Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

# Learning natural ordering of tags in domain-specific Q&A sites*

Junfang JIA[1], Guoqiang LI‡[2]

[1]*School of Computer and Network Engineering, Shanxi Datong University, Datong 037009, China*

[2]*School of Software, Shanghai Jiao Tong University, Shanghai 200240, China*

E-mail: jiajunfang816@163.com; li.g@sjtu.edu.cn

Received Nov. 24, 2019; Revision accepted Feb. 12, 2020; Crosschecked Aug. 19, 2020; Published online Jan. 19, 2021

**Abstract:** Tagging is a defining characteristic of Web 2.0. It allows users of social computing systems (e.g., question and answering (Q&A) sites) to use free terms to annotate content. However, is tagging really a free action? Existing work has shown that users can develop implicit consensus about what tags best describe the content in an online community. However, there has been no work studying the regularities in how users order tags during tagging. In this paper, we focus on the natural ordering of tags in domain-specific Q&A sites. We study tag sequences of millions of questions in four Q&A sites, i.e., CodeProject, SegmentFault, Biostars, and CareerCup. Our results show that users of these Q&A sites can develop implicit consensus about in which order they should assign tags to questions. We study the relationships between tags that can explain the emergence of natural ordering of tags. Our study opens the path to improve existing tag recommendation and Q&A site navigation by leveraging the natural ordering of tags.

**Key words:** Question and answering (Q&A) sites; Tagging; Natural order; Skip gram
https://doi.org/10.1631/FITEE.1900645 **CLC number:** TP182

## 1 Introduction

Recent years have witnessed the development and maturation of domain-specific question and answering (Q&A) sites. These websites have been built around focused communities in which participants share and learn knowledge in some specific domains such as CodeProject for software developers in English (http://www.codeproject.com/script/Answers/List.aspx), SegmentFault for developers in Chinese (http://segmentfault.com/), Biostars for bioinformatics (https://www.biostars.org/), and CareerCup for programming interviews (http://www.careercup.com/).

When posing questions on these Q&A sites, requesters are usually asked to tag their questions with descriptive terms. It allows users to use free terms to tag questions, as opposed to predefined, fixed terminology and rules. However, is tagging really a free action? One might expect that individuals' personal preferences, knowledge background, and information needs, compounded by an ever-increasing number of users, would yield a chaotic pattern of tags as time goes by. Contrary to this intuition, researchers have demonstrated that a certain regularity exists in user activity, tag frequencies, and choices of tags (Golder and Huberman, 2006; Halpin et al., 2007; Gummidi et al., 2019). Many statistical and data mining techniques have been proposed to detect the stability of tag usage and infer ontological or hierarchical relationships among user-created tags (Cattuto et al., 2007; Robu et al., 2009; Xie et al., 2017).

‡ Corresponding author

* Project supported by the Shanxi Datong University Project (No. 2012k6) and Shanxi Datong University Educational Reform Project (No. xjg2015202)

ORCID: Junfang JIA, https://orcid.org/0000-0002-3451-8487; Guoqiang LI, https://orcid.org/0000-0001-9005-7112

Existing work has shown that a large number of users can subconsciously develop implicit consensus about what best describes the content in an online community. However, there has been no work studying the regularities in how users assign tags to content. In this study, we are focusing on one aspect of tag-assigning, the natural ordering of tags, i.e., how users assign tags for their questions.

We refer to this ordering consensus as natural ordering of tags, as we consider tagging a natural product of human effort.

The term "natural" derives from the field of natural language processing (NLP), where corpus-based, statistical methods have been developed to automatically process texts in natural language.

An essential fact underlying these statistical methods is what people write and say is largely regular and predictable (Ponte and Croft, 1998; Xie et al., 2016).

Recently, Hindle et al. (2012) and Allamanis et al. (2014) have shown that software programs that developers write, despite being written in artificial language (like Java), also manifest natural coding convention that can be captured in statistical language models.

In this study, we hypothesize that the same argument applies to tags used to describe questions in a domain-specific Q&A site. Define the set of all tags in a Q&A site to be the vocabulary $V$ of a language. A sequence of tags assigned to a question forms a sentence in the language. We hypothesize that "tag sentences" that Q&A site users use to describe questions are mostly simple and rather repetitive, and thus they have predictable ordering that can be captured in statistical language models.

We validate our hypothesis through the following three research questions:

1. Is there any natural ordering of tags in CodeProject?

We adopt a widely used statistical language model (skip-gram) to model the corpus of tag sentences extracted from 7 840 715 questions in CodeProject. We demonstrate that the large majority of tag sentences exhibit natural ordering and that the learned language model captures the statistical regularity that exists in the corpus of tag sentences.

2. Why natural ordering of tags (if any) emerges from users' tagging activities?

After detecting the natural ordering of tags in domain-specific Q&A sites, we further compare the relative frequency of bi-grams, i.e., a sequence of two tags ($A$ and $B$) in the language model with that of the corresponding bi-grams in reverse order, i.e., ($B$ and $A$). We identify two reasons (i.e., inclusion and convention) that result in one ordering appearing much more frequently than the other. Inclusion represents a general-to-specific ordering between the two tags, such as java and arrays. Convention represents a community-favored ordering between the two tags, such as php and mysql, although there are no logic orders between the tags. We identify one reason (i.e., juxtaposition) that results in the frequencies of the two orderings being equal or very close. Juxtaposition indicates that the two tags represent the different concepts in different contexts and that personal preference plays an important role in its order. So, the community cannot collectively develop consensus about the ordering of the two tags and this leads to the wrong prediction of tag-reordering.

3. Is the natural ordering of tags (if any) domain-specific or domain-agnostic?

Finally, we comparatively study the language models for four domain-specific Q&A sites, i.e., CodeProject, SegmentFault, Biostars, and CareerCup, which serve completely different online communities, i.e., computer programming in English, computer programming in Chinese, bioinformatics, and interviews. Although the four sites serve completely different domains, the usage of tags in them all exhibits natural order. It means that the ordering of tags is domain-agnostic. However, different characteristics of Q&A sites result in the differences in the coverage and prediction accuracy of the language model and the underlying relationships in tags.

We make the following contributions in this study: (1) a statistical language model of tag sentences in domain-specific Q&A sites, (2) a foundational study on the natural ordering of tags, and (3) a proposal for future directions and approaches for tag analysis and recommendation.

## 2 Related work

In information systems, a tag is a keyword, term, or label assigned to a piece of information (e.g., a blog, question, or picture) which can briefly describe its category and content. Tagging is an important feature of Web 2.0 and this mechanism has

been widely adopted by many applications and websites such as Stack Exchange, Flickr, and Pinterest.

With the popularity of social networking, photography sharing, and bookmarking sites, a big pool of tag data has been accumulated in the web, and it is a treasure trove for mining. Many studies have been carried out to investigate crowdsourcing tags. Prior research on tagging falls into two main groups. The first group focuses on empirical studies of tagging content and tagging behaviors. For instance, research has been conducted to find out the motivation for social tagging (Körner et al., 2010; Strohmaier et al., 2010), tagging roles (Thom-Santelli et al., 2008), and the dynamics and consensus of collaborative tagging (Halpin et al., 2007; Robu et al., 2009).

The second group investigates the possible usage and functions of social tags. First, tags have been found useful in searching webs (Heymann et al., 2008; Schenkel et al., 2008), images, videos (Heckner et al., 2009), etc. In addition, social tagging is used for navigation (Storey et al., 2006; Chi and Mytkowicz, 2008). Second, tags have been widely exploited to generate taxonomy (Heymann and Garcia-Molina, 2006) or folksonomy by hierarchical clustering (Gemmell et al., 2008) and association rule mining (Schmitz et al., 2006). Finally, many researchers also work to automatically recommend tags based on different information such as text (Song et al., 2008), images (Sigurbjörnsson and van Zwol, 2008), and heterogeneous information like user behaviors (Feng and Wang, 2012).

Our study lies in the first category, but it can be easily extended for applications in the second category. In the first category, the stability of social tagging is intensively investigated ranging from tag proportions (Golder and Huberman, 2006), tag distribution (Robu et al., 2009), tag cooccurrence (Cattuto et al., 2007), to semantic level of tags (Fu et al., 2010). However, to our knowledge, no one has examined whether certain stability also exists in tagging order. We believe that user tagging behaviors are not arbitrary but obey certain implicit order, i.e., exploring the relation between user tagging behaviors and the content of tags. Some applications like navigation and tag recommendation are discussed.

## 3 Dataset overview

Stack Exchange hosts 143 Q&A sites on diverse topics, such as technology (e.g., Stack Overflow), science (e.g., Mathematics), and humanity (e.g., English language & usage). It periodically releases the data dump (https://archive.org/details/stackexchange) of its sites to the public. In this study, we use the latest data dump (released on March 8, 2015) of the three Stack Exchange sites, i.e., Stack Overflow, Mathematics (Math), and English language & usage (English). When asking a question on Stack Exchange sites, the requester is required to tag the question with one to five tags to briefly describe to which topics the question belongs (an example from Stack Overflow is shown in Fig. 1). Table 1 summarizes the three datasets.

For Stack Overflow, the data dump contains 8 978 719 questions from July 1, 2008 to March 8, 2015. We collected 39 948 unique tags from the 8 978 719 questions. As we aim to explore the order of tags, only questions with two or more tags were preserved (about 88% of all the questions, see Fig. 2). As Stack Exchange sites are collaborative Q&A sites, users with high reputation are allowed to edit others' posts to enhance clarity and quality. Among all the post edits in Stack Overflow, tags of 46 289 questions were reordered. We refer to these questions as tag-reordered (TR) questions, and


**Fig. 1 A question post in CodeProject. Tags are highlighted in the red box (References to color refer to the online version of this figure)**

**Table 1 Dataset description**

| Site | Number of non-tag-reordered questions | Number of tag-reordered questions | Number of tags |
|---|---|---|---|
| Stack Overflow | 7 840 715 | 46 289 | 39 948 |
| Math | 259 060 | 149 | 1385 |
| English | 31 448 | 176 | 910 |

other questions as non-tag-reordered (NTR) questions. For these 46 289 TR questions, we collected two sets of tag sentences from these questions, one for before tag reordering (before-reordering-corpus) and the other for after tag reordering (after-reordering-corpus). For the remaining 7 840 715 NTR questions, we collected one set of tag sentences (non-reordering-corpus). The non-reordering-corpus was used for training and evaluating the language model. We then validated the learned language model using before-reordering-corpus and after-reordering-corpus.

For Math and English, Q&A sites were launched much later than Stack Overflow (Math on July 20, 2010 and English on August 5, 2010). They have fewer questions than Stack Overflow. We collected 1385 and 910 tags from Math and English questions, respectively. There were only 149 and 176 TR questions in Math and English, respectively. In Fig. 2, we can see that a much higher percentage of questions in Math and English have only one tag, and a much lower percentage of questions in Math and English have four or five tags. In this study, we use mainly the non-reordering-corpus of Math and English to comparatively study the natural ordering of tags in the three different Q&A sites. As there are only a very small number of TR questions, we do not perform the validation of language models using the before-reordering-corpus or after-reordering-corpus of Math and English.

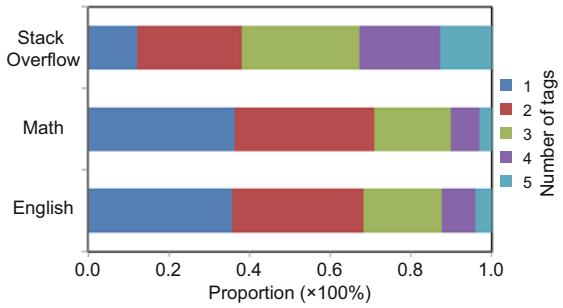


**Fig. 2 Percentage of questions with different numbers of tags in the three datasets (References to color refer to the online version of this figure)**

## 4 Is there any natural ordering of tags in Stack Overflow?

In this section, we describe the skip-gram language model trained by the corpus of tag sentences extracted from CodeProject questions. Then, we present the empirical results of whether there is natural ordering of tags in the corpus of tag sentences.

### 4.1 Background: skip-gram language model

A statistical language model assigns a probability to a sequence of words by means of a probability distribution. In this work, words are tags and the sequence is the order of tags assigned to each question in CodeProject. Thus, given a sequence $s$ of tags, $\{t_1, t_2, \ldots, t_n\}$ $(n \leq 5)$, the probability of sequence $s$ can be estimated based on the product of a series of conditional probabilities:

$$p(s) = p(t_1)p(t_2|t_1)p(t_3|t_1t_2)\ldots p(t_n|t_1t_2\ldots t_{n-1}). \tag{1}$$

The traditional $N$-gram model assumes a Markov property; i.e., the word occurrence probability at the $i^{\text{th}}$ position is influenced by only the $(N-1)$ words which precede the word under consideration:

$$p(t_i|t_1t_2\ldots t_{i-1}) \simeq p(t_i|t_{i-n+1}t_{i-n+2}\ldots t_{i-1}).$$

In our study, as the length of the longest sequence is only five, we set $N = 2$; i.e., we use a bi-gram language model. The probability of the sequence in Eq. (1) in a bi-gram model can be written as

$$p(s) = p(t_1)p(t_2|t_1)p(t_3|t_2)\ldots p(t_n|t_{n-1}).$$

For a sentence, a bi-gram model enumerates the subsequences of all the consecutive words (i.e., one tag followed by the tag next to it). Recall the example in Fig. 1; the question has four tags: ASP.NET, Javascript, jQuery, JSON.

The bi-gram model is {ASP.NET Javascript, Javascript jQuery, jQuery JSON}.

However, the assumption of the $N$-gram model is often invalid. For example, according to the assumption of the $N$-gram model, $p$(JSON|ASP.NET Javascript jQuery) $\simeq$ $p$(JSON|jQuery). However, in this question, the tags ASP.NET and Javascript should also contribute to the occurrence probability of JSON, because JSON here is used as a file format between front-end Javascript and back-end ASP.NET. To leverage such non-continuous context in a tag sequence, we adopt the $K$-skip bi-gram model.

$K$-skip $N$-gram (Rosenfeld, 1994; Siu and Ostendorf, 2000; Goodman, 2001; Guthrie et al., 2006)

provides an effective way to solve the data sparsity problem in the traditional $N$-gram model. In the skip-gram model, words do not need to be consecutive in the text, but can have gaps between them. The $K$-skip $N$-gram model "skips over" the gap (up to $K$ words) to construct an $N$-gram. Using the example in Fig. 1, we can construct the following $K$-skip bi-gram model:

0-skip-bi-gram = {ASP.NET Javascript, Javascript jQuery, jQuery JSON} (i.e., the bi-gram model).

1-skip-bi-gram = {ASP.NET Javascript, ASP.NET jQuery, Javascript jQuery, Javascript JSON, jQuery JSON}.

2-skip-bi-gram = {ASP.NET Javascript, ASP.NET jQuery, ASP.NET JSON, Javascript jQuery, Javascript JSON, jQuery JSON}.

As this question has only four tags, the largest skip is 2-skip. As a question can have at most 10 tags, we can have up to 8-skip bi-gram for a sequence of 10 tags.

Now, we can use the $K$-skip bi-gram to approximate the conditional probability as

$$p(t_n|t_1t_2\dots t_{n-1}) = \lambda_1 p(t_n|t_{n-k-1}) + \lambda_2 p(t_n|t_{n-k}) + \dots + (1-\lambda_1-\lambda_2-\dots-\lambda_k)p(t_n|t_1), \quad (2)$$

where $\lambda_i$ ($i = 1, 2, \dots, k$) are the weights satisfying $0 \le \lambda_i \le 1$. Substituting Eq. (2) into Eq. (1), we can approximate the probability of sequence $p(s)$. Parameters $\lambda_i$'s are obtained by enumerating a possible value in the range [0, 1] until the best prediction rate is achieved. However, in practice, this estimator may not work well. If a sequence $\{t_1, t_2, \dots, t_n\}$ may not occur in the training corpus, then $p(t_n|t_1t_2\dots t_{n-1}) = 0$ will lead to $p(s) = 0$. Smoothing (Chen and Goodman, 1996) is a technique for handling such cases while still producing usable results with sufficient statistical rigor. In our work, we adopt Katz smoothing.

### 4.2 Method

Each question in CodeProject is tagged with a sequence of tags $\{t_1, t_2, ..., t_n\}$ ($n \le 10$). We refer to this sequence of tags as a tag sentence. When tagging a question, is there any natural ordering that the user may implicitly follow to "speak" the tag sentence? Using the corpus of tag sequences, we train a $K$-skip bi-gram model and perform 10-fold cross-validation of the model for tag sequence prediction. As no models can predict a totally random distribution, if the prediction of a tag sequence matches the original tag sentences in most cases, we can infer that some natural ordering of tags exists.

Given a $K$-skip bi-gram model, Model, learned from a corpus of tag sentences and a set of tags $\{t_1, t_2, \dots, t_n\}$ ($n \le 10$), Algorithm 1 first enumerates all the possible sequences of the given set of tags and then selects a sequence (i.e., a tag sentence) with the highest probability according to Model.

**Algorithm 1** Prediction of the sequence of tags

**Input:** $K$-skip bi-gram model Model and a set of tags $\{t_1, t_2, \dots, t_n\}$

**Output:** a predicted order bestOrder of the set of tags $\{t_1, t_2, \dots, t_n\}$

```
Initialize bestProb ← 0, prob ← 0
Initialize a list
    allOrder ← GenerateAllOrder({t_1, t_2, ..., t_n})
for order ∈ allOrder do
    prob ← Model(order)
    if prob > bestProb then
        bestProb ← prob
        bestOrder ← order
    end if
end for
```

### 4.3 Results

We first report the 10-fold cross-validation on the tag sentences to evaluate the coverage and prediction accuracy of the proposed $K$-skip bi-gram model. Then we zoom into the 4-skip bi-gram model (the best one) to further analyze the prediction results.

#### 4.3.1 Cross-validation

There are 156 282 tag sentences in our dataset and we randomly split them into 10 equally-sized sets. In each fold of cross-validation, one set is chosen as test data and the other nine sets are used as training data. That is, 90% of the corpus is used to train the model, and 10% is used to test the model. As the sequence of tags contains at most 10 tags, the $K$-skip ranges from 0 to 8. For each $K$-skip, we perform the 10-fold cross-validation to measure the coverage and prediction accuracy of the tag sequence. To test a learned model, we enter the tag set of each tag sequence in the test data as input

to Algorithm 1, and compare the predicted sequence with the original tag sequence.

Coverage (Rosenfeld, 1995; Abate et al., 2010) is an important metric for testing the quality of a language model in NLP. Given a $K$-skip bi-gram model learned from the training data, the more bi-grams in the test data the model covers, the higher the coverage. Fig. 3a shows the average coverage of the 10-fold cross-validation of different $K$-skip bi-gram models. We can see that the 0-skip bi-gram model (i.e., bi-gram) can cover about 70% of the bi-grams in the test data. Unsurprisingly, a 3-skip (or more) bi-gram model has higher coverage (about 74%) of the bi-grams in the test data.

Then we check the prediction accuracy of the tag sequence by measuring the Levenshtein distance (Levenshtein, 1966) between the original tag sequence and the predicted tag sequence. Note that we do not use perplexity (Bird et al., 1997), because it is only an overall likelihood estimation, while the Levenshtein distance can show how much difference there is between prediction results and ground truth. The Levenshtein distance is commonly used to measure the difference between two sequences in information theory and computer science. In our work, the Levenshtein distance between the two tag sequences is the minimum number of single-tag edits (i.e., insertions or deletions) required to change one sequence into the other. The Levenshtein distance is 0 if the two sequences match. We consider it an accurate prediction. For the example in Fig. 1, the set of tags from the tag sentence is {ASP.NET Javascript jQuery JSON}. If the predicted sequence of this set of tags is {ASP.NET Javascript JSON jQuery}, the Levenshtein distance between the original sequence and the predicted sequence is two. That is, to change the predicted sequence to the original sequence, one needs to delete JSON from the predicted sequence and insert JSON after jQuery.

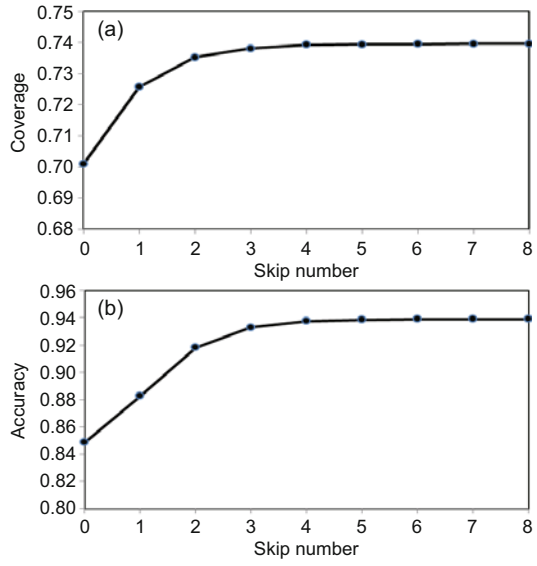


**Fig. 3 Coverage (a) and accuracy (Levenshtein distance=0) (b) with different skip numbers**

Fig. 3b shows the average prediction accuracy of the 10-fold cross-validation of different $K$-skip bi-gram models. We can see that as $K$ increases, the prediction accuracy (Levenshtein distance=0) of the $K$-skip bi-gram model increases from 0.85 to 0.94. That is, when the $K$-skip is more than three, 94% of the predicted tag sequences are exactly the same as the original sequence. Thus, we conclude that natural ordering exists in tag sentences in Stack Overflow.

#### 4.3.2 Prediction results

Despite the promising performance of our language model, we still wonder why there are mistakes in our prediction. Thus, we repeat one iteration of cross-validation in the last part to further analyze the results. This iteration is called the examination process in our study and will be used again in the following sections. We randomly select 90% of tag sentences to train a 4-skip bi-gram model (which has the highest coverage and accuracy), and the remaining 10% of tag sequences are exploited for testing. Then, the tag set of each tag sequence in the test set is used as the input in Algorithm 1, and the predicted sequence is compared with the corresponding tag sequence in the ground truth.

Results show that the learned 4-skip bi-gram model can cover 74% of bi-grams in the test data. The overall prediction accuracy of the tag sequence is 93.6%. Fig. 4 also shows the Levenshtein distance between the predicted tag sequences and the community-edited tag sequences for different-length tag sequences. We can see that our model is more accurate when the tag sequence is short (two or three tags). However, when the tag sequence contains four or five tags, the prediction accuracy becomes worse. This is because the more tags a tag sequence has, the more possible orders there are. However, note that when the sequence length is four or five, the Levenshtein distance of 12% and 26% tag sequences is two (i.e., red pie). This indicates that only two

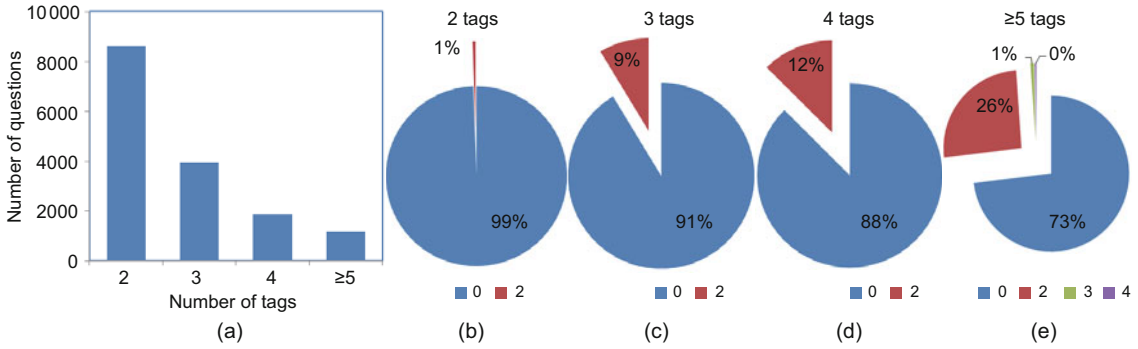


**Fig. 4 Question distribution with different numbers of tags (a), and Levenshtein distance distribution under two (b), three (c), four (d), or no less than five (e) tags. Different colors represent different Levenshtein distances. References to color refer to the online version of this figure**

tags mismatch between the predicted sequence and the community-edited sequence. In fact, as we will show in the next section, most of these mismatches are caused by juxtaposed tags, such as "C# Windows" and "C++ C#." Such juxtaposed tags can be ordered in either way.

## 5 Why can natural ordering of tags emerge from users' tagging activities?

We have shown that there exists natural ordering of tags that the CodeProject users implicitly follow when tagging questions. In this section, we attempt to identify the underlying factors that result in such natural ordering of tags by examining the relationships of tags in all the bi-grams in the 3-skip bi-gram model.

### 5.1 Relationships in bi-grams

The 3-skip bi-gram model trained on all data contains 51 055 unique bi-grams. Among these bi-grams, 3909 bi-grams occur in more than 10 tag sequences. The occurrences of these 3909 bi-grams account for 80.1% of all bi-grams occurrences. Our analysis focuses on these 3909 frequently occurring bi-grams.

Among the 3909 bi-grams, 3697 bi-grams do not have the corresponding order-reversed bi-grams. That is, tags in these 3697 bi-grams appear only in one specific order, not the other way around. For example, tags C# and C#4.0 appear only as C# C#4.0 (i.e., C# precedes C#4.0), but never C#4.0 C#. We refer to these bi-grams as unique-ordering bi-grams.

All the remaining 212 bi-grams have the corresponding order-reversed bi-grams. Note that the order-reversed bi-grams may or may not occur in more than 10 tag sentences. For each pair of bi-gram and its corresponding order-reversed bi-gram, we compute a frequency quotient $q$ by dividing the frequency of the less frequently occurring bi-gram by the frequency of the more frequently occurring bi-gram. For those unique-ordering bi-grams, set the occurrence frequency of their order-reversed bi-grams as one so that the quotient is not 0. As such, the frequency quotient is $0 < q \leq 1$. Fig. 5 shows the distribution of frequency quotient of the 3909 frequently occurring bi-grams. We can see that the quotient of 96.5% of the bi-grams is lower than 0.1. That is, the occurrence frequency of a bi-gram is 10 times higher than that of the corresponding order-reversed bi-gram.

Thus, we use 0.1 as a cutoff threshold. We consider the frequency quotient $q \leq 0.1$ as low quotient,

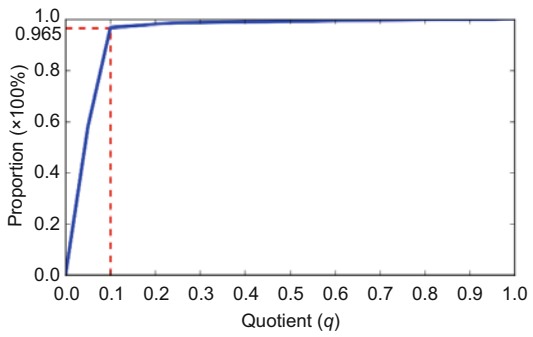


**Fig. 5 Cumulative distribution function of frequency quotient of bi-grams, where the vertical axis represents the proportion of the bi-grams taking on a quotient less than or equal to $q$. The dashed line shows that quotient of 96.5% of bi-grams is lower than 0.1**

and $q > 0.1$ as high quotient. We randomly select 400 low-quotient bi-grams and 100 high-quotient bi-grams to analyze the relationships of the tags in the bi-grams. Table 2 presents 10 lowest-quotient bi-grams and 10 highest-quotient bi-grams in these selected bi-grams.

We manually examine these selected bi-grams and identify three categories of relationships (inclusion, convention, and juxtaposition). We categorize samples based on our own knowledge and the information on Wikipedia. Inclusion means a natural general-specific relationship between the tags, such as "Javascript jQuery" in which jQuery is a Javascript library. Convention means that the two tags have no realistic relationships but the community favors a specific order, such as "ASP.NET SQL-Server," where ASP.NET is a programming language while SQL-Server is a relational database. Juxtaposition means that the two tags represent different concepts without forming consensus order, such as "Visual-Studio VB" (one is the integrated development environment (IDE) while the other is a programming language).

In Fig. 6, we can see that most of the low-quotient bi-grams (51%) have a convention relationship, some (37%) have an inclusion relationship, and only a very small percentage (12%) have a juxtaposition relationship. In contrast, most high-quotient bi-grams (64%) have a juxtaposition relationship, while the rest have inclusion (18%) and convention (18%) relationships. Clearly, inclusion and convention usually result in an ordering of tags appearing much more frequently than the reversed one; e.g., users always align their tags in a certain order based on their communication behaviors. In contrast, for juxtaposition, the order of the two tags in a bi-gram is often arbitrary, resulting in the same or similar occurrence frequencies between the two alternative orderings.

As low-quotient bi-grams account for 96.5% of all bi-grams, the presence of inclusion and convention explains why natural ordering of tags can emerge from user tagging behavior.

### 5.2 Impact of juxtaposition

As the order of the two juxtaposed tags is rather arbitrary, juxtaposition intuitively would negatively affect the prediction of tag sequences. To confirm this intuition, we examine the wrongly predicted tag sequences in the study of the examination process. We report in the examination process that the overall prediction accuracy is about 93.6% (i.e., the Levenshtein distance between the predicted tag sequence and the community-edited tag sequence is 0). Among the 6.4% wrongly predicted tag sequences, 81.9% (i.e., 883) mismatch only a pair of tags when compared with the community-edited tag sequences (i.e., Levenshtein distance=2). For example, the community-edited sequence is {C#,

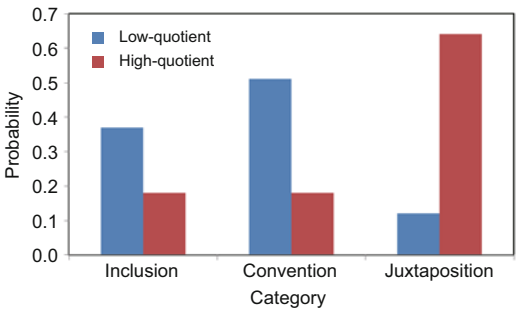


**Fig. 6 Proportion of different categories of relationships of the sampled 400 low-quotient bi-grams and 100 high-quotient bi-grams (References to color refer to the online version of this figure)**

**Table 2 Examples of pairs of 10 lowest- and highest-quotient bi-grams**

| Low-quotient pairs | | | High-quotient pairs | | |
|---|---|---|---|---|---|
| Pair | Quotient | Category | Pair | Quotient | Category |
| VB, VB.NET | $4.53\times10^{-5}$ | Inclusion | .NET, XAML | 0.696 97 | Juxtaposition |
| C#, C#4.0 | $7.70\times10^{-5}$ | Inclusion | WPF, Visual-Studio | 0.754 39 | Inclusion |
| ASP.NET, Javascript | $2.56\times10^{-4}$ | Convention | VBScript, VB | 0.756 10 | Inclusion |
| Mobile, Android | $2.76\times10^{-4}$ | Inclusion | .NET, Ajax | 0.758 62 | Juxtaposition |
| ASP.NET, SQL-Server | $3.34\times10^{-4}$ | Convention | ASP, VB | 0.812 50 | Juxtaposition |
| ASP.NET, jQuery | $4.07\times10^{-4}$ | Convention | .NET, LINQ | 0.888 89 | Inclusion |
| SQL-Server, SQL-Server-2008 | $4.13\times10^{-4}$ | Inclusion | VB, IIS | 0.920 00 | Convention |
| Javascript, jQuery | $4.19\times10^{-4}$ | Inclusion | SQL-Server, VB | 0.941 33 | Juxtaposition |
| C#, VB.NET | $4.28\times10^{-4}$ | Convention | VB, .NET | 0.977 73 | Juxtaposition |
| C#, WinForm | $4.28\times10^{-4}$ | Inclusion | Visual-Studio, VB | 0.997 33 | Juxtaposition |

ASP.NET, WPF}, while the predicted sequence is {C#, WPF, ASP.NET}. The order of ASP.NET and WPF is reversed.

For these 883 wrongly predicted tag sequences, we extract 188 pairs of wrongly predicted tags like "ASP.NET, WPF" in the above example. Although only 91 wrongly predicted tags are larger than 0.1 (red dash line in Fig. 7a), their occurrences take up 78.9% of wrongly predicted tag sequences. That is, the high-quotient bi-grams result in the wrong prediction of our model. As discussed above, tags in the high-quotient bi-grams are usually juxtaposed. We further check the category of relationships of these 91 wrongly predicted bi-grams, and find that tags in 79% of these wrongly predicted bi-grams are juxtaposed. This indicates that most of our prediction errors are indeed caused by such juxtaposed bi-grams.

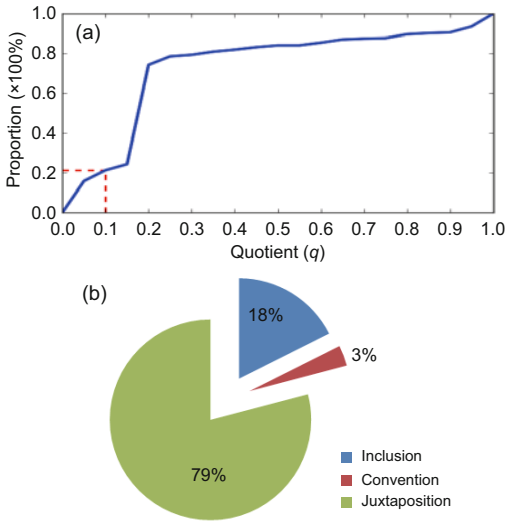


**Fig. 7 Distribution of the frequency quotients of wrongly predicted tag pairs with the vertical axis being the proportion of wrongly predicted tag pairs with a quotient less than or equal to $q$ (a) and the category of relationships of the top 100 wrongly predicted tag pairs (b) (References to color refer to the online version of this figure)**

### 5.3 Visualizing the language model

To help identify and understand the relationships of tags, we use the word tree (Wattenberg and Viégas, 2008) to visualize the learned language model. Here, we use the 4-skip bi-gram language model trained using all the data. Given a root tag $t_0$, the language model will generate many tag candidates as the following tag $t_1$, ordered by probability according to Eq. (2). Set the threshold $P$ so that only tag candidates whose probability is larger than $P$ will be selected as child nodes of $t_0$. We can obtain the whole word tree by recursively expanding the nodes in the tree using the language model.

Fig. 8 shows a partial word tree rooted at the tag C#. Given C#, the top three following tags with the highest probabilities are ASP.NET, Visual-Studio, and LINQ (we show only the top two or three tags with the highest probabilities in the word tree due to space limitation). Then, given "C# ASP.NET," the top two following tags are Javascript and SQL-Server. Given "C# ASP.NET Javascript," the top two following tags are jQuery and Ajax; neither has following tags. Similarly, we can obtain other branches of the word tree.

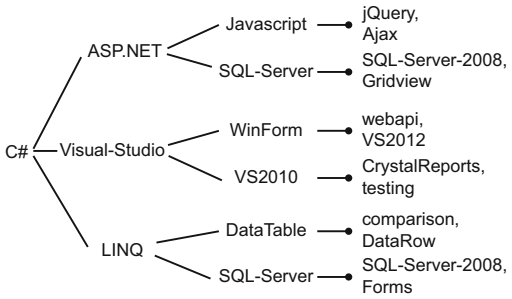


**Fig. 8 A partial word tree rooted at "C#"**

From this tree, we can observe the inclusion relationships between tags. For example, VS2010 is a widely used version of the IDE Visual-Studio, and jQuery is a Javascript library for front-end development. We can also observe the convention relationships. For example, CodeProject users usually put C# before ASP.NET, and put testing behind VS2010. The word tree not only shows us the tagging habit of developers but also displays the potential to develop a brand new navigation pattern, i.e., navigating users in the sequence from the root node to leaf nodes through the tree in Q&A sites. This will be further discussed in Section 7.1.

## 6 Is the natural ordering of tags domain-specific or domain-agnostic?

We have shown that CodeProject tags have natural ordering when they are assigned to questions. CodeProject is a Q&A site for computer programming and all questions are proposed in English. Will

such natural ordering of tags also be present in other Q&A sites serving different domains such as science or in other languages like Chinese? To deepen our understanding of ordering of tags, we apply the analysis method discussed in Sections 7.1 and 7.2 to the three other Q&A sites, i.e., CareerCup, Biostars, and SegmentFault. We use these three Q&A sites as the representatives of interviews, bioinformatics, and Chinese programming focused community.

### 6.1 *K*-skip bi-gram model and results

We perform the same 10-fold cross-validation of the $K$-skip bi-gram language model on the tag sentences extracted from CareerCup (interviews), Biostars (bioinformatics), and SegmentFault (Chinese programming) questions. Similar to CodeProject study, we test different $K$-skip. We use the prediction accuracy metric to evaluate the learned language model and compare the results with those of CodeProject.

Fig. 9 shows the prediction accuracy of the four datasets. Overall, the prediction accuracy increases as $K$ increases. We can see that the prediction accuracy of CareerCup is the highest, 0.98 in the 3-skip bi-gram language model, while the accuracy of SegmentFault is the lowest, only 0.73. Compared with the other three datasets, the accuracy differences at different $K$-skip are broader in Biostars. As shown in Fig. 2, only less than 40% questions in CodeProject, CareerCup, and SegmentFault have three or more tags, while 61.9% questions in Biostars have three or more tags. Thus, the increase of $K$ has larger impact on the prediction accuracy of Biostars.

This comparison illustrates our discovery in two aspects. On one hand, order exists in tags of the other three Q&A sites (CareerCup, Biostars, and SegmentFault). On the other hand, our language model is a feasible way to capture such implicit ordering.

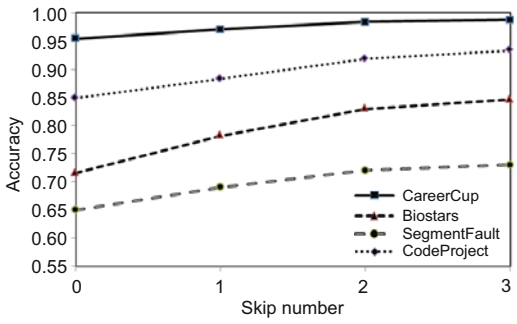


**Fig. 9 Accuracy (Levenshtein distance=0) under different skip numbers**

### 6.2 Relationships in bi-grams

Next, we examine the relationships of tags in bi-grams in the language model of CareerCup, Biostars, and SegmentFault. In the CodeProject study, we collect all the bi-grams that appear in more than 10 tag sequences, which account for about 80% of all bi-gram occurrences. Here, we collect all the bi-grams that appear in more than seven tag sequences in CareerCup and all the bi-grams that appear in more than one tag sequence in Biostars and SegmentFault. Such bi-grams also account for about 80% of all bi-gram occurrences in Math and English. We then compute the frequency quotient of these frequently occurring bi-grams in the four datasets.

Fig. 10 shows the frequency of quotient of frequently occurring bi-grams in the four Q&A sites. In CodeProject and CareerCup, the frequency quotient of 96.7% and 88.3% bi-grams is below 0.1, while the frequency quotient of only 19.5% and 3.1% bi-grams is below 0.1 in Biostars and SegmentFault, respectively. It indicates that English programming and interview tagging concentrate on a small limited set of frequent bi-grams while bioinformatics and Chinese programming tagging are far more diverse and sparse; i.e., no obvious consensus of order of bi-grams is obtained. Due to the sparsity of vocabulary, the prediction accuracy of our language is not as high as those of CodeProject (93%) and CareerCup (98%).

Table 3 lists the top-10 lowest-quotient bi-grams in CareerCup and Biostars. According to our general

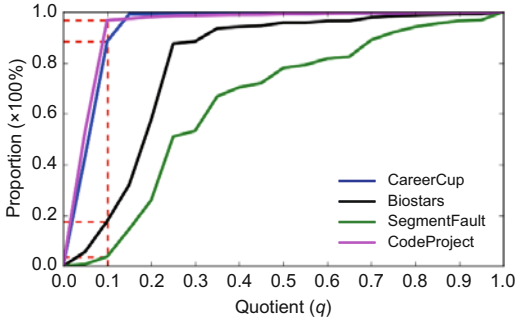


**Fig. 10 Distribution of frequency quotients of bi-grams. The vertical axis represents the proportion of a tag sequence with a quotient less than or equal to $q$. Biostars and SegmentFault are rather different from CodeProject and CareerCup (References to color refer to the online version of this figure)**

**Table 3 Top-10 lowest-quotient bi-grams in CareerCup and Biostars**

| CareerCup | | Biostars | |
|---|---|---|---|
| Pair | Quotient | Pair | Quotient |
| Software engineer/developer, algorithm | $3.59\times10^{-4}$ | genome, sequencing | $4.52\times10^{-3}$ |
| Amazon, software engineer/developer | $4.01\times10^{-4}$ | next-gen, rna-seq | $4.61\times10^{-3}$ |
| Amazon, algorithm | $6.51\times10^{-4}$ | next-gen, snp | $7.04\times10^{-3}$ |
| Microsoft, software engineer/developer | $1.07\times10^{-3}$ | genome, alignment | $7.14\times10^{-3}$ |
| Microsoft, algorithm | $1.41\times10^{-3}$ | genome, snp | $7.30\times10^{-3}$ |
| Google, software engineer/developer | $1.47\times10^{-3}$ | sequence, blast | $9.26\times10^{-3}$ |
| Software engineer/developer, coding | $1.76\times10^{-3}$ | tophat, rna-seq | $1.11\times10^{-2}$ |
| Google, algorithm | $1.93\times10^{-3}$ | cufflinks, rna-seq | $1.14\times10^{-2}$ |
| Software engineer/developer, data structures | $2.61\times10^{-3}$ | vcf, snp | $1.19\times10^{-2}$ |
| Microsoft, software engineer in test | $2.64\times10^{-3}$ | samtools, mpileup | $1.37\times10^{-2}$ |

observation, there are still three relationships in tags of bi-grams in CareerCup, Biostars, and SegmentFault. Most bi-grams with low-quotient (i.e., $\leq 0.1$) have the inclusion and convention relationships in SegmentFault similar to CodeProject. However, for CareerCup and Biostars, the low-quotient bi-grams with convention relationships are much more than that with inclusion relationships. For instance, in CareerCup, the company name is always put before the job title and detailed technique. Bi-grams in Biostars are consistent with human communication behaviors in bioinformatics such as {genome, sequenceing} and {genome, alignment}. Note that we do not further classify the bi-grams in CareerCup, Biostars, and SegmentFault into different categories as we do for CodeProject, because our goal is to check whether natural ordering of tags is domain-specific or domain-agnostic. It will be our future work to further explore it.

To provide readers with a direct sense of our language models in Math and English, we visualize part of our language models with the same procedure as for the Stack Overflow word tree. Word trees rooted at "calculus" in Math and at "grammar" in English are shown in Fig. 11. We can see that both limits and real-analysis are important components of calculus, and past-perfect and past-tense are included under tense. These tags follow the hierarchical structure in our knowledge system. However, as for verbs and nouns, and verbs and prepositions, no obvious relationship (order) can be obtained. Again, judging the implicit order of these tags will require more linguistic knowledge. Furthermore, the dataset of Math and English is much smaller than that of Stack Overflow. As a result, we do not observe as many conventions as we observe in Stack Overflow.

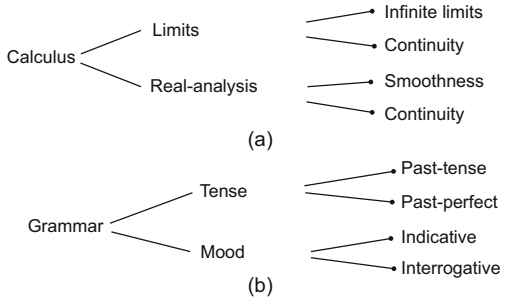


**Fig. 11 Partial word trees rooted at "calculus" in Math (a) and "grammar" in English (b)**

### 6.3 Comparison between CodeProject and SegmentFault

CodeProject and SegmentFault are both about programming, but one is in English while the other in Chinese. It is necessary to compare the tagging behaviors in these two sites. It is very likely to reveal some commonalities and difference between the different cultures in terms of programming.

From Fig. 9, the accuracies of our language model in CodeProject and SegmentFault are 93% and 73%, respectively. The reason for the difference is that the vocabulary in SegmentFault is rather sparse (Fig. 10). As they are both Q&A sites about programming but in different languages, we further check the bi-gram differences in which tags are both English. Ten bi-grams with reversed order in these two datasets are listed in Table 4. Nine of them have convention relationships but the convention order is opposite in the East and West. For instance, Chinese developers would like to put javascript after html, php, and css, while it is just the opposite for English-speaking developers. In addition, these bi-grams seem to already reach consensus in CodeProject as their quotients are rather low ($<0.1$). In

**Table 4 Ten bi-grams with reversed order in SegmentFault and CodeProject**

| SegmentFault | | CodeProject | |
|---|---|---|---|
| Pair | Quotient | Pair | Quotient |
| ios, objective-c | 0.31 | objective-c, ios | $1.6\times10^{-2}$ |
| html, css | 0.78 | css, html | $1.0\times10^{-3}$ |
| html, javascript | 0.99 | javascript, html | $5.9\times10^{-4}$ |
| php, javascript | 0.48 | javascript, php | $1.6\times10^{-3}$ |
| css, javascript | 0.78 | javascript, css | $1.7\times10^{-3}$ |
| html5, javascript | 0.97 | javascript, html5 | $4.3\times10^{-3}$ |
| django, python | 0.73 | python, django | $7.7\times10^{-2}$ |
| c, c++ | 0.48 | c++, c | $3.4\times10^{-2}$ |
| ios, iphone | 0.19 | iphone, ios | $1.8\times10^{-2}$ |
| php, linux | 0.33 | linux, php | $3.8\times10^{-2}$ |

contrast, the quotients of these bi-grams are relatively high (>0.1) in SegmentFault. This is indicative of the controversy inside the community.

Thus, it seems that developers in the West easily reach consensus in tagging order in programming, while much more diversity exists in Chinese developers. Note that the reason may also be the smaller amount of data from SegmentFault, which is about one fifth that of CodeProject. At the same time, CodeProject establishes a policy to support experts to edit the tagging of others, which may also have a certain positive influence on tagging order. This needs further exploration.

## 7 Discussion

The prediction results yielded by our $K$-skip bigram model provide convincing evidence that natural ordering of tags exists in user tagging of even different discipline Q&A sites. Our further exploration of bi-grams in our language model implies that the inclusion and convention relationships between tags contribute most to our language model. In this section, we review not only the practical implications of our study for Q&A site designers and users, but also the theoretical implications for other researchers' work.

### 7.1 Practical implications

Implications of our work for Q&A site designers are two-fold. First, the social order existing in social tagging should be taken into consideration when designing back-end algorithms to assist experts in editing. As our model has demonstrated the feasibility of reordering users' tag sequences automatically in our experiments, it will ease the burden on experts so that they can concentrate on editing other content. One simple practical tag-reordering system based on our language model has been implemented in https://tagreorder.appspot.com/ and one can try some examples. Second, such natural ordered tags can be exploited to navigate in millions of questions. The traditional navigation system always heavily depends on content categorization and hierarchical structure (Zubiaga, 2012) regardless of human behaviors, while our demonstration of tagging order opens the new path to navigation. By imitating user tagging behaviors discussed above, site administrators can design their own navigation according to tagging sequences. For instance, given the question tagging with {java android eclipse google-maps}, users can spot this question by navigation from java to google-maps according to tagging order, though these tags do not satisfy the hierarchical sequence which is widely adopted by traditional navigation. Such navigation examples can also be seen in the above mentioned word trees (Figs. 8 and 11), and the natural tagging order can act as navigational cues that facilitate the exploratory search of information.

For users, our findings have potential in the auto-completion user tagging of these posts. A tag is a word or phrase that describes the topic of a question, and suitable tags can help experts more easily spot their questions. Thus, to improve the quality of tagging, many methods have been proposed to recommend tags using multi-label classification (Xia et al., 2013) and topic modeling (Tuarob et al., 2013; Wang et al., 2014) algorithms based on sole post text. Then more features like tag co-occurrences (Belém et al., 2011) and heterogeneous information (Feng and Wang, 2012) such as users' profiles, social networks, and tag semantics are also incorporated for accuracy enhancement. However, to our knowledge, no one takes the tag order into account. Traditional machine learning methods will always return several candidate tags with different possibilities. For this multi-label classification, given the most possible candidates, our language model can predict the most possible subsequent ones, which narrows down the candidates' scope. Then they can recursively run the classification algorithm in this smaller set of candidates, and this is very likely to improve the prediction accuracy.

### 7.2 Theoretical implications

The existence of natural order among tags also indicates the stability of social tagging in Q&A sites. Thus, our work matches previous work in tagging theory and also provides new insights for researchers for further investigation.

Cattuto et al. (2007) discovered that co-occurring tags always exhibit hierarchical structures that mirror shared structures that are "anarchically negotiated" by the users. This finding is exactly in line with our experimental results that the inclusion relationship between tags plays an important role in predicting tagging order. Not only justifying their conclusion, our analysis further discovered that the convention relationship such as {PHP, .NET} contributes to the co-occurrence and their order. This pattern in social tags is similar to how words are naturally used in human communication.

Beyond word-level analysis, some works (Fu et al., 2010; Wagner et al., 2014) adopt an imitation model to further study the semantic stability of social tagging; i.e., users who can see tags created by others tend to create tags that are semantically similar to these existing tags. Thus, can imitation theory account for the tagging order? Do users try to align their tags in an order similar to that of other users? Does the tag order reflect shared background knowledge or common human memory-structure of different users? Indeed, many innovative ideas were generated by the sudden realization that some cognitive or psychological theory can explain the nature of tagging order. Although it seems that we have only started to harness the potential of social tagging order, we believe this finding is a significant step forward in the understanding of social tagging behaviors.

## 8 Conclusions and future work

Using the $K$-skip bi-gram model, we have shown that tag sequences of large collections of questions in domain-specific Q&A sites have implicit natural ordering. The $K$-skip bi-gram can capture this statistical regularity that exists in the corpus of tag sequences. We have identified three categories of relationship between tags: inclusion, convention, and juxtaposition. Our analysis showed that inclusion and convention usually result in one ordering of tags being preferred by the community over alternative ordering, while juxtaposition usually results in an arbitrary ordering of tags. Juxtaposition has a negative impact on modeling the ordering regularity and is the main reason for wrongly predicted tag sequences. Our study also showed that the language model of tag sequences can automatically reorder tags in a way that accurately matches the tag-reordering actions by high-reputation users in the community. Finally, we have shown that natural ordering of tags is largely domain-agnostic, but different Q&A sites may exhibit different ordering characteristics because of the nature of the domain. In the future, we are interested in developing an automatic tag reordering tool for Q&A sites and novel tag recommendations (e.g., recommending the following tag candidates, recommending analogical tags) based on the language model of tag sequences.

### Contributors

Junfang JIA implemented the system, processed the data, carried out the experiments, and drafted the paper. Guoqiang LI supervised the research, found the suitable datasets, and revised and finalized the paper.

### Compliance with ethics guidelines

Junfang JIA and Guoqiang LI declare that they have no conflict of interest.

### References

Abate ST, Besacier L, Seng S, 2010. Boosting $N$-gram coverage for unsegmented languages using multiple text segmentation approach. Proc 1st Workshop on South and Southeast Asian Natural Language, p.1-7.

Allamanis M, Barr ET, Bird C, et al., 2014. Learning natural coding conventions. Proc 22nd ACM SIGSOFT Int Symp on Foundations of Software Engineering, p.281-293. https://doi.org/10.1145/2635868.2635883

Belém F, Martins E, Pontes T, et al., 2011. Associative tag recommendation exploiting multiple textual features. Proc 34th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.1033-1042. https://doi.org/10.1145/2009916.2010053

Bird S, Boguraev B, Kay M, et al., 1997. Survey of the State of the Art in Human Language Technology. Cambridge University Press, USA.

Cattuto C, Loreto V, Pietronero L, 2007. Semiotic dynamics and collaborative tagging. *PNAS*, 104(5):1461-1464. https://doi.org/10.1073/pnas.0610487104

Chen SF, Goodman J, 1996. An empirical study of smoothing techniques for language modeling. Proc 34th Annual Meeting on Association for Computational Linguistics, p.310-318. https://doi.org/10.3115/981863.981904

Chi EH, Mytkowicz T, 2008. Understanding the efficiency of social tagging systems using information theory. Proc

19th ACM Conf on Hypertext and Hypermedia, p.81-88. https://doi.org/10.1145/1379092.1379110

Feng W, Wang JY, 2012. Incorporating heterogeneous information for personalized tag recommendation in social tagging systems. Proc 18th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.1276-1284. https://doi.org/10.1145/2339530.2339729

Fu WT, Kannampallil T, Kang RG, et al., 2010. Semantic imitation in social tagging. *ACM Trans Comput-Human Interact*, Article 12. https://doi.org/10.1145/1806923.1806926

Gemmell J, Shepitsen A, Mobasher B, et al., 2008. Personalizing navigation in folksonomies using hierarchical tag clustering. Proc 10th Int Conf on Data Warehousing and Knowledge, p.196-205. https://doi.org/10.1007/978-3-540-85836-2_19

Golder SA, Huberman BA, 2006. Usage patterns of collaborative tagging systems. *J Inform Sci*, 32(2):198-208. https://doi.org/10.1177/0165551506062337

Goodman JT, 2001. A bit of progress in language modeling. *Comput Speech Lang*, 15(4):403-434. https://doi.org/10.1006/csla.2001.0174

Gummidi SRB, Xie XK, Pedersen TB, 2019. A survey of spatial crowdsourcing. *ACM Trans Database Syst*, 44(2):1-46. https://doi.org/10.1145/3291933

Guthrie D, Allison B, Liu W, et al., 2006. A closer look at skip-gram modelling. Proc 5th Int Conf on Language Resources and Evaluation, p.1-4.

Halpin H, Robu V, Shepherd H, 2007. The complex dynamics of collaborative tagging. Proc 16th Int Conf on World Wide Web, p.211-220. https://doi.org/10.1145/1242572.1242602

Heckner M, Heilemann M, Wolff C, 2009. Personal information management vs. resource sharing: towards a model of information behaviour in social tagging systems. Proc 3rd Int AAAI Conf on Weblogs and Social Media, p.42-49.

Heymann P, Garcia-Molina H, 2006. Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. InfoLab Technical Report, Stanford.

Heymann P, Koutrika G, Garcia-Molina H, 2008. Can social bookmarking improve web search? Proc Int Conf on Web Search and Data Mining, p.195-206. https://doi.org/10.1145/1341531.1341558

Hindle A, Barr ET, Su ZD, et al., 2012. On the naturalness of software. Proc 34th Int Conf on Software Engineering, p.837-847. https://doi.org/10.1109/ICSE.2012.6227135

Körner C, Kern R, Grahsl HP, et al., 2010. Of categorizers and describers: an evaluation of quantitative measures for tagging motivation. Proc 21st ACM Conf on Hypertext and Hypermedia, p.157-166. https://doi.org/10.1145/1810617.1810645

Levenshtein VI, 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Sov Phys Dokl*, 10(8):707-710.

Ponte JM, Croft WB, 1998. A language modeling approach to information retrieval. Proc 21st Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.275-281. https://doi.org/10.1145/290941.291008

Robu V, Halpin H, Shepherd H, 2009. Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM Trans Web*, 3(4):14. https://doi.org/10.1145/1594173.1594176

Rosenfeld R, 1994. A hybrid approach to adaptive statistical language modeling. Proc Workshop on Human Language Technology, p.76-81. https://doi.org/10.3115/1075812.1075827

Rosenfeld R, 1995. Optimizing lexical and $N$-gram coverage via judicious use of linguistic data. Proc European Conf on Speech Technology, p.1763-1766.

Schenkel R, Crecelius T, Kacimi M, et al., 2008. Efficient top-$k$ querying over social-tagging networks. Proc 31st Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.523-530. https://doi.org/10.1145/1390334.1390424

Schmitz C, Hotho A, Jäschke R, et al., 2006. Mining association rules in folksonomies. In: Batagelj V, Bock HH, Ferligoj A, et al. (Eds.), Data Science and Classification. Springer, Berlin, p.261-270. https://doi.org/10.1007/3-540-34416-0_28

Sigurbjörnsson B, van Zwol R, 2008. Flickr tag recommendation based on collective knowledge. Proc 17th Int Conf on World Wide Web, p.327-336. https://doi.org/10.1145/1367497.1367542

Siu M, Ostendorf M, 2000. Variable $N$-grams and extensions for conversational speech language modeling. *IEEE Trans Speech Audio Process*, 8(1):63-75. https://doi.org/10.1109/89.817454

Song Y, Zhuang ZM, Li HJ, et al., 2008. Real-time automatic tag recommendation. Proc 31st Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.515-522. https://doi.org/10.1145/1390334.1390423

Storey MA, Cheng LT, Bull I, et al., 2006. Waypointing and social tagging to support program navigation. CHI Extended Abstracts on Human Factors in Computing Systems, p.1367-1372. https://doi.org/10.1145/1125451.1125704

Strohmaier M, Körner C, Kern R, 2010. Why do users tag? Detecting users' motivation for tagging in social tagging systems. Proc 4th Int AAAI Conf on Weblogs and Social Media, p.23-26.

Thom-Santelli J, Muller MJ, Millen DR, 2008. Social tagging roles: publishers, evangelists, leaders. Proc SIGCHI Conf on Human Factors in Computing Systems, p.1041-1044. https://doi.org/10.1145/1357054.1357215

Tuarob S, Pouchard LC, Giles CL, 2013. Automatic tag recommendation for metadata annotation using probabilistic topic modeling. Proc 13th ACM/IEEE-CS joint Conf on Digital Libraries, p.239-248. https://doi.org/10.1145/2467696.2467706

Wagner C, Singer P, Strohmaier M, et al., 2014. Semantic stability in social tagging streams. Proc 23rd Int Conf on World Wide Web, p.735-746. https://doi.org/10.1145/2566486.2567979

Wang SW, Lo D, Vasilescu B, et al., 2014. EnTagRec: an enhanced tag recommendation system for software information sites. Proc IEEE Int Conf on Software Maintenance and Evolution, p.291-300. https://doi.org/10.1109/ICSME.2014.51

Wattenberg M, Viégas FB, 2008. The word tree, an interactive visual concordance. *IEEE Trans Vis Comput Graph*, 14(6):1221-1228. https://doi.org/10.1109/TVCG.2008.172

Xia X, Lo D, Wang XY, et al., 2013. Tag recommendation in software information sites. Proc 10th Working Conf on Mining Software Repositories, p.287-296. https://doi.org/10.1109/MSR.2013.6624040

Xie XK, Jin PQ, Yiu ML, et al., 2016. Enabling scalable geographic service sharing with weighted imprecise Voronoi cells. *IEEE Trans Knowl Data Eng*, 28(2):439-453. https://doi.org/10.1109/TKDE.2015.2464804

Xie XK, Lin X, Xu JL, et al., 2017. Reverse keyword-based location search. Proc IEEE 33rd Int Conf on Data Engineering, p.403-434. https://doi.org/10.1109/ICDE.2017.96

Zubiaga A, 2012. Enhancing navigation on Wikipedia with social tags. https://arxiv.org/abs/1202.5469v1