Building Large-Scale Deep Learning System for Entity Recognition in E-Commerce Search

Musen Wen
mwen@ebay.com
eBay Inc.
San Jose, California, USA

Deepak Kumar Vasthimal
dvasthimal@ebay.com
eBay Inc.
San Jose, California, USA

Alan Lu
alalu@ebay.com
eBay Inc.
San Jose, California, USA

Tian Wang
twang5@ebay.com
eBay Inc.
San Jose, California, USA

Aimin Guo
aiguo@ebay.com
eBay Inc.
San Jose, California, USA

ABSTRACT

Named-Entity-Recognition (NER) or Item Aspect Recognition task is fundamental to e-commerce marketplace. A structured listing (i.e. item aspect and catalog) is critical to the success of the marketplace - it helps the seller to put their listings to the right catalog and aspects so that the buyers can easily find the most relevant and accurate listings they are looking for. For e-commerce search engine, item aspects (brand, color, size, texture, etc.) should be automatically recognized from users' shopping queries in order to accurate understand their shopping intent. An accurate query and item aspect understanding helps to match seller's listing to buyer's purchase intend. However, in practice, this still remains challenge in e-commerce marketplace due to a couple reasons, e.g. the sparsity of the data - hundreds of millions of item aspect-name and aspect-value pairs (e.g. brand=Nike); noisy (low quality), not scalable but expensive label data that are obtained via human labeling [12] effort; the lack of context in generally very short search queries; the disperse of context and dis-orderedness in listing title or descriptions, etc. among others. All those imposes a big challenge that affect the performance of aspect recognition for e-commerce practice. In this paper, we introduce an end-to-end machine learning system to build an effective aspect recognition system for search engine that leverages the generated label data from different legacy systems and effort, without any extra human-label effort (and thus no incurred cost). The framework is constructed with multiple machine learning components that are built to optimize the search relevance and conversion as an end goal. We show that the proposed aspect recognition machine learning system improved search results quality and relevance greatly compared to the existing already strong baseline system for e-commerce search.

eBay Inc. San Jose, USA.

BDCAT '19, December 2–5, 2019, Auckland, New Zealand
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7016-5/19/12…$15.00
https://doi.org/10.1145/3365109.3368765

KEYWORDS

name-entity-recognition, aspect recognition, e-commerce, neural networks, text mining, deep learning, big data, artificial intelligence, parallel processing, distributed computation

ACM Reference Format:
Musen Wen, Deepak Kumar Vasthimal, Alan Lu, Tian Wang, and Aimin Guo. 2019. Building Large-Scale Deep Learning System for Entity Recognition in E-Commerce Search. In *Proceedings of the IEEE/ACM 6th International Conference on Big Data Computing, Applications and Technologies (BDCAT '19), December 2–5, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3365109.3368765

1 INTRODUCTION

In online shopping, users try to find out the items of interest through search if they have strong intent on what they are looking for. Search is mostly unstructured. However, e-commerce is a highly structured business, where items follow a category hierarchy. Items generally have attributes (i.e. "aspects") like brand, color, material, size, etc. Due to this nature, a better understanding of users' *unstructured* shopping queries in a *structured* way, the better we can serve the purpose of searching - to retrieve most relevant items, e.g. with the right brand, category, size and color, etc. for users. The task falls into the so called "Named-Entity-Recognition" domain in nature language technology. In e-commerce, it is particularly important to us - from structure data, query understanding to search ranking, item recommendation and personalization, etc. [4]. More speficially, when a buyer types in search box a shopping query like "cheap best sale brand new iPhone 8 64g gold", we need to accurately predict the "structured" attributes, i.e. the aspect name and aspect value pairs (short for "name-value pairs", or "tags" interchangeably) as "product = iPhone", "type = 8", "capacity = 64g", and "color = gold". In short, we need to understand the query in a structured way in order to retrieve the accurate brands and features, etc. per user's search. Although the task looks straightforward, in practice, this poses big challenge for e-commerce business. To name a few of these we faced:

- extreme sparsity of the unstructured and structured aspects from shopping queries and, items in inventory as well. It can be tens of millions of such aspect-name value pairs easily in e-commerce.
- the scalability - lack of high-quality, low-noise labeled data (i.e. tags) and low coverage for some categories - bottlenecks

the systemś overall performance; in another word, to label large volume of aspect-name value tags are very costly and time consuming, which is not scalable in this sense.

- contextual ambiguity (“gold" is color or material? “8" is an iPhone type, shoe's size ,or age?).
- the e-commerce domain challenge - shopping query is generally way too short (2 to 5 words) and while listing titles/descriptions are way too sparse (e.g. “nike shoes size 8,9,10,11,12 available brand new red black fast shipping"), etc.
- the machine learning system needs to be generalized well to recognize unseen shopping queries, etc.

In this paper, to address all of the above challenges, we propose a novel data mining [9] and deep learning based system to take on the entity recognition challenge at large scale. It achieves superior prediction performance and helps improve search relevance in our practice.

2 PROPOSED APPROACH

In this section, we introduce the proposed end-to-end named-entity-recognition system, which contains three major components: the data mining [9] module, the offline modeling model and online prediction. A brief description of the architecture is summarized in Figure 1. Our contribution is

- propose an effective end-to-end aspect recognition [13] method and system that without any extra human-labeling effort, by and cost;
- an effective text mining approach that mines from existing listing tags and producing query tags; with such high-quality, mined data, we build a large scale deep learning based named-entity-recognition system.
- We propose an aspect bit vector based search ranking booster.
- experiments shown that our end-to-end system is better over existing production baseline with improved search relevance

In what follows, we will describe each component sequentially in following subsections.

2.1 Query Tagging: Mining High Quality Tags with User Behavior Data

2.1.1 Query Tagging. In e-commerce space, especially for those whose business have been running for more than couple years, listed items have some sort of aspect tags. These tags are mainly coming from seller, i.e. aspect name-value tags for listing titles. These can be human-labeled data or machine learning system generated tags.

We start with adopting one (or more) type of aspect taxonomy from product listings, which we name it “buyer tags". However, we may not have the query-side tags. Unlike traditional ways using humans to label this aspect name-value pairs, which is costly and time-consuming, and we can not guarantee the quality from different human judges, or different vendors, instead, we leverage user-behavior signals (such as clicks, add2cart, buy or bid) to link the query to listings (Figure 1. Buyer-tag Log Mining Module). For each (query, listing-title) clicked pair, which some frequency threshold guarantee of quality confidence, the query is then tagged with the listing side's aspects - whenever there is a longest fragment matched. In tagging, we use BIO (Begin-Inside-Outside) tagging scheme. A much finer text processing, like normalization, standardization, etc. need to be used to enhance the entity quality within this data mining [9] module. This will generate high quality tags. Notice that, users' click log for example, cover most of the “potentially saleable" aspect name-value pairs, which ensure that our aspect name-value has a scalable and effective (in terms of potentially shopping) data coverage. In another words, “what ever matters is covered".

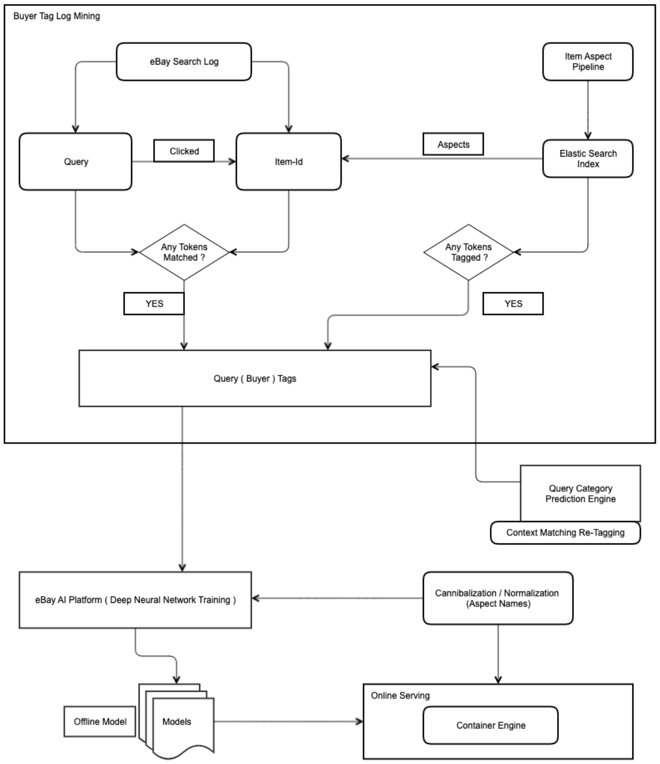

Figure 1: Architecture of Query Aspect Recognition System

2.1.2 Dis-ambiguity: Context-enriched Tagging-quality Enhancement via Query Intention Prediction. One of the biggest challenge for our problem is how can we find the intention [20] and differentiate, for example, “gold" in shopping query “iPhone rose gold" should be effectively differentiated and recognized from shopping query “iPhone 24k gold case", where first “gold" should be predicted as “color" and later as “material". To avoid garbage-in (training data) garbage out (machine learned tag prediction), we propose building another text classification module to predict the query intent - e.g. to predict which product category under the category hierarchy tree the shopping query's intend is. More specifically, we can use fastText [5] to classify the query intent to all leaf-category.

Each query re-tagging training sample from previous step is highly-confident if the predicted query's shopping intent category is in-lined with the clicked listingś category. This can be further relax, e.g. is the upper level category is in-lined, etc. This query intend prediction machine learning procedure reinforces the context alignment to disambiguate, thus significantly boosting the tagging quality.

2.1.3 Text Processing: Normalization of Aspect Names. Depends on downstream application, we need to refine the aspect-name space, this is particular helpful if we have multiple source of legacy aspect sources. For example, in one tagging system, it may use “color:red", while in the other, it uses “colour:red". We need to normalize both “color" and “colour" to “color", e.g. This can be done with the human-in-the-loop part that we will discuss in later subsection.

2.2 Sequence Tagging with Bi-direction LSTM and Conditional Random Field (CRF)

2.2.1 Modeling. Once we can continuously obtain high-quality training data from the data mining module described above, which is the “bread and butter", we end up with a good position to train our customized aspect recognition [13] system. Notice that we have billions of extremely sparse tagged queries (with aspect name-values). In recently year, deep learning technique has shown much better performance over the benchmarks for different named-entity-recognition [19] tasks. The Long-Short-Term-Memory (LSTM) + Conditional Random Field (CRF) method has become the new standard. We adopt the neural architecture from [3] and a recent research work[19] to train our model (Figure 1. in the offline Model Module).

We use the following implementation,

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \\
\mathbf{c}_t &= (1 - \mathbf{i}_t) \odot \mathbf{c}_{t-1} + \\
&\quad \mathbf{i}_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}$$

where σ is a sigmoid function, $\mathbf{x}_t$ is the input at the t-th timestep, and $\odot$ is element-wise multiplication.

A bi-directional LSTM [1] is used, where we represent the word by concatenating both its left and right context representations, $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$

To model the tag decision jointly, a conditional random field (CRF) model is build on top, which is to optimize the log-probability of correct tag sequence[19]:

$$\log(p(\mathbf{y}|\mathbf{X}) = s(\mathbf{X}, \mathbf{y}) - \log\left(\sum_{\tilde{\mathbf{y}} \in \mathbf{Y_X}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})}\right)$$

where $\mathbf{Y}_X$ is all possible tag sequence for sentence $\mathbf{X}$.

Our model consumes billions of training samples and is trained with eBay's proprietary built GPU clusters – namely, eBay AI platform, which we will describe briefly in next subsection. For billions of training samples, a converged model could be obtained within 2 days. The online model serving is developed and hosted on Google Cloud Platform (GCP) which we have tested intensively and maintained end-to-end latency less than 30 milliseconds (see Figure 1 Online Serving Module).

3 MODELING TRAINING PLATFORM

Our large scale training system is built upon eBay's GPU cluster - eBay Artificial Intelligence (AI) Platform. AI platform is an easy to use, open, fast and scalable AI platform on eBay private cloud. A general design goal is to build a modern AI platform that is designed to empower the corporate-wise data science and engineering community to quickly experiment, productize and enable machine learning at large scale.

Some key tenets of AI platform that our modeling practice heavily used:

- Polyglot: Build models in Python, Java, Scala, C++, R or any programming language of choice.
- Train models using heterogeneous software: we run the training with Tensorflow. It also support many other deep learning package such as Caffe, PyTorch, etc. We designed the AI platform to be agnostic to chosen framework.
- Train models on heterogeneous hardware: we accelerate our deep learning model training with the platform's high-performance compute cluster powered by Graphic Processing Units (GPU), Central Processing Units (CPU) with high-memory (all the way to 1TB).
- For production purpose, our platform provides easy access to our e-commerce marketplace data. i.e. provides direct access to large training data-sets residing on our private Hadoop clusters [9] and Network File Storage (NFS).
- Last, the end-to-end aspect recognition system is ran through Machine Learning (ML) pipeline. We construct and fully automate the training pipelines. In more details, the pipelines are constructed using declarative constructs that stitch together and form a complete life-cycle (get data -> prepare data -> extract features -> train model -> evaluate model). Notice that, aspect recognition system for search is one of our many machine learning products that run on AI platform. This is enable by the facts that AI platform operates at scale [6] and can handle 1000s of automated machine learning training jobs consisting of 10s to 100s of task(s) running in parallel daily. Each workflow/job could range from simple serial to complex parallel flows based on different tasks.

4 MODEL TRAINING ARCHITECTURE DESIGN

In this section, we describe our training and deployment platform's architecture design in order to illustrate our effort to build large-scale deep learning in general, and aspect recognition (LSTM+CRF) in particular, systems at corporate usage levels. First, the AI platform is a multi-tenant cloud based platform deployed in our private cloud. A birds eye view of the architecture is described in Figure 2.

For an end-to-end modeling training and deployment process, we can use command line interface, REST client, web browser or python client to submit training job through a secure proxy to API server. API server (REST) would then schedule execution of the training pipeline on Kubernetes platform by launching a workflow engine on Kubernetes. Workflow engine manages the execution of job and provides high availability, resiliency to the task(s) under execution. On the other hand, the training data can be read from distributed Hadoop Data File[11] System (HDFS),

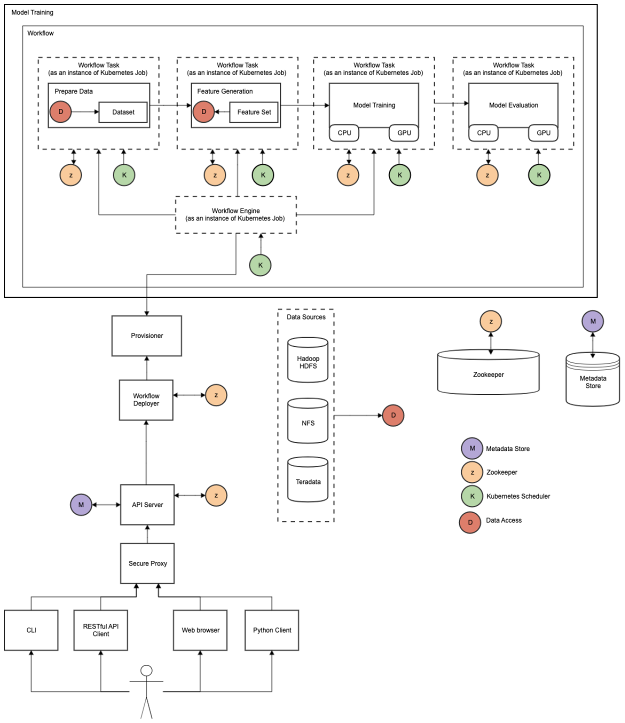

Figure 2: Architecture of AI platform

Teradata or Network File System (NFS). Apache Zookeeper is then used to store configuration information during job execution and it provides distributed synchronization. Once our model training job completes, logs [2] are generated on nodes that perform the actual training and moved to NFS that users could access through an web interface. Above discussion gives a high level overview of distributed, resilient [10] and highly available training platform ready for production.

5 E-COMMERCE APPLICATIONS

Once the entity recognition is built, we have a sequence of downstream optimization and applications. We describe and discuss a few of the most important tasks in following sections.

5.1 Human-in-the-loop Refinement and Continuous Optimization

The machine learning system provides very accurate prediction. Depends on downstream tasks, we further fine tune the system - a process we called "human-in-the loop" is critical for downstream applications. For example, the original tag space differentiate "men's shoe size (US)" and "size" in general. But, depends on various applications, do we need that granularity or not? In one use case, we may need to further cluster the aspects – a process to fine reducing the aspect-name space to impact the query recall size in search. This provides great flexibility, and fully depends on what kinds of downstream tasks we want to tackle.

Continuous optimization is another feature that empowers the whole system to capture the most updated user behavior, trends, aspect name space, etc. By running the data mining pipeline and offline model training continuously, we will see the benefit from it.

5.2 Aspect Matching and Recall Generation

The query aspect recognition [13] system is used in the search engine that powers eBay's mobile shopping bot and cross-border trade business. As one module for recall filtering (after semantic relevance, BM25, etc.) we identify the query aspect and look up our listing index to fetch our recall with items that have matched aspect tags (e.g. brand=nike, color=red, etc.). Recall is further passed to the ranking module to generate final ranking results.

5.3 Rank Booster: Aspect Bit Vector Boosting

Similar to a recent publication from Google's shopping team [8], we propose a query aspect bit-vector booster [16] to serve as either ranking feature or final ranking booster [16] (which is Google's way). We propose, from historical data, per leaf-category, we first estimate top K (where K is reasonably large to cover most items) aspect name-value pairs and encode them with a K bit vector. In serving time, for each query, we have two separate machine learning systems to predict

- query's intended category
- named entities and then evaluate it against the historical top K aspects for this leaf category

A boosting score for a recall can be easily derived based on how many named entities is matched with query. This (transformed) score will be used as a final ranking booster [16].

6 EXPERIMENTS AND RESULTS

We trained the large-scale aspect recognition model on the fine mining data. In Figure 3, Figure 4, Figure 5, Figure 6 and Figure 7 we plot specifically the training loss, overall accuracy, precision, recall and f1 score versus the training epochs. Notice that we has intentionally to add an unspecified Δ to each metric, whenever it applies, so as to normalize the last number to be 100 in order to follow corporate information release rule.

We conduct series of A/B tests [14] on various hyper parameters to determine the best possible parameters. Metrics were automatically generated and plotted by in-house monitoring platform [7]. The model training consisted of 12 epochs to achieve the desired metrics from the best hyper parameters. A unified model for all eBay meta categories is then deployed for downstream search task.

We have in-house build large scale NLP and logistic regression based entity recognition system. The system is built upon a lot of domain expertise-aided feature engineering. This benchmark system has been producing consistent and efficient support for the whole search engine. Both offline evaluation and paid human-judgment relevance evaluation shows the superior of our proposed new method and systems - where the proposed method and system is on par or better with current production system in head/torso queries and sees over half percent relative lift in NDCG@5 (where our application is focus with mobile shopping in this use case) for tail queries, on top of our already very strong benchmark system.

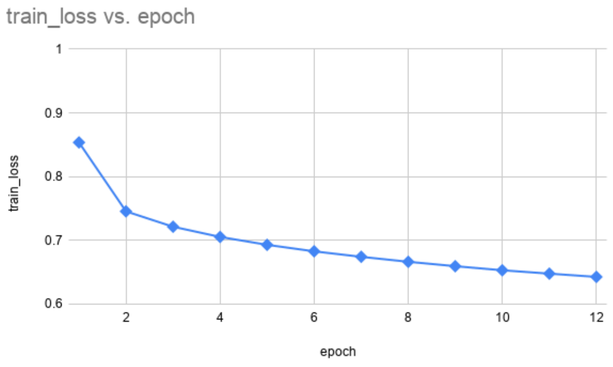

Figure 3: train loss v.s. epoch

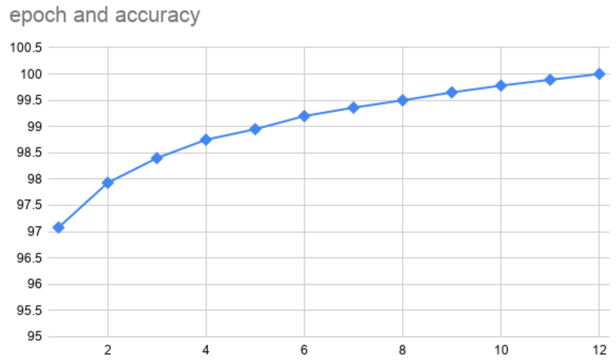

Figure 4: accuracy (plus ∆) v.s. epoch

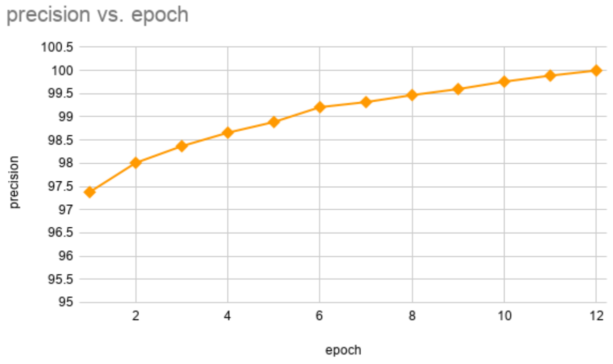

Figure 5: precision (plus ∆) v.s. epoch

7 RELATED WORK

Our proposed query tag mining system is novel and practically very effective. To our best knowledge, this is the first exploration of such low-cost method. Long-Short-Term-Memory [3] plus Conditional Random Field for named-entity-recognition task is proposed very recently[15], which becomes a new trend and benchmark. In the e-commerce domain, the Google shopping team introduced the system that predicts "latent" aspect from shopping queries [8], this is different from our system, where we predict directly the aspect from shopping queries. A combination of the explicit and implicit aspect prediction will form a completed picture for aspect recognition [13] task for e-commerce shopping.

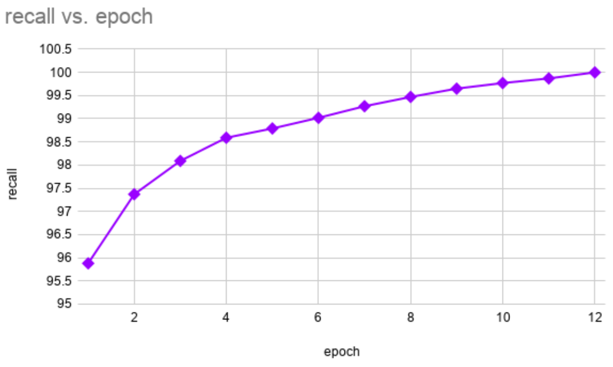

Figure 6: recall (plus ∆) v.s. epoch

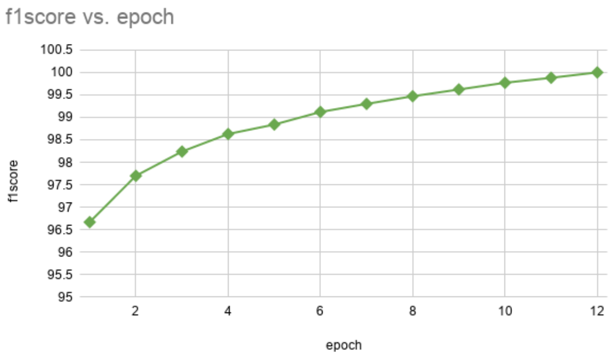

Figure 7: f1-score (plus ∆) v.s. epoch

8 CONCLUSION

In this paper, we introduce our end-to-end system and approach that handle the query aspect recognition task in practice. This is inexpensive, but end up with high-quality de-noised training data, approach. It effectively solves some most challenging problems we face in the e-commerce practice: aspect data sparsity, large scale, extremely noisy data, unstructured query intend, etc. With experiments and human-judged evaluations, we shown the proposed method further improved the already strong baseline we have in production for end-to-end search task. This is a first time, to the best of our knowledge, a *cost-effective* (in terms of human labeling effort [12]) approach to building an effective system for aspect recognition for shopping queries. Augmented with the work from Google shopping [15], it will be a very general and effective framework for the e-commerce industry in the near future.

REFERENCES

[1] Alex Graves and JÃijrgen Schmidhuber. 2005. Framewise phoneme classifica-tion with bidirectional LSTM and other neural network architectures.NEURALNETWORKS(2005).

[2] D. K. V, R. R. Shah and A. Philip, "Centralized log management for pepper," 2011 IEEE Third International Conference on Cloud Computing Technology and Science, Athens, 2011, pp. 1-3.

[3] Sepp Hochreiter and Jürgen Schmidhuber. [n.d.]. Long Short-Term Memory.NeuralComput.9, 8 ([n. d.]).

[4] Mahesh Joshi, Ethan Hart, Mirko Vogel, and Jean-David Ruvini. 2015. DistributedWord Representations Improve NER for e-Commerce. InVS@HLT-NAACL.

[5] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bagof Tricks for Efficient Text Classification.CoRRabs/1607.01759 (2016).

[6] D. K. Vasthimal, S. Kumar and M. Somani, "Near Real-Time Tracking at Scale," 2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2), Kanazawa, 2017, pp. 241-244.

[7] S. Kumar and D. K. Vasthimal, "Raw Cardinality Information Discovery for Big Datasets," 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Washington, DC, USA, 2019, pp. 200-205.

[8] Chao-Yuan Wu, Amr Ahmed, Gowtham Ramani Kumar, and Ritendra Datta. 2017.Predicting Latent Structured Intents from Shopping Queries. InProceedings of the 26th International Conference on World Wide Web (WWW '17). 1133–1141.

[9] Clifton, Christopher (2010). "Encyclopædia Britannica: Definition of Data Mining". Retrieved 2010-12-09

[10] D. Vasthimal, "Robust and Resilient Migration of Data Processing Systems to Public Hadoop Grid," 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, 2018, pp. 21-23.

[11] K. Shvachko, H. Kuang, S. Radia and R. Chansler, "The Hadoop Distributed File System," 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), Incline Village, NV, 2010, pp. 1-10.

[12] Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2004. Text classification by labeling words. In Proceedings of the 19th national conference on Artifical intelligence (AAAI'04), Anthony G. Cohn (Ed.). AAAI Press 425-430.

[13] G. Jones and Bir Bhanu, "Recognition of articulated and occluded objects," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 7, pp. 603-613, July 1999.

[14] D. K. Vasthimal, P. K. Srirama and A. K. Akkinapalli, "Scalable Data Reporting Platform for A/B Tests," 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), Washington, DC, USA, 2019, pp. 230-238.

[15] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami,and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. InProceedings of the 2016 Conference of the North American Chapter of the Associationfor Computational Linguistics: Human Language Technologies. Association forComputational Linguistics.

[16] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15). ACM, New York, NY, USA, 373-382.

[17] David E. Losada Juan M. Fernández-Luna. Advances in Information Retrieval. In Proceedings of the 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005.

[18] Cyril Goutte and Eric Gaussier. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation.Springer-Verlag Berlin Heidelberg 2005

[19] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C. 2016. Neural Architectures for Named Entity Recognition. arXiv e-prints arXiv:1603.01360.

[20] Z. Wang, Y. Qi and J. L. Z. Ma, "User intention understanding from scratch," 2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE), Aalborg, 2016, pp. 1-4.