# Deep Search Query Intent Understanding

Xiaowei Liu, Weiwei Guo, Huiji Gao, Bo Long
LinkedIn, Mountain View, California
{xwli,wguo,hgao,blong}@linkedin.com

## ABSTRACT

Understanding a user's query intent behind a search is critical for modern search engine success. Accurate query intent prediction allows the search engine to better serve the user's need by rendering results from more relevant categories. This paper aims to provide a comprehensive learning framework for modeling query intent under different stages of a search. We focus on the design for 1) predicting users' intents as they type in queries on-the-fly in typeahead search using character-level models; and 2) accurate word-level intent prediction models for complete queries. Various deep learning components for query text understanding are experimented. Offline evaluation and online A/B test experiments show that the proposed methods are effective in understanding query intent and efficient to scale for online search systems.

## KEYWORDS

Query Intent, Query Classification, Natural Language Processing, Deep Learning

## 1 INTRODUCTION

Modern search engines provide search services specialized across various domains (*e.g.*, news, books, and travel). Users come to a search engine to look for information with different possible intents: choosing favorite restaurants, checking opening hours, or restaurant addresses on Yelp; searching for people, finding job opportunities, looking for company information on LinkedIn, etc. Understanding the intent of a searcher is crucial to the success of search systems. Queries contain rich textual information provided explicitly by the searcher, hence a strong indicator to the searcherâĂŹs intent. Understanding the underlying searcher intent from a query, is referred to the task of query intent modeling.

Query intent is an important component in the search engine ecosystem [12, 16, 20]. As shown in Figure 1, when the user starts typing a query, the intent is predicted based on the incomplete character sequence; when the user finishes typing the whole query, a more accurate intent is predicted based on the completed query. Understanding the user intent accurately allows the search engine to trigger corresponding vertical searches, as well as to better rank the retrieved documents based on the intent [2], so that users do not have to refine their searches by explicitly navigating through the different facets in the search engine.

Traditional methods rely on bag-of-words representation and rule based features to perform intent classification [2, 3]. Recently, deep learning based models [10] show significant improvement, which can handle similar words/word sense disambiguation well. However, developing deep learning based query intent models for productions requires considering several challenges. Firstly, production models have very strict latency requirements, and the whole process needs to be finished within tens of milliseconds. Secondly, queries, usually with two or three words in a complete query or several characters in a incomplete one, have limited contexts.

This paper proposes a practical deep learning framework to tackle the two challenges, with the goal of improving LinkedIn's commercial search engine. Two search result blending components were identified where query intent is useful: incomplete query intent for typeahead blending, and complete query intent for SERP blending (search engine result page).

The common part of both systems is to use query intent to assist the ranking of retrieved documents of different types. Meanwhile, the two products have their unique challenges. Typeahead blending has strong latency requirements; the input is an incomplete query (a sequence of characters); and it is okay to return a fuzzy prediction, since users will continue to type the whole query if he/she does not find the results relevant. On the other hand, SERP blending has less latency constraint compared to typeahead but a higher accuracy requirement as it directly affects the search result page.

Based on the characteristics of production applications, we propose different solutions. For typeahead blending, character-level query representation is used as the resulting models are compact in terms of the number of parameters. Meanwhile, it can handle multilinguality well due to the small vocabulary size. For SERP blending, the complete query intent model is word level. Since accuracy is a high standard, BERT is explored to extract query representations which lead to a more robust model.

This paper is motivated by tackling the challenges in query intent prediction, while satisfying production needs in order to achieve real-world impact in search systems. The major contributions are:

- Developed a practical deep query intent understanding framework that can adapt to various product scenarios in industry. It allows for fast and compact models suitable for online systems, with the ability of incorporating traditional features that enables more accurate predictions.
- Developed and deployed character-level deep models to production that is scalable for incomplete query intent prediction. In addition, we propose a multilingual that incorporates language features which is accurate and easier to maintain than traditional per-language models.
- Developed and deployed a BERT based model to production for complete query intent prediction. To our best knowledge, this is the first reported BERT model for query intent understanding in real-word search engines.
- Conducted comprehensive offline experiments and online A/B tests on various neural network structures for deep query intent understanding, as well as in-depth investigation on token granularity (word-level, character-level), DNN components (CNN, LSTM, BERT), and multilingual models, with practical lessons summarized.

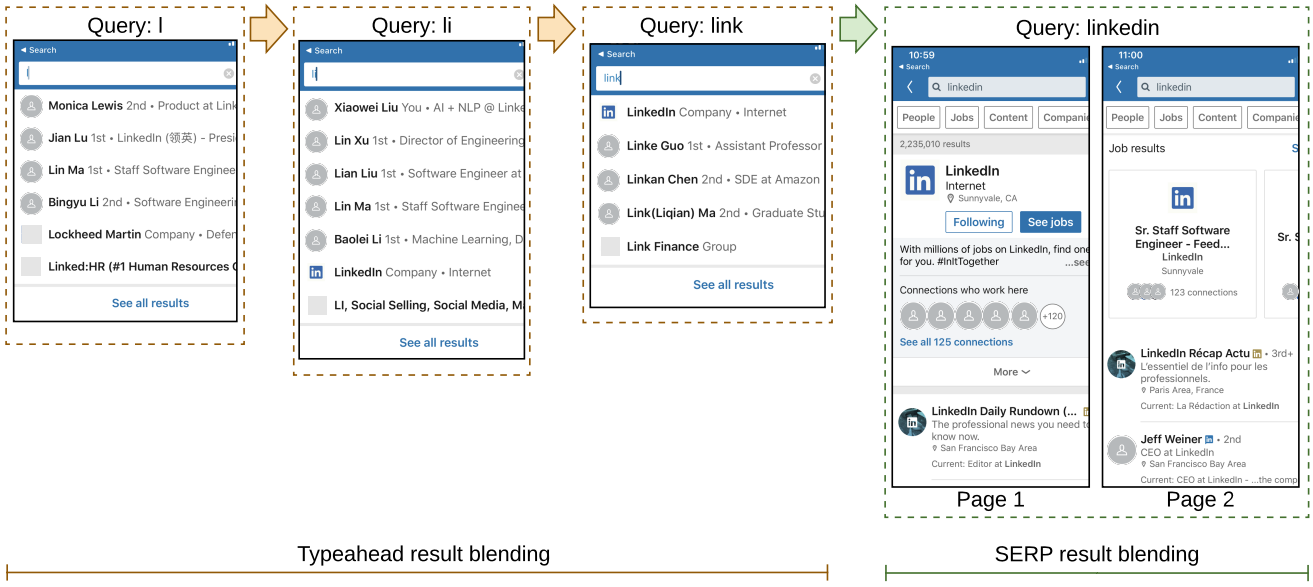


**Figure 1: Query intent in search engines for incomplete queries (typeahead blending) and complete queries (SERP blending).**

## 2 QUERY INTENT UNDERSTANDING AT LINKEDIN

LinkedIn search hosts many different types of documents, *e.g.*, user profiles, job posts, user feeds, etc. When a user issues a query without specifying the document type they are interested in, identifying the intent is crucial to retrieve relevant documents and provide high-quality user experience. At LinkedIn, we define the query intent as the document type.

Query intent is important for result blending [12]: (1) When an intent is not presented in the query, the corresponding vertical search may not be triggered. (2) For the documents retrieved from the triggered vertical searches, a result blending algorithm will rank the documents based on detected intents and other features. In this section, we present two productions where query intent is an important feature for the blending algorithm, followed by how query intent is used in the blending algorithm.

### 2.1 Query Intent in Typeahead Blending

When a user starts typing, the query intent is detected and used in typeahead blending. At LinkedIn, the typeahead product directly displays document snapshots from multiple vertical searches, which is different from traditional query auto completion that only generates query strings. The left three snapshots in Figure 1 shows an example of typeahead blending results at LinkedIn. The example assumes the user is searching for the company "LinkedIn". Blended results are rendered as soon as the user typed one letter "l". Next, given the query prefix "li", the intent prediction has a tendency towards a people result type and many people profiles are ranked higher than company or groups results. After the user types "link" in the third picture, the company result LinkedIn is ranked first.

Query intent for typeahead blending is challenging given that the queries are often incomplete and contains only several letters. In addition, for every keystroke, the system needs to retrieve the documents from different vertical searches and blend the results. It means the query intent models will be called frequently, and each run should be finished within a short time.

### 2.2 Query Intent in SERP Blending

SERP blending is a more common component in search engines than typeahead blending. When a user finishes typing and hits "search", a complete query is issued; the query intent is identified and used for retrieved document blending. The right most block in Figure 1 shows SERP blending results for a complete query "linkedin", including company pages, people profiles, job posts, etc.

Compared to typeahead blending, query intent in SERP blending has a larger latency buffer. Queries contain complete words, however, it still suffers from limited contexts: only several words in a query for intent prediction.

### 2.3 Retrieved Document Blending Systems

Both typeahead blending and SERP blending systems follow a similar design. Multiple features are generated for blending/ranking the retrieved documents: (1) Probability over each intent that is based on query texts and user behaviors (the query intent model output); (2) matching features between the query and retrieved documents; (3) personalized and contextualized features.

In the rest of this paper, we focus on how to generate the query intent probability for typeahead blending and SERP blending.

## 3 PROBLEM DEFINITION

Both incomplete query intent and complete query intent are essentially classification tasks. Without loss of generosity, given a user id $u \in \mathbb{U}$ and a query string $q \in \mathbb{Q}$, the goal is to learn a function $\gamma$ predict the predefined intent $i$ in the finite number of intent classes $\mathbb{I} = \{i_1, i_2, ...i_n\}$:

$$\gamma : \langle \mathbb{Q}, \mathbb{U} \rangle \rightarrow \mathbb{I}$$

For the two tasks, incomplete/complete query intent, the intent class sets are slightly different, due to the design of products. As shown in the next section, deep learning models are applied to query strings; the user ids are used to generate personalized features.

## 4 DEEP QUERY INTENT UNDERSTANDING

In this section, we introduce the proposed intent modeling framework, as well as the detailed design of two applications: incomplete query intent model and complete query intent model.

### 4.1 Product Requirements

As shown in Figure 1, there are two result blending products that rely on query intent: typeahead blending and SERP blending. These two products pose several requirements of the query intent models.

Typeahead blending has a strict latency standard: for every keystroke, the model needs to return the results. Meanwhile, it does not require very high accuracy since the prediction becomes more precise as it receives more characters. On the other hand, SERP blending has more latency buffer, and it requires high accuracy of query intent, otherwise users might abandon the search.

### 4.2 Intent Modeling Framework

Driven by the product requirements, we design a framework for query intent understanding. The overall architecture is in Figure 2.

*4.2.1 Input Representation.* The input to the model is represented in a sequence of embeddings. Two granularity choices are provided: character- and word-level embeddings to support incomplete and complete queries, respectively.

*4.2.2 Deep Modules.* In this framework, several popular text encoding methods are provided to generate query embeddings. This enables good flexibility to adapt the framework to various product scenarios under different latency / accuracy constraints.

**CNN** is powerful at extracting local ngram features in a sequence [15, 17]. The input to the CNN is a sequence of token embeddings, i.e. the embedding matrix. The 1-dimensional convolution layers could involve multiple filters of different heights. The width of the CNN filters is always the same size as the embedding dimension, while the height of the filters could varyâĂŤit represents word or character n-grams covered by the filters. Max-pooling over time is done after each convolution layer.

Compared with CNN, the long distance dependencies can be better captured by **LSTM** [11], especially the character sequence. Bi-directional LSTM [29] is used to model the sequence information from both forward and backward. The last hidden state of both layers are concatenated together to form the output layer.

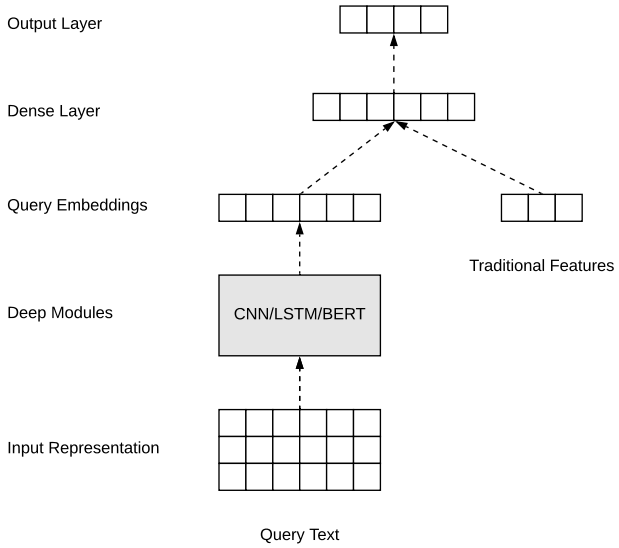


**Figure 2: Deep query intent understanding framework.**

**BERT** [9] uses self attention [26] to explicitly integrate contextual word meaning into the target word, hence it is better at word sense disambiguation. Meanwhile, the pretraining enables using a large amount of unsupervised data. Given a query, BERT takes a sequence of tokens as input and output the contextualized representation of the sequence. A special token `[CLS]` at the beginning of each sequence models the representation of the entire sequence and is used for classification tasks.

*4.2.3 Traditional Features.* Traditional features are hand-crafted features, which are powerful for capturing contextual information that is complementary to the deep textual features. There are various types of traditional features that can be considered in production, such as language features, user profile / behavioral features. These features are especially important for enhancing the limited context for short queries. In a wide-and-deep fashion [7], traditional features are concatenated with the query embeddings, and then fed to a dense layer to get non-linear interactions among the features.

### 4.3 Incomplete Query Intent Modeling

In typeahead search, users usually type a query prefix, and select the results from the drop-down bar. In this case, a large number of incomplete words are generated, which are recognized as out-of-vocabulary words. This motivates us to design the incomplete query intent classification with character-level representations. The character-level models have additional benefits: (1) it is more **robust** to spelling errors, compared to word-level models where words with wrong spelling will be out of vocabulary; (2) the resulting model is **compact** (1.4 Megabytes with 500 characters), as the character vocabulary size is small.

Due to the ability of capturing long range dependency information, LSTM is best suited for this problem, as the sequence could be over 10 characters. In contrast, CNN captures the n-gram letter patterns (e.g., tri-letters). However, it does not keep track of what