context information of the instances in the document. Otherwise, the matching-based approach retrieves candidate concepts which have *isA* relationships with the key instances in a taxonomy we have constructed (which be explained at the end of this section). After that, it scores the coherence between a concept and a document based on the title-enriched representation of the concept.

Key instance extraction. Fig. 3 shows our approach for key instance extraction. Firstly, we rank document words using GBRank [25] algorithm, based on word frequency, POS tag, NER tag, etc. Secondly, we represent each word by word vectors proposed in [17], and construct a weighted undirected graph for top K words (we set $K = 10$). The edge weight is calculated by the cosine similarity of two word vectors. We then re-rank the keywords by TextRank [12] algorithm. Finally, we only keep keywords with ranking scores larger than δ_w (we use 0.5). From our experience, combining GBRank and word-vector-based TextRank helps to extract keywords that are more coherent to the topic of document.

Concept tagging by probabilistic inference. Denote the probability that concept $\mathbf{c}$ is related to document d as $p(\mathbf{c}|d)$. We propose to estimate it by:

$$p(\mathbf{c}|d) = \sum_{i=1}^{|E^d|} p(\mathbf{c}|\mathbf{e}_i^d) p(\mathbf{e}_i^d|d), \tag{1}$$

where E^d is the key instance set of d, and $p(\mathbf{e}_i^d|d)$ is the document frequency of instance $\mathbf{e}_i^d \in E^d$. $p(\mathbf{c}|\mathbf{e}_i^d)$ estimates the probability of concept $\mathbf{c}$ given $\mathbf{e}_i^d$. However, as the *isA* relationship between $\mathbf{e}_i^d$ and $\mathbf{c}$ may be missing, we further infer the conditional probability by taking the contextual words of $\mathbf{e}_i^d$ into account:

$$p(\mathbf{c}|\mathbf{e}_i^d) = \sum_{j=1}^{|X_{E^d}|} p(\mathbf{c}|\mathbf{x}_j) p(\mathbf{x}_j|\mathbf{e}_i^d) \tag{2}$$

$p(\mathbf{x}_j|\mathbf{e}_i^d)$ is the co-occurrence probability of context word $\mathbf{x}_j$ with e_i^d. We consider two words as co-occurred if they are contained in the same sentence. X_{E^d} are the set of contextual words of $\mathbf{e}_i^d$ in d. Denote $\mathbf{C}^{\mathbf{x}_j}$ as the set of concepts containing $\mathbf{x}_j$ as a substring. $p(\mathbf{c}|\mathbf{x}_j)$ is defined as:

$$p(\mathbf{c}|\mathbf{x}_j) = \begin{cases} \frac{1}{|\mathbf{C}^{\mathbf{x}_j}|}, & \text{if } \mathbf{x}_j \text{is a substring of } \mathbf{c}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

For example, in Fig. 3, suppose $\mathbf{e}_i^d$ extracted from d is "Toyota RAV4 (丰田RAV4)". We may haven't establish any relationship between this instance and any concept. However, we can extract contextual words "fuel-efficient (省油)" and "durable (耐用)" from d. Based on these contextual words, we can retrieve candidate concepts that containing these words, such as "fuel-efficient cars (省油的汽车)" and "durable cellphones (耐用的手机)". We then estimate the probability of each candidate concept by above equations.

Concept tagging by matching. The probabilistic inference-based approach decomposes the correlation between a concept and a document through the key instances and their contextual words in the document. However, whenever the *isA* relationship between the key instances of d and $\mathbf{C}$ is available, we can utilize it to get candidate concepts directly, and calculate the matching score between each candidate concept and d to decide which concepts are coherent to the document.

First, we introduce how the *isA* relationship between *concept-instance* pairs can be identified. On one hand, given a concept, we retrieve queries/titles containing the same modifier in the context and extract the instances contained in the queries/titles. For example, given concept "fuel-efficient cars (省油的汽车)", we may retrieve a query/title "fuel-efficient Toyota RAV4 (省油的丰田RAV4)", and

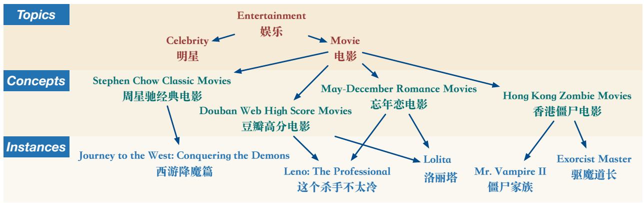


Figure 4: An example to show the extracted topic-concept-instance hierarchy.

extract instance "Toyota RAV4 (丰田RAV4)" from the query/title, as it shares the same modifier "fuel-efficient (省油的)" with the given concept. After we getting a candidate instance **e**, we estimate $p(\mathbf{c}|\mathbf{e})$ by Eqn. (2). On another hand, we can also extract *concept-instance* pairs from various semi-structured websites where a lot of *concept-instance* pairs are stored in web tables.

Second, we describe our matching-based approach for concept tagging. Let $E^d = \{\mathbf{e}_1^d, \mathbf{e}_2^d, \cdots, \mathbf{e}_{|E^d|}^d\}$ donate the set of key instances extracted from d, and $C^d = \{\mathbf{c}_1^d, \mathbf{c}_2^d, \cdots, \mathbf{c}_{|C^d|}^d\}$ donate the retrieved candidate concepts by the *isA* relationship of instances in E^d. For each candidate concept $\mathbf{c}_i^d$, we enrich its representation by concatenating the concept itself with the top N (we use 5) titles of user clicked links. We then represent enriched concept and the document title by TF-IDF vectors, and calculate the cosine similarity between them. If $sim(\mathbf{c}, d) > \delta_u$ (we set it as 0.58), we tag **c** to d; otherwise we reject it. Note that the *isA* relationship between *concept-instance* pairs and the enriched representation of concepts are all created in advance and stored in a database.

Fig. 3 shows an example of matching-based concept tagging. Suppose we extract key instance "Snow White (白雪公主)" from a document, we can retrieve related concepts "bed time stories (睡前故事)" and "fairy tales (童话故事)" based on *isA* relationship. The two concepts are further enriched by the concatenation of top clicked titles. Finally, we match candidate concepts with the original document, and keep highly related concepts.

3.2 Taxonomy Construction

We have also constructed a *topic-concept-instance* taxonomy based on the concepts extracted from queries and query logs. It can reveal the hierarchical relationship between different topics, concepts and instances. Currently our constructed taxonomy consists of 31 pre-defined topics, more than 200, 000 user-centered concepts, and more than 600, 000 instances. Among them, 40,000 concepts contain at least one instance, and 200,000 instances have been identified with a *isA* relationship with at least one concept. Based on the taxonomy, we can improve the user experience in search engines by understanding user implicit intention via query conceptualization, as well as enhance the recommendation performance by matching users and documents at different semantic granularities. We will demonstrate such improvements in detail in Sec. 4.

Fig. 4 shows a three-layered taxonomy that consists of topics, concepts and instances. The taxonomy is a Directed Acyclic Graph (DAG). Each node is either a topic, a concept or an instance. We predefined a list that contains $N_t = 31$ different topics, including entertainment, technology, society and so on. The directed edges indicate *isA* relationships between nodes.

We have already introduced our approach for *isA* relationship between *concept-instance* pairs. We need to further identify the relationship between *topic-concept* pairs. First, we represent each document as a vector through word embedding and pooling, and perform topic classification for documents through a carefully designed deep neural network (see Appendix for details). After that, given a concept **c** and a topic **p**, suppose there are $n^{\mathbf{c}}$ documents that are correlated to concept **c**, and among them there are $n_{\mathbf{p}}^{\mathbf{c}}$ documents that belong to topic **p**. We then estimate $p(\mathbf{p}|\mathbf{c})$ by $p(\mathbf{p}|\mathbf{c}) = n_{\mathbf{p}}^{\mathbf{c}}/n^{\mathbf{c}}$. We identify the *isA* relationship between **c** and **p** if $p(\mathbf{p}|\mathbf{c}) > \delta_t$ (we set $\delta_t = 0.3$). Our experience shows that most of the concepts belong to one or two topics.

4 EVALUATION

In this section, we first introduce a new dataset for the problem of concept mining from user queries and query logs, and compare our proposed approach with variety of baseline methods. We then evaluate the accuracy of the taxonomy constructed from extracted user-centered concepts, and show that it can improve search engine results by query rewriting. Finally, we run large-scale online A/B testing to show that the concept tagging on documents significantly improves the performance of recommendation in real world.

We deploy the *ConcepT* system which includes the capability of concept mining, tagging, and taxonomy construction in Tencent QQ Browser. For offline concept mining, our current system is able to extract around 27,000 concepts on a daily basis, where about 11,000 new concepts are new ones. For online concept tagging, our system processes 40 documents per second. More details about implementation and deployment can be found in appendix.

4.1 Evaluation of Concept Mining

The User-Centered Concept Mining Dataset (UCCM). As user-centered concept mining from queries is a relative new research problem and there is no public dataset available for evaluation, we created a large-scale dataset containing 10, 000 samples. Our *UCCM* dataset is sampled from the queries and query logs of Tencent QQ Broswer, from November 11, 2017 to July 1, 2018. For each query, we keep the document titles clicked by more than 2 users in previous day. Each sample consists of a query, the top clicked titles from real world query log, and a concept phrase labeled by 3 professional product managers in Tencent and 1 PhD student. We have published the *UCCM* dataset for research purposes [1].

Methodology and Compared Models. We evaluate our comprehensive concept mining approach with the following baseline methods and variants of our method:

- **TextRank [12]**. The classical graph-based ranking model for keyword extraction.[2]
- **THUCKE [11]**. It regards keyphrase extraction as a problem of translation, and learns translation probabilities between the words in input text and the words in keyphrases.[3]

[1] https://github.com/BangLiu/ConcepT
[2] https://github.com/letiantian/TextRank4ZH
[3] https://github.com/thunlp/THUCKE

Table 1: Compare different algorithms for concept mining.

Method	Exact Match	F1 Score
TextRank	0.1941	0.7356
THUCKE	0.1909	0.7107
AutoPhrase	0.0725	0.4839
Q-Pattern	0.1537	0.3133
T-Pattern	0.2583	0.5046
Q-CRF	0.2631	0.7322
T-CRF	0.3937	0.7892
QT-Align	0.1684	0.3162
Our approach	**0.8121**	**0.9541**

- **AutoPhrase [16]**. A state-of-the-art quality phrase mining algorithm that extracts quality phrases based on knowledge base and POS-guided segmentation.[4]
- **Pattern-based matching with query (Q-Pattern)**. Extract concepts from queries based on patterns from bootstrapping.
- **Pattern-based matching with title (T-Pattern)**. Extract concepts from titles based on patterns from bootstrapping.
- **CRF-based sequence labeling with query (Q-CRF)**. Extract concepts from queries by CRF.
- **CRF-based sequence labeling with titles (T-CRF)**. Extract concepts from click titles by CRF.
- **Query-Title alignment (QT-Align)**. Extract concepts by query-title alignment strategy.

For the T-Pattern and T-CRF approach, as each click title will give a result, we select the most common one as the final result given a specific query. For the TextRank, THUCKE, and AutoPhrase algorithm, we take the concatenation of user query and click titles as input, and extract the top 5 keywords or phrases. We then keep the keywords/phrases contained in the query and concatenate them in the same order as in the query, then use it as the final result.

We use Exact Match (EM) and F1 to evaluate the performance. The exact match score is 1 if the prediction is exactly the same as groundtruth or 0 otherwise. F1 measures the portion of overlap tokens between the predicted phrase and the groundtruth concept.

Evaluation results and analysis. Table 1 compares our model with different baselines on the UCCM dataset in terms of Exact Match and F1 score. Results demonstrate that our method achieves the best EM and F1 score. This is because: first, the pattern-based concept mining with bootstrapping helps us to construct a collection of high-quality patterns which can accurately extract concepts from queries in an unsupervised manner. Second, the combination of sequence labeling by CRF and query-title alignment can recognize concepts from both queries and click titles under different situations, i.e., either the concept boundary in query is clear or not.

We can see the methods based on TextRank [12], THUCKE [11] and AutoPhrase [16] do not give satisfactory performance. That is because existing keyphrases extraction approaches are better suited for extracting keywords or phrases from a long document or a corpus. In contrast, our approach is specially designed for the problem of concept mining from user queries and click titles. Comparing our approach with its variants, including Q-Pattern, Q-CRF, T-CRF and QT-Align, we can see that each component cannot achieve comparable performance as ours independently. This demonstrates the effectiveness of combining different strategies in our system.

Table 2: Evaluation results of constructed taxonomy.

Metrics / Statistics	Value
Mean #Instances per Concept	3.44
Max #Instances per Concept	59
isA Relationship Accuracy	96.59%

Table 3: Part of the *topic-concept-instance* samples created by *ConcepT* system.

Topics	Concepts	Instances
Entertainment (娱乐)	Movies adapted from a novel (小说改编成的电影)	The Great Gatsby (了不起的盖茨比), Anna Karenina (安娜·卡列尼娜), Jane Eyre (简爱)
Entertainment (娱乐)	Female stars with a beautiful smile (笑容最美的女明星)	Ayase Haruka (绫濑遥), Sasaki Nozomi (佐佐木希), Dilraba (迪丽热巴)
Society (社会)	Belt and Road countries along the route (一带一路沿线国家)	Palestine (巴勒斯坦), Syria (叙利亚), Mongolia (蒙古), Oman (阿曼)
Games (游戏)	Mobile game for office workers (适合上班族玩的手游)	Pokemon (口袋妖怪), Invincible Asia (东方不败)

4.2 Evaluation of Document Tagging and Taxonomy Construction

Evaluation of document tagging. For concept tagging on documents, our system currently processes around 96,700 documents per day, where about 35% of them can be tagged with at least one concept. We create a dataset containing 11, 547 documents with concept tags for parameter tuning, and we also open-source it for research purpose (see appendix for more details). We evaluate the performance of concept tagging based on this dataset. The result shows that the precision of concept tagging for documents is 96%. As the correlated concept phrases may even not show in the text, we do not evaluate the recall rate.

Evaluation of taxonomy construction. We randomly sample 1000 concepts from our taxonomy. As the relationships between *concept-instance* pairs are critical to query and document understanding, our experiment mainly focus on evaluating them. For each concept, we check whether the *isA* relationship between it and its instances is correct. We ask three human judges to evaluate them. For each concept, we record the number of correct instances and the number of incorrect ones.

Table 2 shows the evaluation results. The average number of instances for each concept is 3.44, and the maximum concept contains 59 instances. Note that the scale of our taxonomy is keep growing with more daily user queries and query logs. For the *isA* relationships between *concept-instance* pairs, the accuracy is 96.59%.

[4] https://github.com/shangjingbo1226/AutoPhrase

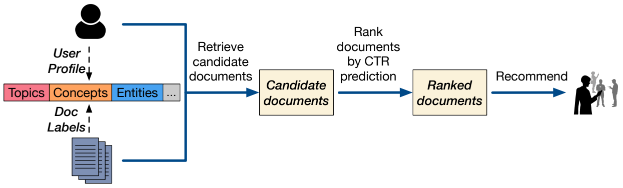


Figure 5: The framework of feeds recommendation in Tencent QQ Browser.

Table 3 shows a part of *topic-concept-instance* tuples from our taxonomy. We can see that the extracted concepts are expressed from “user perspective”, such as “Female stars with a beautiful smile (笑容最美的女明星)” or “Mobile game for office workers (适合上班族玩的手游)”. At the same time, the relationships between concepts and instances are also established based on user activities. For example, when a certain number of users click the documents related to “Sasaki Nozomi (佐佐木希)” when they are searching “Female stars with a beautiful smile (笑容最美的女明星)”, our system will be able to recognize the *isA* relationship between the concept and the instance.

4.3 Online A/B Testing for Recommendation

We perform large-scale online A/B testing to show how concept tagging on documents helps with improving the performance of recommendation in real world applications. Fig. 5 illustrates the recommendation architecture based on our *ConcepT* system. In our system, both users and documents are tagged with interested or related topics, concepts and instances. We first retrieve candidate documents by matching users with documents, then we rank candidate documents by a Click-Through Rate (CTR) prediction model. The ranked documents are pushed to users in the feeds stream of Tencent QQ Browser.

For online A/B testing, we split users into buckets where each bucket contains 800, 000 of users. We first observe and record the activities of each bucket for 7 days based on the following metrics:

- **Impression Page View (IPV)**: number of pages that matched with users.
- **Impression User View (IUV)**: number of users who has matched pages.
- **Click Page View (CPV)**: number of pages that users clicked.
- **Click User View (CUV)**: number of users who clicked pages.
- **User Conversion Rate (UCR)**: $\frac{CUV}{IUV}$.
- **Average User Consumption (AUC)**: $\frac{CPV}{CUV}$.
- **Users Duration (UD)**: average time users spend on a page.
- **Impression Efficiency (IE)**: $\frac{CPV}{IUV}$.

We then select two buckets with highly similar activities. For one bucket, we perform recommendation without the concept tags of documents. For another one, the concept tags of documents are utilized for recommendation. We run our A/B testing for 3 days and compare the result by above metrics. The Impression Efficiency (IE) and Users Duration (UD) are the two most critical metrics in real world application, because they show how many contents users read and how much time they spend on an application.

Table 4: Online A/B testing results.

Metrics	Percentage Lift	Metrics	Percentage Lift
IPV	+0.69%	UCR	+0.04%
IUV	+0.06%	AUC	+0.21%
CPV	+0.38%	UD	**+0.83%**
CUV	+0.16%	IE	**+6.01%**

Table 4 shows the results of our online A/B testing. In the online experiment, we observe a statistically significant IE gain (6.01%) and user duration (0.83%). The page views and user views for click or impression, as well as user conversation rate and average user consumptions, are all improved. These observations prove that the concept tagging for documents greatly benefits the understanding of documents and helps to better match users with their potential interested documents. With the help of user-centered concepts, we can better capture the contained topics in a document even if it does not explicitly mention them. Given more matched documents, users spend more times and reading more articles in our feeds.

4.4 Offline User Study of Query Rewriting for Searching

Here we evaluate how user-centered concept mining can help with improving the results of search engines by query rewriting based on conceptualization. We create a evaluation dataset which contains 108 queries from Tencent QQ Browser. For each query **q**, we analyze the concept **c** conveyed in the query, and rewrite the query by concatenating each of the instances $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n\} \in \mathbf{c}$ with q. The rewritten queries are in the format of “**q** $\mathbf{e}_i$”. For the original query, we collect the top 10 search results returned by Baidu search engine, the largest search engine in China. Assume we replace a query by K different instances. We collect top $\lceil \frac{10}{K} \rceil$ search results from Baidu for each of the rewritten queries, combining and keeping 10 of them as the search result after query rewriting.

We ask three human judges to evaluate the relevancy of the results. For each search reuslt, we record majority vote, i.e., “relevant” or “not relevant”, of the human judges, and calculate the percentage of relevance of original queries and rewritten queries. Our evaluation results show that the percentage of relevant top 10 results increases from 73.1% to 85.1% after rewriting the queries with our strategy. The reason is that the concept mining for user queries helps to understand the intention of user queries, and concatenating the instances belonging to the concept with the original query provides the search engine more relevant and explicit keywords. Therefore, the search results will better match user’s intention.

5 RELATED WORK

Our work is mainly related to the following research lines.

Concept Mining. Existing research work on concept mining mainly relies on predefined linguistic templates, statistical signals, knowledge bases or concept quality. Traditional approaches for concept mining are closely related to the work of noun phrase chunking and named entity recognition [13]. They either employ heuristics, such as fixed POS-tag patterns, to extract typed entities [15], or consider the problem as sequence tagging and utilize large-scale labeled training data to train complex deep neural models based on LSTM-CRF or CNN [7]. Another line of work focus on terminology and keyphrase extraction. They extract noun phrases based on statistical occurrence and co-occurrence signals [4], semantic information from knowledge base [21] or textual features [14]. Recent approaches for concept mining rely on phrase quality. [10, 16] adaptively recognize concepts based on concept quality. They exploit various statistical features such as popularity, informativeness, POS tag sequence ans so on to measure phrase quality, and train the concept quality scoring function by using knowledge base entity names as training labels.

Text Conceptualization. Conceptualization seeks to map a word or a phrase to a set of concepts as a mechanism of understanding short text such as search queries. Since short text usually lack of context, conceptualization helps better make sense of text data by extending the text with categorical or topical information, and therefore facilitates many applications. [9] performs query expansion by utilizing Wikipedia as external corpus to understand query for improving ad-hoc retrieval performance. [18] groups instances by their conceptual similarity, and develop a Bayesian inference mechanism to conceptualize each group. To make further use of context information, [20] utilize a knowledge base that maps instances to their concepts, and build a knowledge base that maps non-instance words, including verbs and adjectives, to concepts.

Relation Extraction. Relation Extraction (RE) is to identify relations between entities and concepts automatically. Generally speaking, Relation Extraction techniques can be classified into several categories: 1) supervised techniques including features-based [5] and kernel based [2] methods, 2) semi-supervised approaches including bootstrapping [1], 3) unsupervised methods [23], 4) Open Information Extraction [3], and 5) distant supervision based techniques [24]. In our work, we combine unsupervised approaches, semi-supervised bootstrapping technique, and supervised sequence labeling algorithm to extract concepts and identify the relationship between entities and concepts.

6 CONCLUSION

In this paper, we describe our experience of implementing *ConcepT*, a user-centered concept mining and tagging system at Tencent that designed to improve the understanding of both queries and long documents. Our system extracts user-centered concepts from a large amount of user queries and query logs, as well as performs concept tagging on documents to characterize the coverage of documents from user-perspective. In addition, *ConcepT* further identifies the *isA* relationship between concepts, instances and topics to constructs a 3-layered *topic-concept-instance* taxonomy. We conduct extensive performance evaluation through both offline experiments and online large-scale A/B testing in the QQ Browser mobile application on more than 800, 000 real users. The results show that our system can extract featured, user-centered concepts accurately from user queries and query logs, and it is quite helpful for both search engines and recommendation systems. For search engines, the pilot user study in our experiments shows that we improve the results of search engine by query conceptualization. For recommendation, according to the real-world large-scale online A/B testing, the Impression Efficiency improves by 6.01% when incorporating *ConcepT* system for feeds recommendation in Tencent QQ Browser.

REFERENCES

[1] Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *International Workshop on The World Wide Web and Databases*. Springer, 172–183.

[2] Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd annual meeting on association for computational linguistics*. ACL, 423.

[3] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*. ACL, 1535–1545.

[4] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International journal on digital libraries* 3, 2 (2000), 115–130.

[5] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. ACL, 427–434.

[6] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2015. Short text understanding through lexical-semantic analysis. IEEE, 495–506.

[7] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[8] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195.

[9] Yinghao Li, Wing Pong Robert Luk, Kei Shiu Edward Ho, and Fu Lai Korris Chung. 2007. Improving weak ad-hoc queries using wikipedia asexternal corpus. In *SIGIR*. ACM, 797–798.

[10] Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining quality phrases from massive text corpora. In *SIGMOD*. ACM, 1729–1744.

[11] Zhiyuan Liu, Xinxiong Chen, Yabin Zheng, and Maosong Sun. 2011. Automatic keyphrase extraction by bridging vocabulary gap. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. ACL, 135–144.

[12] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*.

[13] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30, 1 (2007), 3–26.

[14] Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *ICADL*. Springer, 317–326.

[15] Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *WWW*. International World Wide Web Conferences Steering Committee, 1015–1024.

[16] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1825–1837.

[17] Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018. Directional Skip-Gram: Explicitly Distinguishing Left and Right Context for Word Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. ACL, 175–180. https://doi.org/10.18653/v1/N18-2028

[18] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. 2011. Short text conceptualization using a probabilistic knowledgebase. In *Proceedings of the twenty-second international joint conference on artificial intelligence-volume volume three*. AAAI Press, 2330–2336.

[19] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 697–706.

[20] Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji-Rong Wen. 2015. Query Understanding through Knowledge-Based Conceptualization. In *IJCAI*. 3264–3270.

[21] Ian H Witten and Olena Medelyan. 2006. Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'06)*. IEEE, 296–297.

[22] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 481–492.

[23] Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. 2009. Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. ACL, 1021–1029.

[24] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*. 1753–1762.

[25] Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR*. ACM, 287–294.

A INFORMATION FOR REPRODUCIBILITY

A.1 System Implementation and Deployment

We implement and deploy our *ConcepT* system in Tencent QQ Browser. The concept mining module and taxonomy construction module are implemented in Python 2.7, and they run as offline components. For document tagging module, it is implemented in C++ and runs as an online service. We utilize MySQL for data storage.

In our system, each component works as a service and is deployed on Tencent Total Application Framework (Tencent TAF)[5]. Tencent TAF is a high-performance remote procedure call (RPC) framework based on name service and Tars protocol, it also integrate administration platform, and implement hosting-service via flexible schedule. It has been used in Tencent since 2008, and supports different programming languages. For online document concept tagging, it is running on 50 dockers. Each docker is configured with six 2.5 GHz Intel Xeon Gold 6133 CPU cores and 6 GB memory. For offline concept mining and taxonomy construction, they are running on 2 dockers with the same configuration.

```
Data: Queries and query logs in a day
Result: Concepts
Check whether successfully obtained the queries and logs;
if succeed then
    Perform concept mining by our proposed approach;
else
    Break;
end
```

Algorithm 1: Offline concept mining process.

```
Data: News documents, the vocabulary of instances, concepts,
      and the index between key terms and concepts
Result: isA relationship between concepts and instances
for each document do
    Get the instances in the document based on the
      vocabulary;
    for each instance do
        Get the intersection of concept key terms and the
          terms co-occurred in the same sentence with
          document instances;
        Get related concepts that containing at least one key
          term in the intersection;
        Get <instance, key terms, concepts> tuples based on
          the results of above steps;
    end
end
Get the co-occurrence features listed in Table 5, and classify
  whether existing isA relationship between the instances and
  candidate concepts.
```

Algorithm 2: Offline *isA* relationship discovery between concepts and instances.

```
Data: News documents, isA relationship between instances
      and concepts
Result: Documents with concept tags
for each document do
    Perform word segmentation;
    Extract key instances by the approach described in Fig. 3;
    Get candidate concepts by the isA relationship between
      concepts and key instances;
    for each concept do
        Calculate the coherence between the candidate
          concept and the document by the probabilistic
          inference-based approach;
        Tag the concept to the document if the coherence is
          above a threshold;
    end
end
```

Algorithm 3: Online probabilistic inference-based concept tagging for documents.

```
Data: News documents
Result: Documents with concept tags
for each document do
    Perform word segmentation;
    Extract key terms by TF-IDF;
    Get candidate concepts containing above key terms;
    Get the title-enriched representation of candidate
      concepts;
    Represent document and each candidate concept by
      TF-IDF vector;
    for each concept do
        Calculate cosine similarity between the candidate
          concept and the document;
        Tag the concept to the document if the similarity is
          above a threshold;
    end
end
```

Algorithm 4: Online matching-based concept tagging for documents.

Algorithm 1-4 show the running processes of each component in ConcepT. For offline concept mining from queries and search logs, the component is running on a daily basis. It extracts around 27,000 concepts from 25 millions of query logs everyday, and about 11,000 of the extracted concepts are new. For offline relationship discovery in taxonomy construction, the component runs every two weeks. For online concept tagging for documents, the processing speed is 40 documents per second. It performs concept tagging for about 96,700 documents per day, where about 35% of them can be tagged with at least one concept.

A.2 Parameter Settings and Training Process

We have described the threshold parameters in our paper. Here we introduce the features we use for different components in our system, and describe how we train each component. Table 5 lists

[5] https://github.com/TarsCloud/Tars

Table 5: The features we use for different tasks in ConcepT.

Task	Features
Document key instance extraction	Whether the topic of instance is the same with the topic of document; whether it is the same with the instance in title; whether the title contains the instance topic; the frequency of the instance among all instances in the document; the percentage of sentences containing the instance.
Classify whether a short text is a concept	Whether the short text ever shown as a user query; how many times it has been searched; Bag-of-Word representation of the text; the topic distribution of user clicked documents given that short text as query.
Train CRF for concept mining from query	word, NER, POS, <previous word, word>, <previous word, next word>, <previous POS, POS>, <POS, next POS>, <previous POS, word>, <word, next POS>.

the input features we use for different sub-modules in our *ConcepT* system.

Training process. For concept mining, we randomly sample 15,000 query search logs in Tencent QQ Browser within one month. We extract concepts for these query logs using approaches introduced in Sec. 2, and the results are manually checked by Tencent product managers. The resulting dataset is used to train the classifier in query-title alignment-based concept mining, and the Conditional Random Field in our model. We utilize CRF++ v0.58 to train our model. 80% of the dataset is used as training set, 10% as development set and the remaining 10% as test set.

For concept tagging, we randomly sample 10,000 news articles from the feeds stream of Tencent QQ browser during a three-month period, where each topic contains abut 800 to 1000 articles. We iteratively perform concept tagging for documents based on the approaches described in Sec. 3. After each iteration, we manually check whether the tagged concepts are correlated or not. Then we update our dataset and retrain the models of concept tagging. The iteration process is topped until no more new concepts can be tagged to documents. The resulting dataset is used to train the classifiers and set the hyper-parameters in concept tagging. We use 80% of the dataset as training set, 10% as development set and the remaining 10% as test set.

A.3 Publish Our Datasets

We have published our datasets for research purpose and they can be accessed from https://github.com/BangLiu/ConcepT. Specifically, we open source the following datasets:

- **The UCCM dataset**. It is used to evaluate the performance of our approach for concept mining and it contains 10, 000 samples.
- **The document tagging dataset**. It is used to evaluate the document tagging accuracy of ConcepT, and it contains 11,547 documents with concept tags.
- **Topic-concept-instance taxonomy**. It contains 1000 *topic-concept-instance* samples from our constructed taxonomy.
- **The seed concept patterns for bootstrapping-based concept mining**. It contains the seed string patterns we utilized for bootstrapping-based concept mining from queries.
- **Pre-defined topic list**. It contains our 31 pre-defined topics for taxonomy construction.

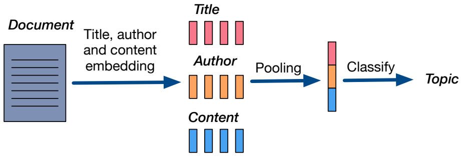


Figure 6: Document topic classification.

Table 6: Examples of queries and the extracted concepts given by ConcepT.

Query	Concept
What are the Qianjiang specialties (黔江的特产有哪些)	Qianjiang specialties (黔江特产)
Collection of noodle snacks cooking methods (面条小吃的做法大全)	noodle snacks cooking methods (面条小吃的做法)
Which cars are cheap and fuel-efficient? (有什么便宜省油的车)	cheap and fuel-efficient cars (便宜省油的车)
Jiyuan famous snacks (济源有名的小吃)	Jiyuan snacks (济源小吃)
What are the symptoms of depression? (抑郁症有什么症状)	symptoms of depression (抑郁症症状)
Large-scale games of military theme (军事题材的大型游戏)	Military games (军事游戏)

A.4 Details about Document Topic Classification

Topic classification aims to classify a document d into our predefined N_t (it is 31 in our system) topic categories, including entertainment, events, technology and so forth. Fig. 6 illustrates our model for document topic classification. We represent the title, author, and content of document d by word vectors. Then we apply max pooling to title and author embeddings, and mean pooling to content embeddings. The results of pooling operations are concatenated into a fix-length vector representation of d. We then classify it by a feed forward neural network. The accuracy of our model is 95% on a labeled dataset containing 35,000 news articles.

A.5 Examples of Queries and Extracted Concepts

Table 6 lists some examples of user queries, together with the concepts extracted by ConcepT. We can see that the concepts are appropriate to summarize the core user intention in queries.