Leveraging Customer Reviews for E-commerce Query Generation

Yen-Chieh Lien[1], Rongting Zhang[2], Maxwell Harper[2], Vanessa Murdock[2] and Chia-Jung Lee[2]

University of Massachusetts Amherst[1], Amazon[2]

Motivation

- Customer reviews contain diverse properties of products
  - Example:
    - Action Camera
      - "Underwater photo", "For kayaking recording"
    - Tent
      - "Strong zipper", "Compact"
- These descriptors uncover actual customers' **shopping consideration**
- **Goal**
  - Generate queries containing key properties about the target products using reviews
  - Generate queries without utilizing real queries and do not require any private, proprietary user data

Pilot Study

- We do an initial pilot study on actual queries and reviews.
- Findings
  - A subset of terms in reviews resembles queries.
  - The subset is composed of combinations of nouns, adjectives and participles.
  - Ex: Headphones
    - Nouns: "earbuds", "headset"
    - Adjectives: "wireless" or "comfortable"
    - Participles: "running" or "sleeping"
    - "Wireless sleeping headphone" is one of real queries
- For reviews, we focus on these 3 types of terms and phrases as candidates by the filtering with POS tagging.

Statistic-based Approach

- For a term t, an importance score $I_t^D = \dfrac{p(t, R_D)}{p(t, R_G)}$ is based on a product-specific review set R_D and a generic review set R_G
- $p(t, R)$ is computed by a frequency-based method.
- We consider bigram phrases by selection with threshold.
- To generalize the results, we additionally train a seq2seq generation model on pairs of reviews and phrases picked by the statistic-based approach.

Zero-shot Generation

- We aim to adapt text-to-text generation models trained on other domains.
- Although there are previous doc2query models which generate queries from corresponding relevant documents, there is a gap between E-commerce queries and queries in search engines and QA systems.
- To overcome the gap, we reformulate queries in MSMARCO by POS filtering and train a doc2query model on new queries.

Ensemble Approach

- Based on a set of candidate phrase C_p selected by statistical and zero-shot approaches, we further apply an ensemble approach to select the most representative terms to build queries.
- Scoring function

$$S_t = freq(t, C_p) \cdot log\left(\frac{|\{p' \in D\}|}{|\{p' \mid p' \in D, t \in C_{p'}\}|}\right)$$

  Which follows TFIDF intuition to find the most distinguishable terms in the candidate set.
- Given a desired query length n, we formulate the pseudo queries for a product by selecting all possible $\binom{k}{n}$ combinations from the top-k results in C_p

Experiment

- For generation, we use T5-base as the architecture to generate terms in Stats-s2s and Zero-shot Generation
- Dataset: 3 product types on Amazon.com
  - Headphones
  - Tents
  - Conditioners

	Headphone		Tent		Conditioner	
	Dev	Test	Dev	Test	Dev	Test
# of reviews	23,165	23,623	19,208	18,734	17,055	17,689
# of sentences	102,281	103,771	97,553	97,320	68,691	70,829

- Evaluation:
  - Intrinsic Similarity Evaluation
    - BLEU and METEOR with real queries
  - Extrinsic Retrieval Evaluation
    - Train a retrieval model on generated queries with weak supervision signals.
    - Fine-tune TinyBERT to retrieve product description.

Results

Similarity

	Headphone		Tent		Conditioner	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
YAKE	0.1014	0.1371	0.2794	0.2002	0.3143	0.1998
Doc2Query	0.1589	0.1667	0.3684	0.2145	0.4404	0.264
Stats-base	0.1743	0.2001	0.3294	0.2201	0.4048	0.2723
Stats-s2s	0.1838	0.2004	0.321	0.2189	0.3931	0.2641
Ensemble	**0.2106***	**0.2024**	**0.394***	**0.2334***	**0.5047***	**0.2956***
Examples	noise cancelling headphone truck driver headphone hearing aids headphone		lightweight tent alps backpacking tent air mattresses queen tent		detangling conditioner shea moisture conditioner dry hair conditioner	

Retrieval Performance

	Headphone			Tent			Conditioner		
	MRR	P@1	P@10	MRR	P@1	P@10	MRR	P@1	P@10
BM25	0.28	0.19	0.06	0.43	0.29	0.11	0.56	0.47	0.14
YAKE	0.23	0.11	0.07	0.46	0.34	0.11	0.54	0.43	0.14
Doc2Query	0.28	0.18	**0.08**	**0.49**	**0.40**	0.12	0.58	**0.49**	0.15
Stats-base	0.28	0.16	0.07	0.44	0.29	0.12	0.54	0.42	0.15
Stats-s2s	0.27	0.17	0.07	0.44	0.32	0.12	0.56	0.46	**0.16**
Ensemble	**0.29**	**0.20**	0.07	0.46	0.33	**0.13**	**0.59**	0.48	0.15