

相互情報量最小化による例文に基づく制御可能な言い換え生成

杉浦昇太¹ 西田典起² 朱中元³ 中山英樹¹

¹ 東京大学大学院 {sugiura, nakayama}@nlab.ci.i.u-tokyo.ac.jp

² 理化学研究所 AIP noriki.nishida@riken.jp

³ NAVER AI Research shu.raaphael@navercorp.com

1 はじめに

言葉を同じ意味内容を表す異なる表現に言い換えることは、コミュニケーションをより円滑にするために重要である。意図が伝わらない場合はより平易な表現に言い換え、あるいは場の状況に合わせてよりくだけた表現や丁寧な表現に言い換える。このように言葉を適切に言い換える能力は、人間のみならず機械にも求められる。

言い換え生成とは、所与の文の言い換えを自動的に生成する自然言語処理技術である。近年、ニューラルネットワークを用いた言い換え生成の手法が数多く提案されており、妥当で自然な言い換えの生成を達成しつつある [1, 2]。一方で、自然言語の持つ表現の多様性を踏まえると、利用者の望むような表現で書かれた文を得るためには、与えられた文に対してただ一つの言い換えを生成するだけでは不十分である。

本研究では、利用者が言い換えるスタイルを制御可能な言い換え生成の手法を提案する。ここでスタイルとは、文体や言い回し、構文など、文の意味内容に関わらないような表層的な表現のことである。図 1 に、提案手法の概要を示す。提案手法では、言い換えの対象となる入力文とともに利用者によって指定された例文を入力として、入力文の意味を保ちつつ例文のスタイルを持つような言い換え文を生成する。例えば、入力文として “What is the best way to earn money?”、例文として “How can I learn English?” という文が提示される状況を考える。システムは入力文の意味を保ちつつ、例文の構造、言い回しなどを可能な限り真似た出力文を生成する。この例では “How can I earn money?” が条件に即した妥当な言い換えであると言える。

Iyyer ら [3] や Chen ら [4] は、出力の統語構造を指定する言い換え生成手法を開発した。しかし、利用

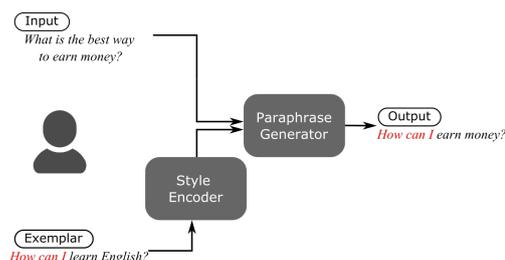


図 1 提案手法の概要。言い換え生成システムは、入力文 (input) と例文 (exemplar) を受け取り、言い換え (output) を生成する。

者が望むような文を出力するのに統語構造だけで十分であるとは限らず、また言語学的な素養のない一般の利用者にとっては統語構造による操作は負担が大きい。そこで本研究では、生の例文に基づき、統語構造だけでなく文体や言い回しなども含めてよりそれに近い文を生成することを目指す。

2 手法

本研究では、データセットとして言い換え文の対を含むパラレルコーパス $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}$ を使用する。ここで、 \mathbf{x} と \mathbf{y} はほぼ同じ意味を持つが、異なる表現で書かれた文である。

提案手法のモデルは、二つの構成要素からなる。符号化モデル g_θ は、例文 \mathbf{e} を符号化してスタイルの潜在変数 \mathbf{z} を生成する機構である。系列変換モデル p_θ は、文 \mathbf{x} と潜在変数 \mathbf{z} を受け取り、出力として \mathbf{x} の言い換え文を生成する機構である。この2つのモデルのパラメータをまとめて θ と表記する。

推論時には、系列変換モデルは入力文 \mathbf{x} を参照して生成すべき文の意味を、 $\mathbf{z}_e = g_\theta(\mathbf{e})$ を参照してスタイルの情報を得て、これらに従って言い換え文を生成する。

学習時には、例文は提示されず、言い換え文の対 (\mathbf{x}, \mathbf{y}) のみが与えられる。そこで、例文ではなく \mathbf{y} を符号化してスタイルの情報 $\mathbf{z}_y = g_\theta(\mathbf{y})$ を抽出し、 \mathbf{x}

と z_y から y を復元するような学習を行う。推論時に適切な言い換えを生成するためには、学習に際して以下の2点に留意する必要がある。

- (1) 入力文 x と潜在変数 z を受け取り、条件に沿った言い換えを生成するような系列変換モデルを学習する。
- (2) 潜在変数 z が、例文 (学習時には y) の意味内容ではなく、スタイルに関わる情報のみを保持するように符号化モデルを学習する。

(2) の重要性について触れる。仮に $z_y = g_\theta(y)$ が y に関する情報を完全に保持している場合、系列変換モデルは z_y のみを見て y を復元するように学習してしまう。そうすると、推論時には入力文 x と同義な文を生成できなくなり、言い換え生成として成立しなくなる。従って、 z が文の意味ではなくスタイルの情報のみを保持するような制約を加えて学習することが必要となる。

2.1 損失関数の設計

学習時は、言い換え対 $(x, y) \in \mathcal{D}$ を受け取り、 x から y を生成するようにモデルを訓練する。Cross-Entropy 損失関数 \mathcal{L}_{GEN} は次のように定義できる。

$$\mathcal{L}_{\text{GEN}} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} -\log p_\theta(y_i | x_i, z_{y_i} = g_\theta(y_i)). \quad (1)$$

相互情報量に基づく損失 符号化モデル g_θ が文の意味内容ではなく、スタイルに関わる情報のみを抽出するように制約を導入する。言い換え対 x と y は意味は同じであるが、それぞれ表層的に異なる言語表現のはずである。つまり、符号化モデルは言い換え文の対が共有する意味に関わる情報ではなく、それらが共有しないスタイルに関わる情報のみを抽出できれば良いということになる。これは、 x と y をそれぞれ符号化したベクトル表現 $z_x = g_\theta(x)$, $z_y = g_\theta(y)$ が、相互に依存することがないような制約を損失関数に導入することで実現できると考えられる。

本研究では、そのような制約を相互情報量 $I(z_x, z_y)$ の最小化問題として定式化する。相互情報量を学習の過程で直接推定することは困難であるため、ここではサンプルベースの相互情報量の上界 [5, 6] を計算し、これを損失関数に加えることで、相互情報量を上から抑えることとする。ミニバッチとして与えられる言い換え対のサンプル $\mathcal{B} = \{(x_j, y_j)\}_{j=1}^M$ につ

いて、これらを符号化モデルによって変換して得られた潜在変数 $\{(z_{x_j}, z_{y_j})\}_{j=1}^M$ を考える。このサンプルを用いて、相互情報量 $I(z_x, z_y)$ の上界は以下のように推定できる。

$$I(z_x, z_y) \leq \mathbb{E} \left[\frac{1}{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \left[\log p(z_{y_j} | z_{x_j}) - \frac{1}{|\mathcal{B}|} \sum_{k=1}^{|\mathcal{B}|} \log p(z_{y_k} | z_{x_j}) \right] \right]. \quad (2)$$

以上から、損失関数 \mathcal{L}_{MI} を以下のように定義する。

$$\mathcal{L}_{\text{MI}} = \frac{1}{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \left[\log q_\phi(z_{y_j} | z_{x_j}) - \frac{1}{|\mathcal{B}|} \sum_{k=1}^{|\mathcal{B}|} \log q_\phi(z_{y_k} | z_{x_j}) \right]. \quad (3)$$

ここで、 $q_\phi(z_y | z_x)$ は、 $p(z_y | z_x)$ を近似したもので、ガウス分布 $q_\phi(z_y | z_x) = \mathcal{N}(\mu_\phi(z_x), \Sigma_\phi(z_x))$ に従うとする。 μ_ϕ と Σ_ϕ は、パラメータ ϕ を持つニューラルネットワークである。 $q_\phi(z_y | z_x)$ の負の対数尤度を最小化することで、 ϕ の学習を行う。

$$\mathcal{L}_{\text{NLL}} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} -\log q_\phi(z_{y_i} | z_{x_i}). \quad (4)$$

パラメータ θ に関しては、次の損失関数を最小化することで最適化する。

$$\mathcal{L} = \mathcal{L}_{\text{GEN}} + \beta \mathcal{L}_{\text{MI}}. \quad (5)$$

β は \mathcal{L}_{MI} 項の大きさを決めるハイパーパラメータである。学習には確率的勾配降下法を利用し、各イテレーションの中で ϕ と θ それぞれを更新する。

2.2 モデルの設計

符号化モデルは、Transformer Encoder [7] と多層パーセプトロンによって構成されている。Transformer の最終層の隠れベクトルの平均をとり、これを多層パーセプトロンで変換したものを出力とする。

系列変換モデルの構造は、通常の Transformer と同じである。ただし、デコーダの最初のステップの入力には SOS トークンではなく、符号化モデルが出力した潜在変数 z に対してアフィン変換を適用して得られたベクトルを用いる。

μ_ϕ と Σ_ϕ はそれぞれ多層パーセプトロンである。 Σ_ϕ は対角行列を出力する。

3 実験と考察

3.1 データセット

Quora Question Pairs¹⁾は、約 14 万件の質問文の言い換えの対を含むデータセットである。10 万件を学習データ、1 万件を検証データとした。テストデータについては、100 件のデータに対して適当な例文を手動で用意した。まず参照文 y と構文が近く、また単語一致率の高い文を、学習データと検証データに利用しない文の集合から抽出した。その中で、入力文 x および参照文 y とは意味内容の異なる文を一つ手動で選び、必要があれば多少の編集を行い、それを例文 e とした。以上のようにして、参照文と意味は異なるものの、スタイルの似ている文をアノテーションした。用意したテストデータの一部を付録に掲載する。

3.2 評価指標

テストデータは、入力文 x 、例文 e 、参照文 y の三つから構成されている。モデルが x と e から生成した文を、以下の二つの観点から評価する。

- (1) 妥当性: 生成された文が入力文の言い換えとなっているか、つまり生成された文の意味が入力文の意味と同じであるか？
- (2) 可制御性: 生成された文が例文と似たようなスタイルの文となっているか？

参照文との比較 本研究では、問題設定に合ったテストデータを用意しているため、より参照文と類似している文を生成できれば、(1) 妥当性と (2) 可制御性を満たしていることになる。そのため、本研究では BLEU [8] により参照文との類似性を評価した。

入力文との比較 生成した文と入力文の意味的な類似度を調べることで、(1) 妥当性について評価する。意味的な類似度の計算方法として Sentence-BERT 類似度を使う。具体的には、入力文と出力文をそれぞれ Sentence-BERT [9] によってベクトルに変換し²⁾、それらのコサイン類似度を意味的な近さとする。

例文との比較 生成した文と例文のスタイルの類似度を調べることで、(2) 可制御性を評価する。スタイルの類似度として、次の二つの尺度を用いた。

1) <https://www.kaggle.com/c/quora-question-pairs>
2) 本研究では <https://github.com/UKPLab/sentence-transformers> の学習済みモデルを用いた。

表 1 Quora のテストデータにおける結果。それぞれ 3 回の試行の結果を平均したものを示している。BLEU は参照文、Sentence-BERT 類似度は入力文、TED 及び LCS は例文と、生成文を比較したものを示す。

| Method | reference | source | exemplar | |
|--------------------|--------------|-----------|----------|--------|
| | BLEU | S-BERT(↑) | TED(↓) | LCS(↑) |
| Seq2Seq [7] | 25.98 | 87.05 | 7.87 | 3.75 |
| SCPN [3] | 41.41 | 83.50 | 4.06 | 4.97 |
| Ours $\beta = 0$ | 39.05 | 67.00 | 3.08 | 6.51 |
| Ours $\beta = 0.1$ | 47.27 | 76.53 | 3.08 | 6.31 |
| Ours $\beta = 1$ | 53.85 | 83.50 | 3.66 | 5.72 |
| Ours $\beta = 10$ | 27.19 | 87.88 | 7.46 | 3.81 |

TED (tree edit distance of syntax tree) TED によって文同士の構文的な類似度を測る。TED では、2 つの文の句構造 (終端ノードは除去) に関する編集距離を計算する。距離であるため、値が小さいほど構文的に近いことを表す。zss³⁾ によって TED を計算した。

LCS (longest common sequence) LCS によって文同士の言い回しの共有度を測る。LCS では、与えられた列の集合に共通する最長の部分列を求める。LCS の長さは、文書要約の自動評価尺度である ROUGE-L [10] に見られるように、自然言語の類似度評価にも利用されてきた。本研究では、文を単語の系列とみなし、生成文と例文との LCS の長さを評価する。LCS が長いほど、生成文が例文に含まれる言い回しを含んでいることを示す。

3.3 ハイパーパラメータ

Transformer は、エンコーダレイヤ及びデコーダレイヤのスタック数をそれぞれ 6、ヘッド数を 8、隠れ層と単語埋め込みの次元数を 256 とした。

モデルの学習には Adam [11] を用い、Vaswani ら [7] と同じ学習率のアニーリングを行った。ミニバッチサイズは 128 で、検証データに対する損失が 10 エポックに渡って減少しなかったときに学習を終了する早期終了を行った。正則化のため、0.1 のドロップアウトを用い、また交差エントロピー誤差の計算に際して 0.1 のラベル平滑化を行った。

Stanford CoreNLP Toolkit [12] によって単語分割を行い、さらに SentencePiece [13] によってサブワードに分割した。サブワードの語彙数は 8000 とした。

3.4 結果

実験結果を表 1 に示す。Seq2Seq [7] は例文による操作がなく、入力文のみから言い換え文を生成す

3) <https://github.com/timtadh/zhang-shasha>

表 2 Quora のテストデータにおいて生成された言い換え結果の例。入力文 (source) と例文 (exemplar) をイタリック体で、生成結果を立体で表記している。

| | | |
|-----|----------|--|
| (A) | source | <i>Is it possible to make a time machine ?</i> |
| | exemplar | <i>Can we survive a heart attack ?</i> |
| | Seq2Seq | Is it possible to make a time machine ? |
| | SCPN | Is it possible a time machine ? |
| | Ours | Can we create a time machine ? |
| (B) | source | <i>Why is space expanding ?</i> |
| | exemplar | <i>How and why is history repeating ?</i> |
| | Seq2Seq | Why is the universe expanding ? |
| | SCPN | Why and is expanding ? |
| | Ours | How and why is space expanding ? |
| (C) | source | <i>Do you think we are on the verge of World War III ?</i> |
| | exemplar | <i>How likely is a world economy crash ?</i> |
| | Seq2Seq | Is World War 3 more imminent than expected ? |
| | SCPN | Is World War a world III ? |
| | Ours | How likely is a World War III ? |
| (D) | source | <i>How can we prevent global warming ?</i> |
| | exemplar | <i>What should I do about women rights ?</i> |
| | Seq2Seq | How do I prevent global warming ? |
| | SCPN | How do we can prevent it warming global warmings ? |
| | Ours | What should I do to stop global warming ? |

る。SCPN [3] は例文の句構造に関する条件の下で入力文の言い換えを生成する。

参照文との類似度を示す BLEU スコアをみると、 $\beta = 1$ のときの提案手法が最も品質の高い言い換えを生成できていることがわかる。従来の Seq2Seq と比較して大幅なスコアの向上が見られ、例文による操作の元で望ましい言い換えが生成できていることが定量的に確認できた。

注目すべきことに、 β の値が小さすぎる場合も大きすぎる場合も BLEU スコアが大幅に低下する。 β が小さすぎるときは入力文との Sentence-BERT 類似度が小さくなっている。これは入力文の意味を生成結果が保てておらず、その結果、BLEU スコアが小さくなっていると考えられる。一方で、 β が大きすぎるときは例文との TED が大きく、LCS も小さくなっている。これは例文のスタイルが生成結果に反映されていないということを表し、その結果として、BLEU スコアが小さくなっていると考えられる。

定量評価に加えて、生成された結果を定性的に分析した。表 3.4 に $\beta = 1$ のときの生成結果を示す。提案手法では、 $\beta = 1$ のとき、入力文の意味を保ちつつ、例文のスタイルに似た表現を生成できていることが分かる。

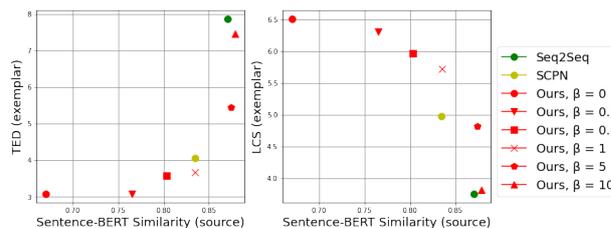


図 2 Sentence-BERT 類似度と TED(左) 及び LCS(右) の関係。

3.5 妥当性と可制御性のトレードオフ

ここでは、入力文の意味内容の保持と例文による操作性のトレードオフについて調べる。図 2 に、生成結果を入力文と比較した Sentence-BERT 類似度と、例文と比較した TED 及び LCS の関係を示す。

TED が低いほど例文の構文構造を反映していることとなるため、図の右下にある点がより良い結果となる。Seq2Seq の生成結果は、例文の情報を推論時に全く使っていないため、当然例文との構造的な距離は大きい。SCPN は例文の句構造を推論時に用いてそれに沿った言い換えを出力するため、TED のスコアが Seq2Seq と比較して小さくなっている。提案手法の結果は、 β の値が増加するにつれて図中左下から右上に推移している。 $\beta = 1$ の結果をみると、SCPN の結果と Sentence-BERT 類似度はほぼ同じであるにも関わらず、TED がより小さくなっている。すなわち、SCPN と比較してより良いトレードオフを達成していることが確認できる。

LCS は高いほど例文の言い回しを含む文が生成できていると言えるため、図の右上にある点がより良い結果となる。入力文との意味的な類似度を低下させず、LCS を高くできている。すなわち、言い換えとしての品質を落とさないうまま、語彙的にも例文により近い文を生成できている。

4 おわりに

本研究では、例文を用いた制御可能な言い換え生成の手法を提案した。文のスタイルに関する情報のみを抽出するために、相互情報量に基づく損失関数を導入した。評価用に、Quora データセットについて例文のアノテーションを行った。

実験の結果、提案手法によって入力文の意味を保持しつつ例文のスタイルを持つような言い換えが生成できていることを定量・定性的に確認した。

参考文献

- [1] Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual LSTM networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2923–2934, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.
- [2] Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3865–3878, Brussels, Belgium, October–November 2018. Association for Computational Linguistics.
- [3] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [4] Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5972–5984, Florence, Italy, July 2019. Association for Computational Linguistics.
- [5] Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. Improving disentangled text representation learning with information-theoretic guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7530–7541, Online, July 2020. Association for Computational Linguistics.
- [6] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. CLUB: A contrastive log-ratio upper bound of mutual information. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 of *Proceedings of Machine Learning Research*, pp. 1779–1788, Virtual, 13–18 Jul 2020. PMLR.
- [7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998–6008. Curran Associates, Inc., 2017.
- [8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
- [9] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [10] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.
- [11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [12] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60, Baltimore, Maryland, June 2014. Association for Computational Linguistics.
- [13] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

A 作成したテストデータ

表 3 に、本研究で評価のために作成した例文の一部を、入力文及び参照文と共に示す。

表 3 評価用に用意した Quora のテストデータの一部。作成した例文を太字で示している。

| | |
|-----------|---|
| source | Is the iPhone really more expensive ? Why or why not ? |
| reference | Why is iPhone so expensive ? |
| exemplar | Why is mathematics so hard ? |
| source | Why is UK better than USA ? |
| reference | Which country do you like the most , UK or USA ? Why ? |
| exemplar | Which sports do you like the most , soccer or basketball ? Why ? |
| source | What are the movies worth watching after getting high ? |
| reference | Which are the Best movies to watch after getting high ? |
| exemplar | Which are the best books to read after getting negative ? |
| source | How many songs can a 16 GB iPhone hold -LRB- approximately -RRB- ? |
| reference | How many songs does a 16 GB iPhone hold ? |
| exemplar | How many calories does a Dominos pizza have ? |
| source | Which are some of the most popular questions on Quora with high number of followers ? |
| reference | Which question on Quora has the most followers ? |
| exemplar | Which province in Canada needs the most immigrants ? |
| source | What are the best tips to study philosophy at university ? |
| reference | What are the best tips to study philosophy ? |
| exemplar | What are the best tips to save money ? |
| source | Why is space expanding ? |
| reference | How and why is space expanding ? |
| exemplar | How and why is history repeating ? |
| source | How can I download Tor browser ? |
| reference | How do I download the Tor browser ? |
| exemplar | How do I find the dipole moment ? |
| source | What is the best question anyone has ever asked you ? How did you answer ? |
| reference | What is the best question you have ever been asked ? |
| exemplar | What is the best joke you have ever heard ? |
| source | Is it possible to make a time machine ? |
| reference | Can we create a time machine ? |
| exemplar | Can we survive a heart attack ? |
| source | How can we make money from YouTube ? |
| reference | How can you make money on YouTube ? |
| exemplar | How can you find people on Instagram ? |
| source | Why China is threat to India ? |
| reference | Is China a threat to India ? |
| exemplar | Is Engine a competitor to Google ? |
| source | How do I get a Rhodes scholarship ? |
| reference | What should I do to get a Rhode 's scholarship ? |
| exemplar | What should I do to become an automobile engineer ? |
| source | What are some ways to make more time ? |
| reference | How do I make more time ? |
| exemplar | How do I get more clients ? |