Automatic Generation of Grounded Visual Questions

Shijie Zhang  
Tianjin University  
Tianjin, China  
shijiezhang@tju.edu.cn

Lizhen Qu  Shaodi You  
Data61-CSIRO  
Australian National University  
Canberra, Australia

Zhenglu Yang  
Nankai University Tianjin, China

Jiawan Zhang  
Tianjin University  
Tianjin, China

Abstract

In this paper, we propose a new task and solution for vision and language: generation of grounded visual questions. Visual question answering (VQA) is an emerging topic which links textual questions with visual input. To the best of our knowledge, it lacks automatic method to generate reasonable and versatile questions. So far, almost all the textual questions are generated manually, as well as the corresponding answers. To this end, we propose a system that automatically generates visually grounded questions. First, visual input is analyzed with deep caption model. Second, the captions along with VGG-16 features are used as input for our proposed question generator to generate visually grounded questions. Finally, to enable generating of versatile questions, a question type selection module is provided which selects reasonable question types and provide them as parameters for question generation. This is done using a hybrid LSTM with both visual and answer input. Our system is trained using VQA and Visual7W dataset and shows reasonable results on automatically generating of new visual questions. We also propose a quantitative metric for automatic evaluation of the question quality.

1. Introduction

Multi-model learning of vision and language is an emerging task in artificial intelligence. Early work focus on describe visual scenes using natural languages (image caption) [3, 9, 15, 12, 20, 21, 22, 32, 34]; and recently automatic answering of visual related queries (VQA) quickly attracts attention. [1, 24, 7, 29, 35, 36]. By far, VQA systems assume query answers exists in the visual input with no ambiguity (Grounded).

We consider one step forward in the scope artificial intelligence: if an AI can answer a visual related question, is it possible to ask grounded questions regarding the visual input? To the best of our knowledge, existing VQA systems relying on human specified questions [1, 24, 7, 29, 35, 36], which is not automatic. Or they generate questions with few limited patterns [29, 36], which is not versatile. Or they generate ambiguous questions that answers are not determined by the image[30]. Thus, it is desired that an AI can ask versatile grounded visual questions. The possible applications could be: education, user query prediction, interactive navigation, reinforce learning and etc.

In this paper, we consider the above mentioned task as VQG: automatic generation of grounded visual questions. The generation is described to be versatile and natural and the in the meanwhile, the answers make sure the question answer is determined by the image. We propose a base-line system along with a benchmark for the above mentioned task. As illustrated in Fig. 2, first, visual input is analyzed with VGG-16 network and DenseCap [14], which provides a over-sample information coverage for possible questions. Such information both visual and textual are used as input for our question generator. Second, the question type selector sample the dense caption pool and determine possible question types. Finally, the question type, along with dense visual and textual information are used for question generator for final questions. We provide the base-line eval-
ution by comparing the similarity between human-asked questions and our automatically generated questions. The similarity is evaluated using BLEU [26].

The contributions of this paper. First, to the best of our knowledge, this is the work to consider ask versatile visually grounded questions. Second, we propose a fully automatic base-line system for the proposed task. And finally, we provide the evaluation strategy along with insights on further systems to ask and learn from visual input.

The rest of the paper is organized as follows: Section 2 discusses the related work, followed by the detail of system in Section 3. Section 4 introduces the experiments, followed by the discussion in Section 5. Section 6 concludes the paper.

2. Related Work

Textual description of visual information has been widely exploited. This includes joint learning of visual and textual information [2, 18, 37]. A typical task is to described the image scene with a few sentences, say, image caption. This includes early working on describing the contents [2] and later the context and/or scenario [3, 9, 15, 12, 20, 21, 22, 32, 34].

Visual Question and Answering Automatic answering of textual queries is an important task [13], and can be back dated to Turing test. An extension of retrieving textual information from visual input is to answer textual queries regarding a visual input. Such task is usually called visual question answering (VQA) [8, 24, 25, 28, 29, 33, 35].

VQA methods are now been evaluated on a few dataset [1, 24, 7, 29, 35, 36]. For these dataset, while images are collected by sub-sampling MS Coco [23], the questions-answer paired are manually manually generated [1, 7, 35, 36] or by using NLP tools [29] that converts limited types of image captions into query forms.

Visual Question Generation While asking questions automatically is well exploited in NLP, it is yet remain un-exploited for visual related questions. Such questions are strongly desired by creating VQA dataset. Early methods simply converting image labeling into questions, which only allows generation of low-level questions. To diversify the question, however, still requires heavy labor force [1, 7, 24]. Zhu et al. [36], recently categorizes the manually generated questions into 7W question types, say, what, where, when and etc. Yu et al. [35] exploited generating question which chops off a few key words in a caption. As they describe there task as fill in the blank. Similar, Ren et al. convert the fill in the blank question into a question form, but yet the question type is limited by the caption format.

Aside from that, very recently Mostafazadeh et al. [30] exploit general human like question according to visual input. However, what they generate are ambiguous open questions where no determined answer is available within the visual input.

In a word, automatically generation of reasonable, and in the meanwhile, versatile close-form questions remains a challenging problem.

3. Question Generation

Our goal is to generate visually grounded questions directly from images with diverse question types. We start with randomly picking a caption from a set of automatically generated captions, which describes a certain region of image with natural language. Then we sample a question type based on the caption because we cannot ask all types of questions for that caption. In the last step, our question generator learns the correlation between the caption and the image, generates a question of the chosen type.

Formally, our model takes as input a single raw image \( x \), generates a set of captions \( \{c_1, c_2, ..., c_M\} \), followed by yielding a set of grounded questions \( \{q_1, q_2, ..., q_M\} \). Herein, a caption or a question is a sequence of words.

\[
    w = \{w_1, ..., w_L\}
\]

Where \( L \) is the length of the word sequence. Each word \( w_i \) employs 1-of-\( K \) encoding, where \( K \) is the size of the vocabulary.

For each image \( x_i \), we apply a dense caption model (DenseCap) [14] trained on the Visual Genome dataset [19] to produce a set of captions \( C_i \). Then the generative process is described as follows:

1. Choose a caption \( c_n \) from \( C_i \).
2. Choose a question type \( t_n \) given \( c_n \).
3. Generate a question \( q_n \) conditioned on \( c_n \) and \( t_n \).

where \( t_n \) is a \( T \)-dimensional one hot vector and \( T \) is the number of question types.

Given the parameters \( \theta \), for each image \( x_i \), the joint distribution of \( c_n, t_n \) and \( q_n \) is given by:

\[
P(q_n, t_n, c_n | x_i, C_i, \theta) = P(q_n | c_n, x_i, t_n) P(t_n | c_n) P(c_n | C_i)
\]

(2)

where \( P(q_n | c_n, x_i, t_n) \) is the distribution of generating question, \( P(t_n | c_n) \) and \( P(c_n | C_i) \) are the distributions for sampling question type and caption respectively. More details are given in the following sections.

Since we do not observe the alignment between captions and questions, \( c_n \) is latent. Sum over \( c \), we obtain:

\[
P(q_n, t_n | x_i, C_i) = \sum_{c_n \in C_i} P(q_n, t_n, c_n | x_i, C_i, \theta)
\]
Let $Q_i$ denote the set of question of the image $x_i$, the probability of the training dataset $D$ is given by taking the product of the above probabilities over all images and their questions.

$$P(D|\theta) = \prod_i \prod_n P(q_n, t_n | x_i, C_i)$$

For word representations, we initialize a word embedding matrix $E \in \mathbb{R}^{300 \times K}$ by using Glove [27], which are trained on 840 billions of words. For the image representations, we apply a VGG-16 model [31] trained on ImageNet [6] without fine-tuning to produce 300-dimensional feature vectors. The dimension is chosen to match the size of the pre-trained word embeddings.

Compared to the question generation model [30], which generates only one question per image, the probabilistic nature of this model allows generating questions of multiple types and referring to different region of interests, because each caption predicted by DenseCap is associated with a different region.

### 3.1. Sample captions and question types

The caption model DenseCap generates a set of captions for a given image. Each caption $c$ is associated with a region and a confidence $o_c$ of the proposed region. Intuitively, we should give a higher probability to the caption with higher confidence than the lower one. Thus, given a caption set $C_i$ of an image $x_i$, we define the prior distribution as:

$$P(c_k | C_i) = \frac{\exp(o_k)}{\sum_j \exp(o_j)}$$

A caption is either a declarative sentence, a word, or a phrase. We are able to ask many different types of questions but not all of them for a chosen caption. For example, for a caption “floor is brown” we can ask “what color is the floor” but it would be awkward to ask a who question. Thus, our model draws a question type given a caption with the probability $P(t_n|c_n)$ by assuming it suffices to infer question types given a caption.

Our key idea is to learn the association between question types and key words/phrases in captions. The model $P(t_n|c_n)$ consists of two components. The first one is a Long Short Term Memory (LSTM) [11] that maps a caption into a hidden representation. LSTM is a recurrent neural network taking the following form:

$$h_t, m_t = LSTM(x_t, h_{t-1}, m_{t-1})$$

where $x_t$ is the input and the hidden state of LSTM at time step $t$, and $h_t$ and $m_t$ are the hidden states and memory states of LSTM at time step $t$, respectively. As the representation of the whole sequence, we take the last state $h_L$ generated at the end of the sequence. This representation is further fed into a softmax layer to compute a probability vector $p_t$ for all question types. The probability vector characterizes a multinomial distribution, from which we draw samples of question types.

### 3.2. Generate questions

At the core of our model is the question generation module, which models $P(q_n|c_n, x_i, t_n)$, given a chosen caption $c_n$ and a question type $t_n$. It is composed of three modules: i) an LSTM encoder to generate caption embeddings; ii) a correlation module to learn the association between images and captions; iii) a decoder consisting of an LSTM decoder and an ngram language model.

A grounded question is deeply anchored in both the sampled caption and the associated image. In our preliminary experiments, we found it useful to let the LSTM encoder LSTM$(x_t, h_{t-1}, m_{t-1})$ to read the image features prior to reading captions. In particular, at time step $t = -1$, we initialize the state vector $m_{-1}$ to zero and feed the image features as $x_{-1}$. At the 0-th time step, the encoder reads in a special token $S_0$ indicating the start of a sentence, which is a good practice adopted by many caption generation models [1]. After reading the whole caption of length $L$, the encoder yields the last state vector $m_L$ as the embedding of caption.

The correlation module takes as input the caption embeddings from the encoder and the image features from VGG-16, produces a 300-dimensional joint feature map. We apply a linear layer of size $300 \times 600$ and a PReLU [10] layer in sequel to learn the associations between captions and images. Since an image gives an overall context and the cho-
sen caption provides the focus in the image, the joint representation provides sufficient context to generate grounded questions. Although the LSTM encoder starts with reading the image features before captions, the generated questions seem to be less “grounded” without this correlation module.

Our decoder extends the LSTM decoder of [32] with a ngram language model. The LSTM decoder consists of an LSTM layer and a softmax layer. The LSTM layer starts with reading the joint feature map and the start token $S_0$ in the same fashion as the caption encoder. From time step $t = 0$, the softmax layer predicts the most likely word given the state vector at time $t$ yielded by the LSTM layer. A word sequence ends when the end of sequence token is produced.

**Joint decoding** Although the LSTM decoder alone can generate questions, we found that it would frequently produce repeated words and phrases such as “the the”. The problem didn’t disappear even the beam search [] was applied. It is due to the fact that the state vectors produced at adjunct time steps tend to be similar. Since repeated words and phrases are rarely observed in text corpora, we discount such occurrence by joint decoding with a ngram language model. Given a word sequence $w = \{w_1, ..., w_N\}$, a bigram based language model is defined as:

$$P(w) = \prod_{i=2}^{N} P(w_i|w_{i-1})P(w_0)$$

Instead of using neural models, we adopt the word count based estimation of model parameters. In particular, we apply the KneserNey smoothing [17] to estimate $P(w_i|w_{i-1})$, which is given by:

$$\frac{\text{max} (\text{count}(w_{i-1}, w_i) - d, 0) \times \lambda(w_{i-1})}{\text{count}(w_i)} + \lambda(w_{i-1})P_K N(w_i)$$

where $\text{count}(x)$ denotes the corpus frequency of term $x$, $P_K N(w_i)$ is a back-off statistic of unigram $w_i$ in case the bigram $(w_i, w_{i-1})$ does not appear in the training corpus. The parameter $d$ is usually fixed to 0.75 to avoid overfitting for low frequency bigrams. And $\lambda(w_{i-1})$ is a normalizing constant conditioned on $w_{i-1}$.

We incorporate bigram statistics with the LSTM decoder from the time step $t = 1$ because the LSTM decoder can well predict the first words of questions. The LSTM decoder essentially captures the conditional probability $P_l(q_t|q_{<t})$, while the bigram model considers only the previous word $P_b(q_t|q_{<t-1})$ by using word counts. By interpolating these two, we obtain the final probability as:

$$P(q_t|q_{<t}) = (1 - \beta)P_l(q_t|q_{<t}) + \beta P_b(q_t|q_{t-1})$$

where $\beta \in [0, 1]$ is an interpolation weight. In addition, we fix the first $k$ words of questions during decoding according to the chosen question types.

### 3.3. Training

The key challenge of training is the involvement of the latent variables indicating the alignment between captions and gold standard questions for a deep neural network. We estimate the latent variables in a similar fashion as EM but computationally more efficient.

Suppose we are given the training set $\{(x_1, q_1), ..., (x_N, q_N)\}$, the loss is given by:

$$l(\theta) = \sum_{i=1}^{N} \sum_{n \in Q_i} - \log P(q_{in}, t_n|x_i, C_i)$$

Suppose $Q(c_n)$ denote some proposed distribution such that $\sum_n Q(c_n) = 1$ and $Q(c_n) \geq 0$. Consider the following:

$$\log P(q_{in}, t_n|x_i, C_i) = \log \sum_{c_k \in C_i} P(c_k, t_n, c_k|x_i, C_i, \theta) \frac{P(q_{in}, t_n, c_k|x_i, C_i, \theta)}{Q(c_k)}$$

$$\geq \sum_{c_k \in C_i} Q(c_k) \log P(q_{in}, t_n, c_k|x_i, C_i, \theta) / Q(c_k)$$

(5)

The last step used Jensens inequality. The Equation (5) gives a lower bound of the loss $l(\theta)$. When the bound is tight, we have $Q(c_k) = P(c_k|q_{in}, t_n)$.

To save the EM loop, we propose a non-parametric estimation of $P(c_k|q_{in}, t_n)$. As a result, for each question-image pair $(x_i, q_{in})$, we maximize the lower bound by optimizing:

$$\arg \min_{\theta} -P(c_k|q_{in}, t_n) \log[P(c_k|q_{in}, x_i, t_n)P(t_n|c_n)] + \text{const}$$

This in fact assigns a weight $P(c_k|q_{in}, t_n)$ to each instance. By using a non-parametric estimation, we are still able to apply BackProp and the SGD style optimizing algorithms by just augmenting each instance with an estimated weight.

Given a question $q$ and a caption set $C$ from the train set, we estimate $P(c_k|q, t)$ by using the kernel density estimator []:

$$P(c_k|q, t) = P(c_k|q) = \frac{s(q, c_k)}{\sum_{c_j \in C} s(q, c_j)}$$

(6)

where $s(q, c)$ is a similarity function between a question and a caption. We assume $c_k$ are conditionally independent of $t$ because we can directly extract the question type from the question $q$ by looking at the first few words.

For a given question, there are usually very few matched captions generated by DenseCap, hence the distribution of captions given a question is highly skewed. It is sufficient to randomly draw a caption each time to compute the probability based on Equation (6).

We formulate the similarity between a question and a caption by using both string similarity and embedding based similarity measures.
The surface string of a caption could be an exact or partial match of a given question. Thus we employ the Jaccard Index as string similarity measure between the surface string of a caption and that of a question.

\[ \text{sim}_s(q, c) = \frac{q \cap c}{q \cup c} \]

where \( c \) and \( q \) denote their surface string respectively. Both strings are broken down to a set of char-based trigrams during the computation so that this measure still gives a high similarity if two strings differ only in some small variations such as singular and plural forms of nouns.

In case of synonyms or words of similar meanings come with different form such as “car” and “automobile”, we adopt the pre-trained word embeddings to calculate their similarity by using the weighted averaged of word embeddings:

\[ \text{sim}_w(q, c) = \cos\left( \sum_{w_i \in q} \sum_{w_j \in c} \frac{\text{IDF}(w_i)}{\text{IDF}(w_j)} E_{w_i} \cdot \sum_{w_k \in c} \frac{\text{IDF}(w_k)}{\text{IDF}(w_j)} E_{w_k} \right) \]

where \( \cos \) denotes the cosine similarity, \( \text{IDF}(x) \) is the inverse document frequency of word \( x \) defined by \( \frac{|\{d \in D | x \in d\}|}{|D|} \), and \( D \) is the corpus containing all questions, answers, and captions.

The final similarity measure is computed as the interpolation of the two measures:

\[ s(q, c) = \alpha \text{sim}_s(q, c) + (1 - \alpha) \text{sim}_w(q, c) \]

where the hyperparameter \( \alpha \in (0, 1) \).

4. Experimental Setup

4.1. Datasets

We conduct our experiments on two data-sets: VQA-Dataset [1] and Visual7W [36]. The former is the most popular benchmark for VQA and the latter is a recently created dataset focusing on visually grounded questions. VQA: is a subsample from the MS-COCO dataset [23], which contains 254,721 images and 764,163 manually composed questions respectively. Each image is associated with three questions on average. Visual7W: is composed of 327,939 QA pairs on 47,300 COCO images, which are collected from the MS-COCO dataset [23] as well. In addition, it includes 1,311,756 human-generated answers in form of multiple-choice and 561,459 object groundings from 36,579 categories. Each image is associated with five questions on average.

4.2. Baseline

In this paper, we consider a baseline by training the image caption generation model NeuralTalk2 [32] on image-question pairs. The baseline is similar to [30], which is the only work generating questions from visual input, to the best of our knowledge. Instead of visually grounded questions, Mostafazadeh et al.’s method generates only abstract and ambiguous questions. The model of neuraltalk2 differs from [30] only in the RNN used in the decoder. NeuralTalk2 adopts LSTM while [30] considers GRU [4]. The two RNN models achieve almost identical performance in language modeling [5].

4.3. Evaluation measures

As a common practice for evaluating generated word sequences we employ four different evaluation metrics: BLEU [26], METEOR [4], ROUGE [4], and CIDEr [4] to evaluate questions generated by our model against the best matching ground truth question of the same images, which can be regarded as an analogy of precision. Notice that, the generated questions could be reasonable and grounded but not covered by the ground truth.

To measure the diversity of our generated questions, we also compute the same set of evaluation measures by comparing each reference sentence with the best matching generated sentence of the same images. This provides an estimate of coverage in analogy of recall.

4.4. Implementation details

We optimize all models with Adam [16]. We fix the batch size to 32. We set the maximal epochs to 5 for Visual7W and the maximal epoch to 1 for VQA. The corresponding model hyperparameters were tuned on the validation sets.

5. Results and Discussions

Could our model generate meaningful and grammatically correct questions? For each generated question, we find the gold standard question with maximal score (BLEU etc.) and average them over all generated questions. We evaluate the baseline also in the same manner.

As illustrated by Figure Fig. 3, the average question quality generated by our model is way better than NeuralTalk2 in particular in terms of CIDEr, one of the newest measure designed for evaluating sentences generated from images. Most questions generated by our model are visually grounded and grammatically correct. The inclusion of ngram language model provides a superb smoothing effect that removes almost all the repeated words/phrases.

Figure 4 also shows a weakness of comparing generated questions with reference ones because many reasonable and grounded questions are not covered by ground truth.

During the training of our model, we observe also a nice progress that our model was only able to generate similar questions starting with what but tend to generate more diverse questions as the learning progresses. This is evident by the line of CIDEr, as shown by Figure Fig. 3.
How diverse are our generated questions? We summarize the generated questions by their types in Table 1, in which we show their frequency of the top 50 most frequent question types. And we can see that our method shows a reasonable diversity that covers a wide spectrum of question types. In contrast, the questions generated by NeuralTalk2 fall into only 2 types starting with "what". However, there is still a gap between the generated questions and the manually constructed reference ones in terms of versatility. The may be caused by the imbalanced distribution of question types in the training datasets.

NeuralTalk2 generates questions of single type, because we selected only 7W questions.

6. Conclusion

In this paper, we propose method to automatically generate versatile visually grounded questions. First, visual input is analyzed with deep caption model. Second, the captions along with VGG-16 features are used as input for our proposed question generator to generate visually grounded questions. Finally, to enable generating of versatile questions, a question type selection module is provided which selects reasonable question types and provide them as parameters for question generation. Experiments on VQA and Visual7W dataset and demonstrates that the proposed method can generate reasonable questions and in the meanwhile more versatile than manually provided questions. For future work, we consider automatically generation of visual question-answer pairs, which will likely to enforce VQA systems.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In Proceedings of the IEEE International Conference on Computer Vision, pages 2425–2433, 2015. 1, 2, 5
[2] K. Barnard, P. Duygulu, D. Forsyth, N. d. Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. Journal of machine learning research, 3(Feb):1107–1135, 2003. 2
Fig. 4. Comparison between manually composed questions and automatically generated questions by our method. Our method generates reasonable grounded questions and more versatile the manually provided ones.

[3] X. Chen and C. L. Zitnick. Learning a recurrent visual representation for image caption generation. *arXiv preprint arXiv:1411.5654*, 2014. 1, 2

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 5

[5] J. Chung, C. Gülcehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. *CoRR, abs/1502.02367*, 2015. 5

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 3

[7] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question. In *Advances in Neural Information Processing Systems*, pages 2296–2304, 2015. 1, 2
Table 1. Top 50 most frequent question types. Left, Our method, full automatic; and right, Visual7W[36], mostly manual.

| Rank | Our (Automatic) | Count | Percentage | Vis7w (Manual) | Count | Percentage |
|------|-----------------|-------|------------|----------------|-------|------------|
| 1    | 'what is '      | 10207 | 36.4275517 | 'what is '     | 5292  | 18.865096  |
| 2    | 'where is '     | 3321  | 11.8522484 | 'how many '    | 3125  | 11.152748  |
| 3    | 'what color '   | 3118  | 11.1277659 | 'what color '  | 3115  | 11.117052  |
| 4    | 'how many '     | 3040  | 10.8493933 | 'who is '      | 2222  | 7.9300496  |
| 5    | 'what are '     | 2585  | 9.22555318 | 'where is '    | 2197  | 7.84082798 |
| 6    | 'why is '       | 1852  | 6.6095646  | 'what are '    | 1214  | 4.33261956 |
| 7    | 'why are '      | 599   | 2.13775874 | 'where was '   | 1168  | 4.16845111 |
| 8    | 'what does '    | 538   | 1.9200571  | 'where are '   | 1032  | 3.68308351 |
| 9    | 'where are '    | 352   | 1.25624554 | 'what is '     | 1020  | 3.64025696 |
| 10   | 'is this '      | 336   | 1.19914347 | 'when was '    | 818   | 2.91934333 |
| 11   | 'what animals ' | 238   | 0.84939329 | 'how is '      | 513   | 1.83083512 |
| 12   | 'is the '       | 227   | 0.81013562 | 'why are '     | 504   | 1.7987152  |
| 13   | 'is it '        | 224   | 0.79942898 | 'what kind '   | 463   | 1.65239115 |
| 14   | 'what colors '  | 206   | 0.73518915 | 'what does '   | 384   | 1.37044968 |
| 15   | 'how is '       | 162   | 0.57815846 | 'what type '   | 340   | 1.21341899 |
| 16   | 'what time '    | 120   | 0.42826552 | 'when is '     | 248   | 0.88508208 |
| 17   | 'which the '    | 116   | 0.41399001 | 'what animal ' | 173   | 0.61741613 |
| 18   | 'what kind '    | 101   | 0.36045682 | 'who has '     | 168   | 0.59591713 |
| 19   | 'how are '      | 98    | 0.34975018 | 'what time '   | 161   | 0.57458958 |
| 20   | 'what brand '   | 59    | 0.21056388 | 'how does '    | 153   | 0.54603854 |
| 21   | 'what is '      | 53    | 0.18915061 | 'how are '     | 150   | 0.53531919 |
| 22   | 'what animal '  | 49    | 0.17487509 | 'what has '    | 120   | 0.42826552 |
| 23   | 'what country ' | 48    | 0.17103621 | 'who are '     | 114   | 0.40685225 |
| 24   | 'what number '  | 39    | 0.1391863  | 'what sport '  | 101   | 0.36045682 |
| 25   | 'what team '    | 37    | 0.13204854 | 'what colors ' | 94    | 0.33547466 |
| 26   | 'what game '    | 34    | 0.1213419  | 'when will '   | 83    | 0.29621699 |
| 27   | 'how tall '     | 23    | 0.08208423 | 'what shape '  | 82    | 0.29264811 |
| 28   | 'why does '     | 20    | 0.07137759 | 'what do '     | 73    | 0.26052819 |
| 29   | 'what city '    | 20    | 0.07137759 | 'what number ' | 66    | 0.23546046 |
| 30   | 'what two '     | 19    | 0.06780871 | 'what animals '| 64    | 0.22840828 |
| 31   | 'what fruit '   | 16    | 0.05710207 | 'what pattern '| 62    | 0.22127052 |
| 32   | 'what vegetables'| 15   | 0.05353319 | 'who took '    | 60    | 0.21413276 |
| 33   | 'what airline ' | 14    | 0.04996431 | 'what game '   | 60    | 0.21413276 |
| 34   | 'are the '      | 12    | 0.04282655 | 'where does '  | 58    | 0.206995   |
| 35   | 'how long '     | 10    | 0.03566879 | 'what material '| 55    | 0.19628837 |
| 36   | 'how much '     | 10    | 0.03566879 | 'what direction'| 54    | 0.1971949 |
| 37   | 'what direction'| 9     | 0.03211991 | 'what room '   | 51    | 0.18201285 |
| 38   | 'what street '  | 8     | 0.02855103 | 'why does '    | 50    | 0.17844397 |
| 39   | 'how high '     | 8     | 0.02855103 | 'what brand '  | 50    | 0.17844397 |
| 40   | 'what room '    | 6     | 0.02141328 | 'what else '   | 47    | 0.16773733 |
| 41   | 'what fruits '  | 6     | 0.02141328 | 'how do '      | 45    | 0.16059957 |
| 42   | 'how can '      | 6     | 0.02141328 | 'why was '     | 45    | 0.16059957 |
| 43   | 'what language '| 6     | 0.02141328 | 'what can '    | 42    | 0.14989293 |
| 44   | 'are these '    | 5     | 0.0178444  | 'why do '      | 41    | 0.14632405 |
| 45   | 'how old '      | 5     | 0.0178444  | 'what season ' | 40    | 0.14275517 |
| 46   | 'how does '     | 4     | 0.01427552 | 'why the '     | 38    | 0.13561742 |
| 47   | 'how do '       | 3     | 0.01070664 | 'where do '    | 35    | 0.12491078 |
| 48   | 'who or '       | 3     | 0.01070664 | 'what was '    | 34    | 0.1213419 |
| 49   | 'who are '      | 3     | 0.01070664 | 'what vehicle '| 31    | 0.11063526 |
| 50   | 'what numbers ' | 3     | 0.01070664 | 'what food '   | 31    | 0.11063526 |
