Central Intention Identification for Natural Language Search Query in E-Commerce

Xusheng Luo*
Alibaba Group
lxs140564@alibaba-inc.com

Yu Gong*
Alibaba Group
gongyu.gy@alibaba-inc.com

Xi Chen
Alibaba Group
gongda.cx@taobao.com

ABSTRACT

This paper is a preliminary work, which studies the problem of finding central intention of natural language queries with multiple intention terms in e-commerce search. We believe it is a new and interesting topic since natural language based e-commercial search is still very young currently. We propose a neural network model with bi-LSTM and attention mechanism, aiming to find the semantic relatedness between natural language context words and central intention term. Initial experimental result reports that our model outperforms baseline method and shows a positive and important gain brought by a deep network model, comparing to rule based approach.

KEYWORDS

Query Intent & Understanding, Natural Language Query

ACM Reference Format:
Xusheng Luo[1], Yu Gong[1], and Xi Chen. 2018. Central Intention Identification for Natural Language Search Query in E-Commerce. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR 2018 eCom).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As the AI technologies develop rapidly, the services provided by e-commerce companies become more and more intelligent. One inevitable tendency, different from earlier online shopping experiences, is that customers will be able to use natural language instead of key words when searching for the products they want to buy. For example, customers can ask the online shopping search engine: *"I would like to buy a red fashionable short dress under 200 dollars."* instead of type key words like *"short dress, red, fashion, cheaper than 200"*. Comparing to key words, using natural language is a more comfortable way for people to go online shopping since it is the way we communicate with each other in daily life.

The very first step for search engine to understand user query is to identify the query intention. In the case of the previous query, that means to know it is a dress the customer want to buy. Here *"short dress"* is an **intention term** (a term can be a word or a phrase), which indicates the e-commercial category of a product. The recognition of intention term is usually performed by a module called *Query Tagging*, which is similar to *Named Entity Recognition*[7, 8]. Sometimes, there will be more than one intention term within a single natural language user query such as *"I want to buy a pair of stockings for my new short dress."* (Figure 1), where *"stockings"* and *"short dress"* are both intention terms, which makes it more difficult for machines to identify the true intention of this query (*stockings* rather than *short dress*). Cases like this are not rare in natural language queries, as we found that there are around 20% of voice queries (voice query is more likely to be in natural language form since people tend to use natural language as they speak), which contains more than one intention term after query tagging. This motivates us to identify the central intention of a user query among all intention terms so that our machine can better understand search queries.

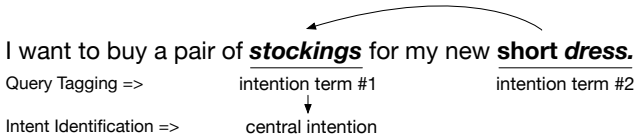

Figure 1: Example query with multiple intention terms

Multiple intention terms in one query is also common in nowadays key-word based e-commerce search. However, those queries are tend to be short and in fix-pattern such as "*laptop backpack*", where "*laptop*" and "*backpack*" are both intention terms and we all know the true intention is backpack. In general, we will analyze the query log and corresponding click log to find out what products the users are clicking and viewing after type the query in the search box and then we construct a multi-terms→ central-term map offline. Thus, next time when we see a query with multiple short intention terms, we can easily know the actual intention by looking up the map. However, this method is not helpful and limited when dealing with natural language queries, which are much longer and more complicated. With natural language interaction grows, there will be more and more new intention combinations.

We believe a deep model can work more effectively and hence we dig a little deeper towards this topic and make the following contributions:

- We propose a new and interesting topic when e-commerce search meet natural language queries with multiple intention terms. And we attempt to identify the central intention so that search engine can better understand queries.
- We present a neural network with bi-LSTM and attention mechanism to effectively capture the rich semantic relatedness between context words and intention term in user query; Based on that, we identify the central intention.

*Equal contribution.

In: J. Degenhardt, G. Di Fabbrizio, S. Kallumadi, M. Kumar, Y.-C. Lin, A. Trotman, H. Zhao (eds.): Proceedings of the SIGIR 2018 eCom workshop, 12 July, 2018, Ann Arbor, Michigan, USA, published at http://ceur-ws.org

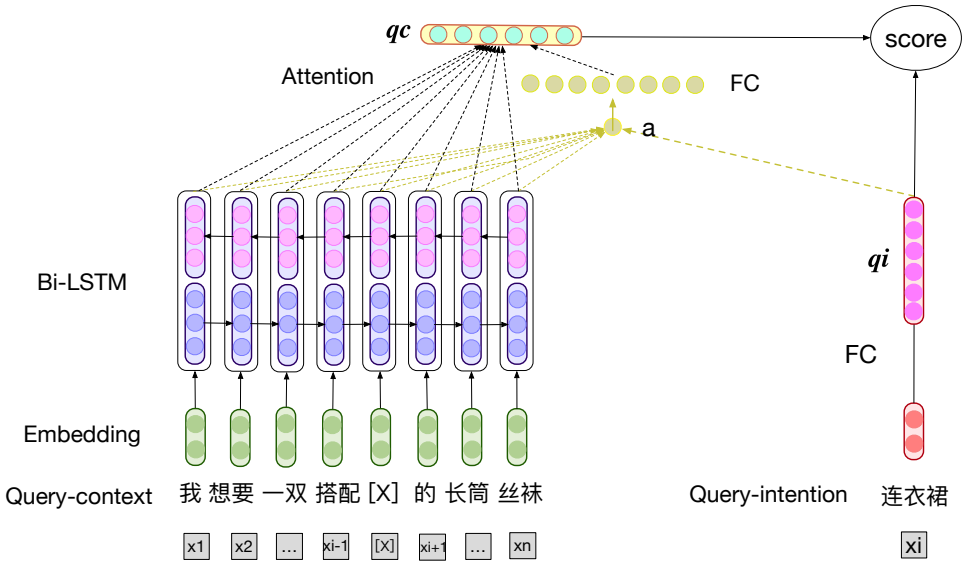

Figure 2: Overview of proposed model

- We try to construct a dataset for experiments and find an alternative way to train our model although there is no direct ground truth available;
- The proposed neural network model outperforms baseline method which is based on dependency parsing. Future work is ongoing towards data collection, model upgrade, etc.

2 APPROACH

The central intention identification task is defined as follows. The input query is a sequence of word terms $q = (x_1, x_2, \ldots, x_n)$, with at least two intention terms. A term x_i can be a word or a phrase. Our task is to output only one intention term x_i as the central intention, while other intention terms modify the central intention. Defined in this way, we actually make a hypothesis that each search query contains only one actual goal product. We do not consider queries where a user ask for two or more items at the same time.

Now, we describe our neural network model and baseline method for query intention identification. Figure 2 gives a general view of the proposed neural network model. Given the context words of a query $qc = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, which is the terms left after taking the intention term way , together with the intention term $qi = x_i$, our model will output a score $score(qc, qi)$, measuring the compatibility between them.

2.1 Term Embedding

Typically, a term contains up to three words, thus we simply represent it as the average embedding of the words it contains. We train word embeddings and term embeddings on large text corpus. Embeddings are fed to model as input and will be updated during training.

2.2 Bi-LSTM with Attention

Recurrent neural networks (RNNs) are a powerful family of neural networks designed for sequential data and have shown great promise in many NLP tasks. RNNs take a sequence of vector $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ and return another sequence $(\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n)$ that represents the hidden state information about the sequence at each time step in the input. In theory, RNNs can learn long dependencies but in practice they seem to be biased towards their most recent inputs of the sequence. Thus, LSTMs [3] are proposed and they have shown great capabilities to capture long-range dependencies.

To encode the query context, we first look up an embedding matrix $E_x \in \mathbf{R}^{d \times v}$ to get the term embeddings $\boldsymbol{q} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{i-1}, [\boldsymbol{X}], \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_n)$. Here, $[\boldsymbol{X}]$ is a wildcard embedding to indicate the position of intention term in the query. d denotes the dimension of the embeddings and v denotes the vocabulary size of natural language words. Then, the embeddings are fed into a bidirectional LSTM networks. If we use unidirectional LSTM, the outcome of current word is only based on the words before it so the information of the words after it is totally lost. To avoid this, we use bi-LSTM which consists a forward network handles the query from left to right and a backward network does in the reverse order. Therefore, we get two hidden state sequences, $(\overrightarrow{\boldsymbol{h}_1}, \overrightarrow{\boldsymbol{h}_2}, \ldots, \overrightarrow{\boldsymbol{h}_n})$ from forward network and $(\overleftarrow{\boldsymbol{h}_1}, \overleftarrow{\boldsymbol{h}_2}, \ldots, \overleftarrow{\boldsymbol{h}_n})$ from backward network. We concatenate the forward hidden state of each word with corresponding backward hidden state, resulting in a representation $\boldsymbol{H}_i = [\overrightarrow{\boldsymbol{h}_i}; \overleftarrow{\boldsymbol{h}_i}] \in \mathbf{R}^{k \times 1}$. Thus, we obtain the representation of each word in the query context.

Attention mechanisms [1, 4] have become an integral part of sequence modeling and transduction models in various NLP tasks, which allows better understanding sequential data. Based on our assumption, different intention terms should have different attention towards the same query, The extent of attention can be calculated by the relatedness between each word representation $\boldsymbol{H}_i$ and an intention embedding $\boldsymbol{qi}$, where $\boldsymbol{qi} = W_i^T \boldsymbol{x}_i$ and $W_i \in \mathbf{R}^{k \times 1}$. We propose the following formulas to calculate the attention weights.

$$a_i = \frac{\exp(w_i)}{\sum_{i=1}^{n} \exp(w_i)} \tag{1}$$

$$w_i = W_a^T(tanh[\boldsymbol{H}_i; \boldsymbol{qi}]) + b \quad (2)$$

Here, a_i denotes the attention weight of the ith term in the query context, in terms of intention $\boldsymbol{e}$, where $\boldsymbol{qi}$ is a hidden representation of one intention term. n is the length of the query. $W_a \in \mathbf{R}^{2k\times 1}$ is an intermediate matrix and b is an offset value. These two parameters are randomly initialized and updated during training. Subsequently, the attention weights a (Figure 2) are employed to calculate a weighted sum of the query terms, resulting in a semantic representation $\boldsymbol{qc}$ which represents the query context, according to the specific intention term.

$$\boldsymbol{qc} = \sum_{i=1}^{n} a_i \boldsymbol{H}_i \quad (3)$$

Thus, the final output score which is regarded as a measurement of the compatibility of query context qc and intention term qi can be calculated as follows.

$$S(qc, qi) = \boldsymbol{qc} \cdot \boldsymbol{qi} \quad (4)$$

Therefore, we use intention term qi as attention query to guide the model weighting each context term differently, aiming to better justify compatibility between current intention term and the whole user query. When we consider an intention term, we will re-read the query to find out which part of the query should be more focused (handling attention). We believe that this attention mechanism is beneficial for the system to better understand the query with the help of the intention term, and leads to a performance improvement.

2.3 Training and Prediction

Since there is no ground truth currently and it is extremely costly to annotate the central intention for user queries with multiple intention terms. Thus, we choose those natural language queries with only one intention term as our training data. We believe it is a reasonable degeneration since our goal is to dig the semantic relationship between natural language context words and some target intention term. This relatedness can be learned from single-intention queries and then apply to multi-intention queries. We use a dynamic programming max-matching algorithm to match terms in the query to an existing dictionary containing all the intention terms such as "连衣裙 (Dress)" and "丝袜 (Stocking)". We only keep queries with only one exactly matched intention term. After this "query tagging" step, we can identify the intention term and regard <query context, intention term> pair in each query as a positive sample. Then we randomly choose some unrelated intention terms as negative samples. We use hinge loss to train the model:

$$loss = \sum_{qi' \in N} \max(0, 1 - score(qc, qi) + score(qc, qi')) \quad (5)$$

Where qc is the query context, qi is the positive query intention and qi' is the corrupted query intention term from negative samples N. The function $score$ represents the model output.

We evaluate our model on a dataset labeled by human. Each query in our testing set contains more than one intention term. When testing a query with one intention term of it, we take away the intention term and feed the rest of query, i.e. query context into model. The intention term with highest output score is considered as the central intention.

2.4 Baseline

We use a rule based method as our baseline method. We perform dependency parsing on the input user query. A dependency parser analyzes the grammatical structure of a sentence, establishing relationships between "head" words and words which modify those heads. Among all the intention terms, we choose the one at highest position in the parsing tree as the central intention. As shown in Figure 3, we use an internal e-commercial query parser as our baseline method. In this example of query *"I want to buy a pair of stockings for my new short dress (*我 想要 一双 搭配 连衣裙 的 长筒 丝袜*)"*, "丝袜 (Stocking)" is at a higher position than "连衣裙 (Dress)" in the parsing tree. Thus we choose "丝袜 (Stocking)" as the central intention of this query.

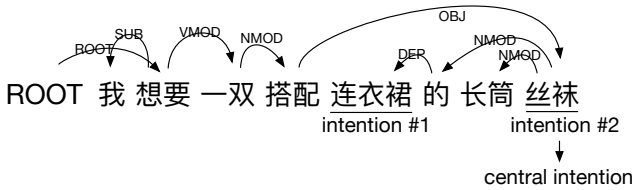

Figure 3: Dependency parsing example of query with multiple intention terms

3 EXPERIMENTS

3.1 Dataset

We train our model on 10,000 single intention Chinese voice search queries and test on two datasets. We filter out queries whose length is shorter than 10 words. One is single-intention query set. We construct it by corrupting the intention term of 10,000 single-intention queries with randomly chosen intention terms. The other one is multi-intention query set. It contains 300 multi-intention search queries, which consists of 150 2-intentions queries, 100 3-intentions queries and 50 4-intentions queries. The size of this dataset is limited since it need a lot of human labeling efforts. We use an e-commerce query tagging tool to preprocess all the training and testing queries.

3.2 Implement Details

We pre-train word and term embeddings on a large Chinese e-commerce corpus. This corpus comes from a module in Chinese e-commerce giant **Taobao*** named "有好货"†, which is written by online merchants. We use word2vec [5] CBOW model with context window size 5, negative sampling size 5, iteration steps 5 and hierarchical softmax 1. The size of pre-trained word embeddings is set to 200. For Out-Of-Vocabulary (OOV) words, embeddings are initialized as zero. All embeddings are updated during training. We use an e-commerce Chinese word segmentation tool for word segmentation.

*https://www.taobao.com/
†https://h5.m.taobao.com/lanlan/index.html

Table 1: Real cases of central intention identification

#1	我 想要 穿着 显瘦 only 牌子 的 连衣裙 最好 是 能 搭配 耳坠 的。 I want to buy an ONLY-brand thin-looking dress which is suitable for earrings.
#2	汽车 上面 用 的 那个 小的 吸尘器 有没有 的? Do you have small vacuum cleaner for cars?
#3	黄色 T恤衫 前面 就是 有 2个 耳坠 那种。 Yellow T-shit with a pair of earrings in the front.
#4	打 篮球 踢 足球 都 可以 穿 的 nike 鞋，没有 鞋带。 Nike shoes without shoelace, for both basketball and soccer.

Table 2: Accuracies on Single-Intention Queries

Approach	Acc
Model (- attention)	0.803
Model (+ attention)	**0.813**

Table 3: Accuracies on Multi-Intention Queries

Approach	2-intents	3-intents	4-intents
Baseline	0.60	0.54	0.32
Model (- att)	0.67	0.66	0.40
Model (+ att)	**0.68**	**0.67**	**0.46**

For recurrent neural network component in our system, we used a two-layers LSTM network with unit size 512. All natural language queries are padded to a maximum sentence length of 30. We use Adam optimizer, and the learning rate is initialized with 0.01.

For baseline method, we use an internal e-commercial query parser to do dependency parsing. This parser is similar as the famous Stanford Dependency Parser [2] but is optimized specially for e-commercial scenario.

3.3 End-to-end Result

Now we report the experimental results as follows. First we show the accuracy on single-intention query set. The goal of this experiment is to evaluate the training quality explicitly. The model has to identify the correct intention terms from the corrupted ones. As shown in Table 2, it achieves 0.813 in accuracy. Considering the user queries always contain a lot of noises, this number shows power of our model at learning semantic relations between natural language query context and query intention. Besides, the result proves that attention mechanism is effective in this task.

In the experiment on multi-intention query set, we assigned three human annotators to judge whether the model output is correct, i.e. whether the intention term with the highest score is the central query intention. Based on majority voting, we calculate the accuracy in Table 3. Our model with attention mechanism outperforms baseline method and the one without attention mechanism by up to 13%. Baseline method based on dependency parsing suffers from bad performance on short sentence, since search queries in e-commerce tend to be short and less grammatical. On the other hand, deep neural network model shows potential to learn rich semantic relatedness between context words and intention terms regardless of sentence size.

3.4 Case Study & Error Analysis

In Table 1, we show some real cases of intention identification of search queries. In each case, the underlined terms are the intention terms recognized by query tagging and the red-colored term is the central intention identified by model. Take the first query "我 想 要 穿着 显瘦 only 牌子 的 连衣裙 最好 是 能 搭配 耳坠 的。" as example, the baseline method using e-commercial dependency parsing regards "耳坠 (Earring)" as root thus discards terms including "连衣裙 (Dress)" which is actually the true central intention. Our model can output the correct intention after seeing enough semantic information in training data and believes "穿着","显瘦", "only" are more likely to describe "连衣裙 (Dress)" rather than "耳坠（Earring）".

Since this work is in the preliminary stage, we actually find several problems in our experiments. First, the quality of queries are not as high as what we expect. Currently the main interactive way between a customer and online e-commerce search engine is still based on key words. Thus, at current stage, it is hard to get enough high-quality natural language query log. That is why we choose voice queries as the source of natural language queries. However, the precision of speech recognition becomes a problem, especially when people say something very domain-specific.

Second, the habit of using key words to do online shopping can not be easily changed. Within voice queries, there still exists quite a few queries which are some combination of several similar key words which actually mean the same product. However, the goal of our model is to dig the semantic relatedness between query words and intention terms. This idea can not hold if the terms of a query are not in natural order or the query is not even a natural language sentence.

Besides, we also find some cases where simple rule or patterns may works better than deep models. For example, the central intention of "连衣裙上面的绿色纽扣(Green buttons of dress)" is "纽扣 (button)" but it becomes "连衣裙(dress)" if we change only one word to "连衣裙上面有绿色纽扣 (Dress with green buttons)". Although these cases are rare and extreme, it is indeed a challenge for our model. Maybe some syntactic and rule based features should be fed to model somehow to help it deal with this problem.

4 FUTURE WORK

In this paper, we explore the area where e-commerce search queries are in natural language form and multiple intention terms are appearing together in the same query. We proposed a deep neural network to identify the true intention and made some delighted progress comparing to rule based method. In the future, we will try to construct a larger and cleaner dataset for both training and testing and make it public. This work is a preliminary attempt currently and it need to be further improved such as adding syntactical and rule based features to the model in the future.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 740–750.

[3] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[4] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[6] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

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

[8] Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 142–147.