Table 8: Performance comparison of whether to incorporate traditional features.

Model	Trad Features	Accuracy	F1 (*people*)	F1 (*job*)
LR-BOW	-	-	-	-
CNN	False	+1.27%	−1.56%	+5.27%
CNN	True	+6.40%	+1.36%	+17.35%
LSTM	False	+1.26%	−1.54%	+5.15%
LSTM	True	+6.69%	+1.47%	+17.99%
LiBERT	False	+1.90%	−1.14%	−6.16%
LiBERT	True	+8.35%	+2.46%	+20.60%

Table 9: Online performance comparison for word-level complete query intent model vs production baseline in SERP blending.

Model	Metrics	% lift
CNN-word	CTR@5	neutral
	CTR@5 for job results	+0.43%
	SAT click	neutral
LiBERT (vs. CNN-word)	CTR@5	+0.17%
	CTR@5 for job results	+0.51%
	SAT click	+1.36%

Table 9 shows the online metrics gain of CNN model over logistic regression. It improves two job search related metrics: the overall click-through rate of job postings at top 1 to 5 positions in the search result page (CTR@5, which is the CTR at position 1 to 5 for Job Results), and total number of job search result viewers (Job Viewers). This is consistent with the significantly improved F1 score of job intent (offline results in Table 7).

Later, we conducted online experiments of LiBERT over CNN (Table 9). Since LiBERT can further improve the accuracy of all intents, more online metrics are significant, including SAT clicks (the satisfactory clicks on documents with a long dwell time), and CTR@5 over all documents.

5.4 Scalability

In the online system, latency at 99 percentile (P99) is the major challenge for productionization. For query intent, offline precomputation of the frequent queries does not help, since the P99 queries are usually rare, and there are large number of unseen queries each day at LinkedIn search. The latency comparison of the character-level models (CNN-char and LSTM-char) for incomplete queries is shown in Table 10. The P99 latency numbers are computed on Intel(R) Xeon(R) 8-core CPU E5-2620 v4 @ 2.10GHz machine with 64-GB memory.

In terms of complete queries, the CNN/LSTM models have even less latency, since number of words are generally smaller than number of characters. For BERT models, the latency increases significantly. It is worth noting that by pre-training a smaller BERT model on in-domain data, we are able to reduce the latency from 53ms (BERT-Base) to 15ms (LiBERT) without hurting the relevance performance, which meets the search blending latency requirement.

Table 10: Latency and model size comparison on different query intent models at 99 percentile.

	Model	#Params	P99 Latency
Incomplete Query Intent	Tri-letter	-	-
	CNN-en	123k	$+0.38ms$
	LSTM-en	379k	$+2.49ms$
	LSTM-i18n-embed	1M	$+2.72ms$
Complete Query Intent	LR-BOW	-	-
	CNN-word	6.5M	$+0.45ms$
	LSTM-word	6.5M	$+0.96ms$
	BERT-Base [9]	110M	$+53.2ms$
	LiBERT	10M	$+15.0ms$

6 LESSONS LEARNED

We would like to share our experiences and lessons learned through our journey of leveraging state-of-the-art technologies and deploying the deep models for query intent prediction to online production systems. We hope they could benefit others from the industry who are facing similar tasks.

Online Effectiveness/Efficiency Trade-off. Our experiments with BERT-based models show that pre-training with *in-domain* data improved fine-tuning performances compared to the off-the-shelf models significantly. Aside from relevance gains that BERT brings us, latency is a big issue for productionizing these large models. We found that *reducing the pre-trained model sizes* significantly reduces the inference latency while providing accurate predictions.

For the incomplete query intent, we cannot deploy complicated models due to the latency constraint in typeahead search. In order to maximize the relevance performance within a strict latency constraint, we investigated the characteristics of LSTM and CNN models w.r.t. effectiveness and efficiency. LSTMs give *superior relevance performance* for its ability to capture long range information in sequences, especially when the sequence is incomplete, whereas CNNs are *much faster* in inference speed by design. We were able to deploy compact LSTM models to the typeahead product while achieving an optimal prediction performance with inference latency being tolerable in production.

Token Granularity. In the incomplete query intent models for the typeahead search product, the choice of character-level models is effective in modeling the character sequences in the incomplete queries. It's worth mentioning that these models require a smaller vocabulary size compared to word or subword level models, which result in much more *compact models* suitable for online serving.

The character-level granularity allows for combining vocabularies from different languages since many languages share similar characters. The design of multilingual models could further benefit online systems for *robust predictions* across multiple markets yet *easier maintenance* than per-language models.

Combining Traditional Features. Traditional features are informative for promoting deep query intent understanding. From our experiments we show that simply discarding the handcrafted features and user personalized features and replacing them with deep learning models hurts the relevance performances. We find that incorporating the traditional features in a wide-end-deep fashion is crucial in successful intent prediction.

7 RELATED WORK

Query intent prediction has been an important topic in modern search engines [12, 16, 20, 25]. We firstly introduce traditional methods, then show how deep learning models are applied to this problem. Finally, we discuss the existing works regarding incomplete query intent modeling.

7.1 Traditional Methods

Early query intent works use rule based methods [13, 24], as a high precision low recall strategy. However, it is hard to maintain as the rules become more complicated, and the recall is low.

More recently, statistical models have shown significant improvements in unsupervised (query clustering) [1, 8, 23] or supervised (query classification) approaches. In supervised methods, common features include unigram, language model scores, lexicon matching features, etc [2, 3, 5].

7.2 Deep Learning for Query Intent Classification

Deep learning approaches have shown significant improvement in text classification tasks, where multiple CNN and LSTM based methods [15, 17, 19, 21, 28] have been proposed. The closest related work is an CNN-based approach [10] for classifying types of web search queries, with similar network architecture as in [17]. It is worth noting that although the most recent technology on [9] has shown promising improvement for text understanding including intent classification [6], the performance on short text such as search queries is not clear.

To our best knowledge, we have yet to see BERT models applied to real-world search systems for query intent prediction.

7.3 Incomplete Query Intent

There have been character-level models for classic NLP tasks such as text classification and language modeling [14, 18, 27]. However, we have yet to find previous work of character-level models on **incomplete** query intent prediction in search productions. We further extended our approach to a multilingual model where one model could serve many languages in international markets.

One related topic to typeahead search result blending is query auto completion [4]. However, the two topics are fundamentally different in terms of system architecture. The former has complicated indexing systems to retrieve documents. A ranking system is build to blend the documents where query intent is a feature. The latter's objective is only to generate a complete query without any document side information.

8 CONCLUSION

This paper proposes a comprehensive framework for modeling the query intent in search systems for different product components. The proposed deep learning based models are proven to be effective and efficient for online search applications. Discussions about the challenges for deploying these models to production as well as our insights in making these decisions are provided. We hope the framework as well as the experiences during our journey could be useful for readers designing real-world query understanding and text classification tasks.

REFERENCES

[1] Luca Maria Aiello, Debora Donato, Umut Ozertem, and Filippo Menczer. 2011. Behavior-driven clustering of queries into topics. In *CIKM*. 1373–1382.

[2] Jaime Arguello, Fernando Diaz, Jamie Callan, and Jean-Francois Crespo. 2009. Sources of evidence for vertical selection. In *SIGIR*. ACM, 315–322.

[3] Azin Ashkan, Charles LA Clarke, Eugene Agichtein, and Qi Guo. 2009. Classifying and characterizing query intent. In *ECIR*. Springer, 578–586.

[4] Fei Cai and Maarten De Rijke. 2016. A survey of query auto completion in information retrieval. *Foundations and Trends® in Information Retrieval* (2016).

[5] Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang, Jian-Tao Sun, Enhong Chen, and Qiang Yang. 2009. Context-aware query classification. In *SIGIR*. ACM, 3–10.

[6] Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909* (2019).

[7] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *DLRS*. 7–10.

[8] Jackie Chi Kit Cheung and Xiao Li. 2012. Sequence clustering and labeling for unsupervised query intent discovery. In *WSDM*. 383–392.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Homa B Hashemi, Amir Asiaee, and Reiner Kraft. 2016. Query intent detection using convolutional neural networks. In *WSDM*.

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

[12] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. 2009. Understanding user's query intent with wikipedia. In *WWW*. ACM, 471–480.

[13] Bernard J Jansen, Danielle L Booth, and Amanda Spink. 2008. Determining the informational, navigational, and transactional intent of Web queries. *Information Processing & Management* 44, 3 (2008), 1251–1266.

[14] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[15] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *ACL*. 655–665.

[16] In-Ho Kang and GilChang Kim. 2003. Query type classification for web document retrieval. In *SIGIR*. ACM, 64–71.

[17] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[18] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*.

[19] Ji Young Lee and Franck Dernoncourt. 2016. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, San Diego, California, 515–520. https://doi.org/10.18653/v1/N16-1062

[20] Xiao Li, Ye-Yi Wang, and Alex Acero. 2008. Learning query intent from regularized click graphs. In *SIGIR*. ACM, 339–346.

[21] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *IJCAI*.

[22] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[23] Xiang Ren, Yujing Wang, Xiao Yu, Jun Yan, Zheng Chen, and Jiawei Han. 2014. Heterogeneous graph-based intent learning with queries, web pages and wikipedia concepts. In *WSDM*. 23–32.

[24] Daniel E Rose and Danny Levinson. 2004. Understanding user goals in web search. In *WWW*. 13–19.

[25] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. 2011. Intent-aware search result diversification. In *SIGIR*. ACM, 595–604.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[27] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*. 649–657.

[28] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630* (2015).

[29] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639* (2016).