Search Query Refinement for Japanese Named Entity Recognition in E-commerce Domain

Yuki Nakayama **Ryutaro Tatsushima** **Erick Mendieta**
{yuki.b.nakayama, chen.a.zhao, erick.mendieta}@rakuten.com
Koji Murakami **Keiji Shinzato**
{koji.murakami, keiji.shinzato}@rakuten.com
Rakuten Institute of Technology, Rakuten Group, Inc.

Abstract

In the E-Commerce domain, search query refinement reformulates malformed queries into canonicalized forms by preprocessing operations such as "term splitting" and "term merging". Unfortunately, most relevant research is rather limited to English. In particular, there is a severe lack of study on search query refinement for the Japanese language. Furthermore, no attempt has ever been made to apply refinement methods to data improvement for downstream NLP tasks in real-world scenarios. This paper presents a novel query refinement approach for the Japanese language. Experimental results show that our method achieves significant improvement by 3.5 points through comparison with BERT-CRF as a baseline. Further experiments are also conducted to measure beneficial impact of query refinement on named entity recognition (NER) as the downstream task. Evaluations indicate that the proposed query refinement method contributes to better data quality, leading to performance boost on E-Commerce specific NER tasks by 11.7 points, compared to search query data preprocessed by MeCab, a very popularly adopted Japanese tokenizer.

1 Introduction

Modern E-Commerce services rely on named entity recognition (NER) to understand customer demands from their input search queries. In general, NER in E-Commerce has enjoyed remarkable success with regard to recognizing important attributes such as brand names from search queries. However, it is prone to poor performance upon malformed queries such as "Ni ke" and "adidasmask", which unfortunately occurs frequently due to inevitable typos. To canonicalize the malformed terms in user inputs, two fundamental operations are involved: term merging (e.g., "Ni ke" → "Nike") and term splitting (e.g., "adidasmask" → "adidas mask"). Considering the fact that most previous studies focus on English search queries, we propose a two-stage method based on sequence labeling to refine Japanese queries in the E-Commerce domain (Figure 1). First, we canonicalize raw inputs by token matching using three dictionaries: one manually constructed, one from Wikipedia and the other containing past queries from an E-Commerce site (§3.1). Second, BERT-CRF (Souza et al., 2019) is used to predict chunk tags for out-of-the-dictionary terms on the character level. Meanwhile BERT-CRF leverages the results of the dictionary matching (§3.2) to obtain better input representations.

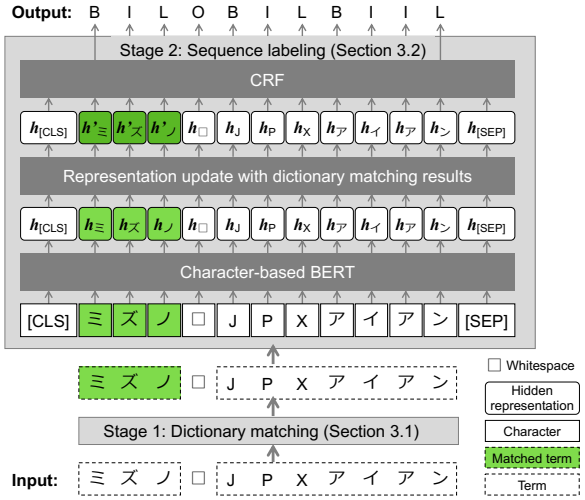

Figure 1: Overview of our query refinement method. The translation of the input is "Mizuno□JPXIron."

Experiments on real Japanese queries demonstrate that the combined two-stage method outperforms the standalone approaches like dictionary matching and BERT-CRF (§4.1). Furthermore, we illustrate that query refinement helps improve data quality, which boosts the accuracy of brand recognition, a classic in-demand NER task in the E-Commerce domain. Solid experiments show that a sequence tagging model trained with refined search queries achieves a better F_1 score by 11.7 points on

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), pages 447–452

customized datasets (§4.2). In brief, the contribution of this paper is twofold. First, a novel idea of Japanese query refinement is designed for practical E-Commerce applications. Second, quantitative analysis is provided to verify how query refinement benefits brand recognition as a real-world use case.

2 Related Work

Search Query Refinement Li et al. (2022) used distant supervision for E-Commerce query refinement via character-level attention-based BiLSTM-CRF. Unique to Japanese, acronym and abbreviation expansion count as basic refinement operations according to (Uchiumi et al., 2011). Li et al. (2012) proposed an approach based on the Hidden Markov Model for query refinement including the two operations. They proposed feature functions which measure the probability of mistyping, e.g., "water proof" to "waterproof." To compute such probability, a large set of query-correction pairs (6.5 million queries) was used. However, their method is not applicable for Japanese due to data scarcity (no refined Japanese examples of non-romanized query-correction pairs). Yang et al. (2022) experimented query refinement in Amazon EC with emphasis on spelling correction. MeCab (Kudo, 2006) is a representative Japanese morphological tool commonly used for text preprocessing, which at times refines stale user inputs as well. As is mentioned in §1, we show that over-tokenization adversely affects E-Commerce NER performance by comparing the proposed refinement method to MeCab. To our best knowledge, no prior study yet exists about query refinement directly targeting Japanese and its E-Commerce applications.

E-Commerce Named Entity Recognition In general, NER in E-Commerce context refers to a wide spectrum of NLP problems centered at attribute value extraction from product description and search queries (Xu et al., 2019). For example, given a product description such as "2019 Summer Women Button Decorated Print Dress Long Dresses Plus Size by GUCCHI", their purpose is identifying "GUCCHI" as a brand attribute, "Plus Size", as a size attribute. Cowan et al. (2015) used the conditional random field (CRF) with handcrafted features to recognize three types of entities in travel search queries. Kozareva et al. (2016) applied LSTM-CRF for brand extraction from shopping queries. Zhai et al. (2016) proposed several grammar rules to infer query structures. Jiang et al. (2021) introduced weakly supervised label completion to enhance training data effectiveness. Nevertheless, in case of the Japanese language, search query refinement remains relatively untouched in E-Commerce NER problems.

3 Proposed Method

We formalize query reformulation as a character-level sequence labeling problem. Consider query $q = (c_1^1, c_2^1, \ldots, c_{i-1}^1, c_i^0, c_{i+1}^2, \ldots, c_n^m)$ where n is the number of characters in q, m is the number of terms in q and c_i^j represents the i-th character in q and that it belongs to the j-th term. If c is whitespace, j is 0. Terms are tokens made from whitespace splitting. Given a query q with the length of n, the model is trained to return $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ where y_i is one of BILOU chunk tags (Sekine et al., 1998; Ratinov and Roth, 2009).

Figure 1 displays overview of our two-step query refinement method. Input queries are first matched with entries from three separate dictionaries before being fed into the character-based BERT-CRF (Souza et al., 2019). Prepending a dictionary matching step ahead of the model is necessary as search queries are likely to lack context due to the short length making BERT-CRF alone struggle to perform very well. If prediction from BERT-CRF is different from one from the dictionary matching, we give precedence to the dictionary matching.

3.1 Stage 1: Dictionary Matching

We conduct the leftmost-longest matching on a term sequence of q with entries in dictionaries, and then assign a chunk tag to each character in matched terms. We construct the following three dictionaries for the dictionary matching. Since their accuracy is different, we use them separately in the following order, instead of integrating them into a single dictionary.

1. Dictionary Manually Tailored (ManDic) We manually list up phrases associated with the E-Commerce domain such as brands and product series. The number of phrases is $78,539$.

2. Dictionary from Wikipedia (WikiDic) We use Wikipedia dump to gather phrases not covered in ManDic. We collect 1.9 million articles from the Japanese Wikipedia dump as of March, 2021 and 12.7 million articles from the English Wikipedia dump as of February 2021, respectively. We adopt titles in those articles as entries of the dictionary after removing strings surrounded by parenthesis.

To increase the coverage, we extract phrases from body text in Japanese Wikipedia articles. We first tokenize the text, and then gather word bi-, tri- and four-grams. Next, we compute pointwise mutual information (PMI) of those collected n-grams, and add them if the score exceeds the threshold θ_1. PMI is calculated as follows:

$$\mathrm{PMI}(x_1, x_2) = \log_2 \frac{N \times \mathrm{F}(x_1, x_2)}{\mathrm{F}(x_1) \times \mathrm{F}(x_2)} \quad (1)$$

where F(x_1,x_2) denotes frequency that the words x_1 and x_2 appear as adjacent query terms and N denotes the total number of terms. Given n query terms $t_1 \square \ldots \square t_i \square \ldots \square t_m$ divided by whitespace in a query, PMI for a tri-gram $t_i t_{i+1} t_{i+2}$ and PMI for a four-gram $t_i t_{i+1} t_{i+2} t_{i+3}$ are calculated by Equations (2) and (3), respectively.

$$\begin{aligned}&\mathrm{PMI}_{\mathrm{tri}}(t_i t_{i+1} t_{i+2}) \\ &= \max\{\mathrm{PMI}(t_i t_{i+1}, t_{i+2}), \mathrm{PMI}(t_i, t_{i+1} t_{i+2})\}\end{aligned} \quad (2)$$

$$\begin{aligned}&\mathrm{PMI}_{\mathrm{four}}(t_i t_{i+1} t_{i+2} t_{i+3}) \\ &= \max\{\mathrm{PMI}(t_i, t_{i+1} t_{i+2} t_{i+3}), \\ &\qquad \mathrm{PMI}(t_i t_{i+1}, t_{i+2} t_{i+3}), \\ &\qquad \mathrm{PMI}(t_i t_{i+1} t_{i+2}, t_{i+3})\}\end{aligned} \quad (3)$$

Computing PMI based on frequency in the queries, we can remove irrelevant n-grams to the queries.

3. Dictionary from Query Logs (QryDic) We use queries to collect phrases not described in Wikipedia. Assuming that terms either starting with a prefix or ending with a suffix are likely to be a single phrase, we regard query terms including either a prefix or a suffix as entries of the dictionary.

Furthermore, we extract adjacent words with high correlation from the queries as dictionary entries. More precisely, we tokenize all queries, and then compute PMI of all word bi-, tri- and four-grams using Equation 1. We use the frequency that the words x_1 and x_2 adjacently appear in query terms as F(x_1,x_2) and the total number of words in the all queries as N. If the scores of an n-gram is larger than the threshold θ_2, we regard it as a dictionary entry.

3.2 Stage 2: Sequence Labeling

Despite the dictionaries described in §3.1, there exist character sub-sequences in query q that match no dictionary entries. We employ character-based BERT-CRF to tag those sub-sequences. We create a string [CLS,q,SEP], and then feed it to BERT (Devlin et al., 2019) to obtain their hidden representations $\boldsymbol{H} = [\boldsymbol{h}_{\mathrm{CLS}}, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_n, \boldsymbol{h}_{\mathrm{SEP}}]$. CLS and SEP are special tokens to represent a classifier token and a separator, respectively.

To obtain better hidden representations, we leverage the results of the dictionary matching. Similar to Watson et al. (2018), for each sub-sequence q_t in q that matches the dictionary entry t, we first average hidden representations corresponding to q_t, and regard it as the representation of the term $\boldsymbol{g}_t$. Next, we take the average over $\boldsymbol{g}_t$ and each hidden representation from q_t to reflect the representation of the term. The reason why we limit sub-sequences that match dictionary entries is to avoid reflecting hidden representations obtained from malformed terms such as "adidasmask" to $\boldsymbol{H}$.

After updating $\boldsymbol{H}$ with the results of the dictionary matching, we feed it to a CRF layer to compute the probability of a chunk tag sequence.[1]

	Train	Dev.	Test
Number of queries	7,385	1,824	1,000
w/ over-merged terms	310	74	97
w/ over-tokenized terms	899	209	144
w/o malformed terms	6,187	1,641	763

Table 1: Statistics of the query refinement task data.

	Train	Dev.	Test
Number of queries	9,000	2,248	2,225
w/ brands	2,467	944	901
w/o brands	6,533	1,304	1,324

Table 2: Statistics of the brand extraction task data.

4 Experiments

We evaluate our query refinement method and its impact on NER tasks using real-world search queries obtained from an e-commerce platform in Japan. To compute PMI, we use all queries issued in the past two years after tokenizing the queries with MeCab with UniDic (Den et al., 2008). As a result, WikiDic and QryDic contain $101,152,647$ and $53,570,779$ entries, respectively.

Dataset For query refinement, two Japanese annotators help to label BILOU chunk tags for each character in 10,209 queries. For NER, one more Japanese annotator labels brand tags, the most essential named entity class in real-world busi-

[1]We feed the updated $\boldsymbol{H}$ to a fully-connected layer to adjust the dimension of hidden representations to the number of chunk tags before the CRF layer.

ness, among 13,473 queries. The statistics of both datasets are shown in Tables 1 and 2.

Model We implemented our model using PyTorch. Model parameters and the values of θ_1 and θ_2 were determined using the development set. The followings are details of the model training for the query refinement task. We run model training on NVIDIA-V100 GPU (Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz). When fine-tuning, we keep the dropout probability at 0.1 and an optimum number of epochs determined in the development set. The initial learning rate is $1e^{-5}$, and the batch size is 32. We used the pre-trained Character-BERT model on Japanese Wikipedia[2]. The number of dimensions for hidden representation is 768, the number of transformer blocks is 12, the number of self-attention heads is 12, and the total number of parameters for the pre-trained model is $110M$. The vocabulary size is $6,144$. The training time was 10 minutes.

4.1 Query Refinement Task

As baselines, we use MeCab with UniDic, retrained MeCab[3] with UniDic, BiLSTM-CRF and BERT-CRF. As evaluation measure, we compute chunk-level F_1-score considering only exact matches. We used an evaluation script [4] of Lekhtman et al. (2021) for the computation.

Table 3 shows the experimental results. From the table, we can observe that our method achieved a higher F_1 score of 83.86 that outperformed the four baselines. The differences between our method and the baselines were significant ($p < 0.01$) under the two-tailed paired t-test. We can also see that our method performed better than the dictionary matching or BERT-CRF alone. This means that both approaches complement each other. Moreover, we found that all three dictionaries enhanced accuracy. Lastly, since the performance of the model that updates the representations using all terms is slightly lower than our method, we can conclude that selecting terms to update the representations works effectively.

We did error analysis on 272 query words for our method. We observed that the most frequent error cases were due to QryDic (64 errors). For instance, "キャラクターエプロン刺繍" was not correctly separated into "キャラクター (character)", "エプロン (apron)", and "刺繍 (embroidery)" since it was a dictionary entry due to a higher PMI score than θ_2. To avoid collecting such incorrect entries, better computation of the association for tri- and four-grams, such as (Levine et al., 2021), is necessary. Meanwhile, BERT-CRF tends to mistakenly split an item name into multiple words when it consists of a combination of an alphabet, a number, and a Japanese character, such as "リバーシブル D-86 (Reversible D-86)."

Method	F_1
MeCab w/ UniDic	$32.58^{\dagger}$
Re-trained MeCab w/ UniDic	$46.30^{\dagger}$
Character-based BiLSTM-CRF	$79.66^{\dagger}$
Character-based BERT-CRF (BERT-CRF$_{char}$)	$80.36^{\dagger}$
Dictionary matching	$82.41^{\dagger}$
w/o ManDic	$78.20^{\dagger}$
w/o WikiDic	$80.40^{\dagger}$
w/o QryDic	$76.30^{\dagger}$
Dictionary matching + BERT-CRF$_{char}$	$83.66^{\ddagger}$
w/ updating $\boldsymbol{H}$ using all terms	$83.46^{\dagger}$
w/ updating $\boldsymbol{H}$ using only matched terms (ours)	**83.86**

Table 3: Experimental results on the query refinement task. We performed a single trial for training all models above with the same random seed value. † and ‡ indicate statistically significant difference at 1% and 5%, respectively.

4.2 Brand Extraction Task

To prove the effectiveness of our query refinement method in a real-world scenario, we compare the performance of brand extraction models with and without refinement. We formulate brand extraction as a word-level sequence labeling problem. To enhance reproducibility, we use a BiLSTM-based sequence tagger, [5] publicly available from FLAIR (Akbik et al., 2019).

Implementation details: The sequence tagger model consists of preprocess, embedding layer, fully-connected layer, BiLSTM layer, and fully-connected layer. The preprocess step performs either MeCab or our method. In the embedding layer, we concatenated flair embedding and word embedding to compute an embedding for each word. Word embedding is trained with word2vec from 240 million queries randomly picked up in 2018. We used skip-gram as a model and ignored

[2] https://huggingface.co/cl-tohoku/bert-base-japanese-char-v2

[3] To train MeCab by ourselves, we manually segmented queries in the training data for our query refinement model and labeled a part-of-speech and pronunciation for each word.

[4] https://github.com/tonylekhtman/DILBERT/blob/main/other_eval.py

[5] https://github.com/flairNLP/flair/blob/master/flair/models/sequence_tagger_model.py

Pre-processor	Precision	Recall	F_1
MeCab	71.5%	32.9%	45.0
Our refinement	59.9%	53.4%	**56.7**

Table 4: Results on the brand extraction task.

Pre-processor	Segmentation & brand extraction results
MeCab	カネテツ|デリカ|フーズ
Our refinement	カネテツデリカフーズ

Table 5: Example of a query that our query refinement contributes to extract a correct brand. The input query is " カネテツ□デリカ□フーズ," where '□' represents whitespace. Extracted brands are highlighted in blue, and '|' indicates a boundary between terms.

all words with a total frequency lower than 100. The dimension of the word vector is 300. The first fully-connected layer is used to obtain hidden vectors for input of BiLSTM layer. The second fully-connected layer is used to convert hidden vectors to labels. Learning rate is 0.1. Batch size is 64. Flair embedding (Akbik et al., 2018) is contextual string embeddings that capture latent semantic information beyond standard word embeddings. Our flair embeddings are trained from 100 million queries randomly selected in 2018. The number of dimensions is $4,096$, which comprises $2,048$ for a forward model and $2,048$ for a backward model, respectively.

Table 4 shows the results of the brand extraction task. The method "MeCab" tokenizes queries using MeCab with UniDic, and then feeds the token sequences to the sequence tagging model. Meanwhile, "Our refinement" feeds queries refined by our method to the model. The tagging model with our refinement method outperformed the one with MeCab by 11.7 points in terms of the F_1 score. A working example from our method is in Table 5 while the MeCab tokenizer failed. The sequence tagger successfully recognized "カネテツデリカフーズ (Kanetetsu Derica Foods)" as a brand entity by refining the three query terms.

5 Conclusion

Whereas most existing research about search query refinement is limited to English, this paper designs a novel idea targeting Japanese in E-Commerce use case scenarios. We combined BERT-CRF with keyword matching as novel Japanese query refinery which outperforms both schemes when used alone. Moreover, we verified through experiments that refine queries lead to much better performance on downstream NLP tasks like product brand recognition. This query refinement system is already adopted in the e-commerce company that provided us with the queries.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Brooke Cowan, Sven Zethelius, Brittany Luk, Teodora Baras, Prachi Ukarde, and Daodao Zhang. 2015. Named entity recognition in travel-related search queries. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 3935–3941.

Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. 2008. A proper approach to Japanese morphological analysis: Dictionary, model, and evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Haoming Jiang, Danqing Zhang, Tianyu Cao, Bing Yin, and Tuo Zhao. 2021. Named entity recognition with small strongly labeled and large weakly labeled data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1775–1789, Online. Association for Computational Linguistics.

Zornitsa Kozareva, Qi Li, Ke Zhai, and Weiwei Guo. 2016. Recognizing salient entities in shopping queries. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 107–111, Berlin, Germany. Association for Computational Linguistics.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. jp*.

Entony Lekhtman, Yftah Ziser, and Roi Reichart. 2021. DILBERT: Customized pre-training for domain adaptation with category shift, with an application to aspect extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 219–230, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2021. Pmi-masking: Principled masking of correlated spans. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Yanen Li, Huizhong Duan, and ChengXiang Zhai. 2012. A generalized hidden markov model with discriminative training for query spelling correction. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 611–620.

Zhao Li, Donghui Ding, Pengcheng Zou, Yu Gong, Xi Chen, Ji Zhang, Jianliang Gao, Youxi Wu, and Yucong Duan. 2022. Distant supervision for e-commerce query segmentation via attention network. In *Intelligent Processing Practices and Tools for E-Commerce Data, Information, and Knowledge*, pages 3–19. Springer.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. 1998. A decision tree method for finding and classifying names in Japanese texts. In *Sixth Workshop on Very Large Corpora*, pages 171–178, Quebec, Canada.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf.

Kei Uchiumi, Mamoru Komachi, Keigo Machinaga, Toshiyuki Maezawa, Toshinori Satou, and Yoshinori Kobayashi. 2011. Japanese abbreviation expansion with query and clickthrough logs. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 410–419.

Daniel Watson, Nasser Zalmout, and Nizar Habash. 2018. Utilizing character and word embeddings for text normalization with sequence-to-sequence models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 837–843, Brussels, Belgium. Association for Computational Linguistics.

Huimin Xu, Wenting Wang, Xinnian Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5214–5223.

Fan Yang, Alireza Bagheri Garakani, Yifei Teng, Yan Gao, Jia Liu, Jingyuan Deng, and Yi Sun. 2022. Spelling correction using phonetics in E-commerce search. In *Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 63–67, Dublin, Ireland. Association for Computational Linguistics.

Ke Zhai, Zornitsa Kozareva, Yuening Hu, Qi Li, and Weiwei Guo. 2016. Query to knowledge: Unsupervised entity extraction from shopping queries using adaptor grammars. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 255–264, New York, NY, USA. Association for Computing Machinery.

A Appendices

A.1 Licenses

- Mecab is free software that is available followed by GPL(the GNU General Public License), LGPL, GNU, or BSD license.
- Unidic dictionary is free software availabe under GPL v2.0, LGPL v2.1, or BSD.
- Pre-trained Character BERT model described in Section 4.1 is distributed under the terms of the Creative Commons Attribution-ShareAlike 3.0.
- FLAIR library described in Section 4.2 is licensed under the following MIT license: The MIT License (MIT) Copyright © 2018 Zalando SE, https://tech.zalando.com