Spelling Correction using Phonetics in E-commerce Search

Fan Yang, Alireza Bagheri Garakani, Yifei Teng
Yan Gao, Jia Liu, Jingyuan Deng, Yi Sun
Amazon
Seattle, Washington, USA
{fnam,alirezg,yifeit,yanngao,hliujia,jingyua,yisun}@amazon.com

Abstract

In E-commerce search, spelling correction plays an important role to find desired products for customers in processing user-typed search queries. However, resolving phonetic errors is a critical but overlooked area. The query with phonetic spelling errors tends to appear correct based on pronunciation but is nonetheless inaccurate in spelling (e.g., "bluetooth sound system" vs. "blutut sant sistam") with numerous noisy forms and sparse occurrences. In this work, we propose a generalized spelling correction system integrating phonetics to address phonetic errors in E-commerce search without additional latency cost. Using India (IN) E-commerce market for illustration, the experiment shows that our proposed phonetic solution significantly improves the F1 score by 9%+ and recall of phonetic errors by 8%+. This phonetic spelling correction system has been deployed to production, currently serving hundreds of millions of customers.

1 Introduction

Search is critical to provide a great customer shopping experience in E-commerce. Usually, as the first step of the search workflow, spelling correction is responsible to reduce the irrelevant and sparse search results caused by spelling errors in search keywords. In addition, low latency is required considering spelling correction is only part of many modules in the search workflow. Despite the large amount of research on correcting spelling errors (Hládek et al., 2020), addressing phonetic errors is an important, but overlooked area. Phonetic spelling error typically happens when the query has similar pronunciation but is nonetheless inaccurate in spelling (e.g., "bluetooth sound system" vs. "blutut sant sistam" in English). Figure 1 describes the phonetic error percentage of misspelled queries based on a human annotated spelling correction dataset sampled from the Amazon search query log. We find that this type of spelling error dominates in multiple E-commerce markets with various languages, existing mostly on generic item terms (e.g., "nacklesh" vs. "necklace") and brand terms (e.g., "scalkendy" vs. "skullcandy"). This issue might damage customer trust and present greater challenges when E-commerce offers more products (i.e., brand names) with sensational spelling (e.g., Hasbro's Playskool [school]) and attracts customers with low written proficiency.

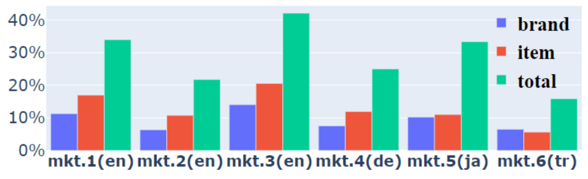


Figure 1: Phonetic error ratio on brand terms (blue), generic item terms (red) or any terms (green) out of all spelling errors in multiple markets (languages).

However, the traditional spelling correction system is not able to address phonetic spelling errors well because it usually searches the correction of given spelling errors up to a certain edit distance (Damerau, 1964) apart (Gorin et al., 1971; Whitelaw et al., 2009), while phonetic spelling errors could lead to large edit distance with variously noisy forms (e.g., EditDist("blutut" vs. "bluetooth")=4).

There exists multiple attempts to address phonetic errors by generating soundslike equivalent candidates based on phonetic algorithms (Atkinson, 2006). Although soundslike equivalent candidates may handle phonetic errors better, they tend to be too noisy to cover the correct spelling of non-phonetic errors in a limited size satisfying low latency requirements (shown in Section 3).

To address these limitations, we propose a generalized spelling correction system that enables us to integrate phonetics into E-commerce search without additional latency cost. It includes a new hybrid candidate generation method with phonetic mapping as well as an effective candidate ranking method leveraging phonetic signals. In particular, our major contributions include: (1) we propose an effective hybrid candidate generation method, which aims to capture the complementary candidates across edit distance and soundslike based methods to address both phonetic and non-phonetic spelling errors; (2) we propose a flexible candidate ranking stage by leveraging phonetic signals, which tends to rank the correct spelling of phonetic errors to the top; (3) our offline study shows a 9%+ speller overall improvement with an 8% improvement on phonetic errors without additional latency cost by incorporating phonetics into our generalized spelling correction system. To the best of our knowledge, this work is the first to propose an efficient and effective phonetic solution with ablation study in E-commerce search, and this phonetic solution can be easily applied to any automatic spelling correction system with candidate generation or ranking stage.

2 Problem Formulation and Modeling

In this section, we formally define our generalized spelling correction structure using phonetics. Most current search engines detect and correct spelling errors automatically. One of the most popular structures defined by (Kukich, 1992) includes candidate generation and candidate ranking steps based on the noisy channel model (NCM) (Jurafsky and Martin, 2008). The basic idea of NCM is to find the spelling correction C^* given the input query Q and its spelling correction candidates $C = c_1 \dots c_n$ via the Bayes' Rule:

$$C^* = \arg\max_{c} P(Q|C)P(C), \tag{1}$$

where the language model $P(C)$ represents the probability of the C to be correct, and the error model $P(Q|C)$ represents the chance of the transformation from C to Q. We define NCM score as the logarithm summation of the language and error model score. On top of this popular noisy channel structure, we introduce our generalized spelling correction system using phonetics below.

Hybrid candidate generation: The candidate generation step is to generate the correction candidates given an input query. In our proposed hybrid candidate generation stage, the first step is to leverage an auto-split-combine module tokenizing an input query into tokens and split (combine) tokens when the resulting bigram has a higher probability in the search query log. Second, each token's candidates are generated from the vocabulary dictionary (built from the search query log) up to a certain edit distance. Similar to (Sun et al., 2010; Whitelaw et al., 2009), a trie-based data structure is leveraged that allows to efficiently search within a maximum edit distance (e.g., a common setting is 2 to avoid high latency). Token candidates are sorted based on NCM score built on the search query log limited by a certain size. Caching is applied to avoid duplicated efforts. Considering the potentially large distance introduced by phonetic spelling errors (e.g., EditDist["blutut", "bluetooth"]=4), we design the hybrid candidate generation stage. It additionally includes a phonetic mapping with the key representing the pronunciation and its value being a list of token soundslike candidates, complementing edit-distance based token candidates. Specifically, we leverage phonetic algorithms' encoding (Vykhovanets et al., 2020) to convert the token to the key of the phonetic mapping. The token candidates in the phonetic mapping are extracted from tokenized Amazon product titles sorted by its frequency because Amazon product titles contain rich product information (brands, generic items, etc.), which commonly suffers from phonetic errors. In addition, the top candidates are required to have the same phonetic key, while the remaining candidates can allow a small edit distance of the phonetic key, which introduces noise, but with higher coverage.

Table 1 shows an example of the phonetic mapping in English leveraging Double Metaphone (DM) (Philips, 2000) phonetic algorithm with the phonetic encoding key: "PLTR". Top 10 candidates have the same DM phonetic key "PLTR" and the remaining 30 candidates allow one edit distance. For the given token "blutur" with DM key "PLTR", the correct candidate "bluetooth" is the $13th$ candidate with phonetic key "PLTT".

Top-K	Token Candidates
1-10	"platter","builder","boulder",...
11-40	"leather","holder","bluetooth",...

Table 1: The phonetic mapping of DM phonetic key: "PLTR"

Token candidates compose both edit-distance and phonetic mapping based candidates. Following Eq. (1) to approximately find the top-K query-level

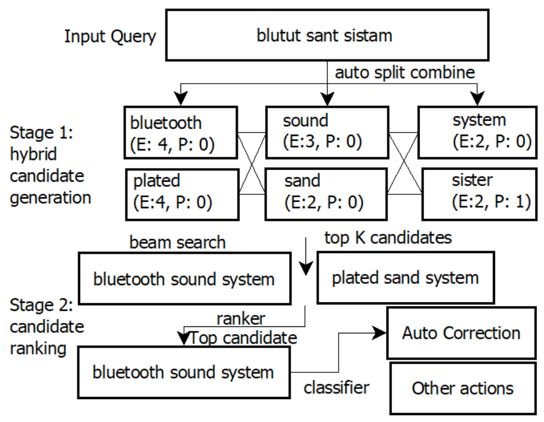


Figure 2: General workflow of spelling correction system. "E": edit distance of the token candidates. "P": edit distance based on phonetic encoding key.

candidates, we leverage the beam search algorithm with the beam search size K, tunable to balance the system latency and its performance. Figure 2 shows the general workflow of the spelling correction system including the hybrid candidate generation stage based on the input query "blutut sant sistam". Take the input token "sant" as an example, "sound" is generated from phonetic mapping, while "sand" is from both edit-distance and phonetic mapping based candidates with maximum edit-distance setting being 2.

Candidate ranking: Although the hybrid candidate generation stage can return candidates with sorted NCM score, we add this candidate ranking stage to flexibly leverage phonetic features and powerful ranking algorithms. Typical ranking features could be candidates' language scores from different sources (e.g., search query log, product titles, etc.) and different types of error scores characterized by the probability of each edit in different edit levels (e.g.,character-level (Mays et al., 1991; Church and Gale, 1991), subword-level (Brill and Moore, 2000) ,phrase-level (Sun et al., 2010)). We specifically add the phonetic distance feature: the edit distance of the phonetic encoding key between the input query and its candidate. The goal of adding this feature is to allow the model to rank the correct candidate of a phonetically misspelled query higher. For the ranking algorithm, the system is flexible to support linear models (e.g., logistic regression (Cox, 1958)), tree based model (e.g., XGBoost (Chen and Guestrin, 2016)) and deep learning models in point-wise and pairwise level. In addition, a classification module can be optionally added after the ranking stage (e.g., (Whitelaw et al., 2009)) to balance between auto-correction and no-correction speller actions. Figure 2 includes the workflow of the candidate ranking stage.

3 Experiments

In this section, we formally evaluate our proposed spelling correction system integrating phonetics. We use IN E-commerce market for illustration. We train and evaluate the candidate ranker based on the human annotated dataset with 30000 query-correction pairs from the search query log. We split 33.33% for training, validating, and testing. The evaluation metric is the F1 score, which is a harmonic mean of the precision and recall. Precision represents the speller's accuracy rate when its action is auto-correction, while recall is defined as the speller's accuracy rate for misspelled queries. We also report the recall of queries with phonetic errors and top percentile (TP) 99 latency of the spelling correction system in milliseconds. We treat the top 1 candidate as the correction of a given input query with auto-correction action when the top 1 candidate is different from the input query, although the system supports more complicated classifiers after the ranking stage. Relative impacts (instead of absolute values) are reported for legal requirements.

We first conduct an ablation study for the candidate ranking module. We apply a standard setting in the candidate generation step that allows 20 edit distance based token candidates with allowed maximum edit distance 2 and beam search size 12. Typical ranking features based on different types of language and error models are included, and we focus on the impact of the phonetic distance feature. Table 2 shows F1 score, recall of phonetic errors (column name: "ph-recall"), and TP99 latency evaluated in the test dataset. Column name "ph-dist" indicates if the phonetic distance feature is added in the ranking model. Row [i] leverages pair-wised logistic regression algorithm, while row [ii, iii] leverage XGBoost LambdaMART algorithm to rank candidates. We have the following observations. First, switching from logistic regression to XGBoost ranking algorithm (comparing row [ii] to row [i]), we find 7%+ improvement on F1 score and 6%+ on recall of phonetic errors with minor latency increase (i.e., 1.5%). The nonlinear structure and the nature of handling feature interaction effects in tree-based models (i.e., XGBoost) might be the reason that our method outperforms linear ranking models (e.g., logistic regression). Second, adding

row	hyperparameter		performance		
	algorithm	ph-dist	F1	ph-recall	TP99
[i]	logistic regression	no	-	-	13
[ii]	XGBoost	no	+7.2%	+6.3%	+1.5%
[iii]	XGBoost	yes	+10.6%	+8.9%	+3.8%

Table 2: Candidate ranking ablation study (c. row[i])

the phonetic distance feature contributes to 3%+ F1 and 2%+ phonetic error recall improvement (comparing row [iii] to row [ii]). This supports our motivation that the phonetic distance feature tends to rank the correct candidate of a phonetically misspelled query higher, improving both the overall F1 score and recall of phonetic errors.

Adopting XGBoost LambdaMART algorithm with phonetic distance feature in the candidate ranking stage, we evaluate the hybrid candidate generation performance in Table 3 (row [i] in Table 3 is row [iii] in Table 2). Columns "ph-s" and "ed-s" represent the size of phonetic mapping and edit-distance based token candidates. Column "beam-s" is the beam search size and "ed-th" is the maximum edit distance allowed for edit-distance based token candidates. Phonetic mapping is created based on DM phonetic algorithm, limiting the top 10 candidates to have same DM phonetic key and allowing maximum 1 edit distance of the phonetic encoding key for the remaining candidates. Row [i] serves as a baseline model without phonetic mapping based token candidates. Row [ii] shows the impact of maximum edit distance setting on edit-distance based token candidates. We find that enlarging the maximum edit distance leads to high latency (371%+ increase) by comparing row [ii] with row [i], but with minor improvement on F1 and phonetic error recall. This indicates resolving phonetic errors (with distant edit distance to the correction) by directly searching the correct spelling within the edit distance threshold is not feasible. Row [iii-vi] shows the impact of adjusting different sizes of phonetic mapping and edit-distance based token candidates. We find that increasing edit-distance based token candidate size (comparing row [iii] to row [i]) leads to minor improvement (F1 + 0.6%, ph-recall +1.9%), but complementing token candidates by adding phonetic-mapping based candidates outperforms baseline by 6%+ on F1 and 5%+ on phonetic error recall (comparing row [iv] to row [i]). If phonetic mapping is the only source of token candidates (comparing row [v] to row [i]), there exists

row	hyperparameter				performance		
	ph-s	ed-s	beam-s	ed-th	F1	ph-recall	TP99
[i]	0	20	12	2	-	-	13.5
[ii]	0	20	12	3	+0.4%	+0.2%	+371%
[iii]	0	40	12	2	+0.6%	+1.9%	+16.3%
[iv]	20	20	12	2	+6.2%	+5.1%	+22.2%
[v]	20	0	12	2	-11.2%	-16.8%	-1.5%
[vi]	40	0	12	2	-6.6%	-8.8%	+23.7%
[vii]	20	20	8	2	**+6.2%**	**+5.4%**	**-3%**
[viii]	20	20	1	2	-5.9%	-8.8%	-20.7%

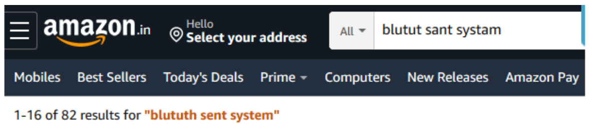
Table 3: Candidate generation ablation study (c. row[i])

a performance regression on both F1 and phonetic error recall. Furthermore, enlarging the phonetic mapping (row [vi]) is not an optimal solution considering its worse performance than hybrid sources (row [iv]). Then, we select the beam search size (row [vii,viii]) to control the similar TP99 for best F1 and phonetic error recall, compared with baseline (row [i]). The setting in row [vii] achieves 6%+ F1 score and 5%+ phonetic error recall improvement with flat latency. Aggregating the phonetic feature's impact (compare row [vii] in Table 3 to row [ii] in Table 2), the spelling correction system improves the F1 score by 9%+ and phonetic error recall by 8%+ by integrating phonetics into hybrid candidate generation and candidate ranking steps.

We adopt row [i] in Table 2 and row [vii] in Table 3 as baseline speller and phonetic speller for online A/B testing. We find significant business metrics gain without additional latency cost. Moreover, Figure 3 shows an example that the phonetic speller is able to resolve the irrelevant and/or sparse results based on the search query: "blutut sant systam". That is, there are only 82 search results returned by the baseline speller, while more than 9000 results are shown after applying the phonetic speller.

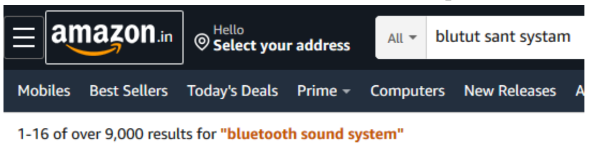
4 Conclusions and Future Work

In this work, we developed a generalized spelling correction system integrating phonetics into both hybrid candidate generation and candidate ranking stages on E-commerce domain. We demonstrated that our proposed phonetic solution improves more than 9% on F1 score and 8% on recall of phonetic errors without additional latency cost. This solution can be applied to any automatic spelling correction system with candidate generation or ranking stage. Online A/B testing showed positive business



1-16 of 82 results for "blututh sent system"

(a) Search results based on baseline speller



1-16 of over 9,000 results for "bluetooth sound system"

(b) Search results based on phonetic speller

Figure 3: Baseline speller vs. phonetic speller for search query: "blutut sant systam"

metrics while reducing sparse/irrelevant search results. Future directions include applying similar phonetic integrating ideas to other spelling correction frameworks (Jayanthi et al., 2020; Park et al., 2021) considering the growing popularity of the use of encoder-decoder deep learning architectures.

References

Kevin Atkinson. 2006. Gnu aspell 0.60. 4.

Eric Brill and Robert C Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th annual meeting of the association for computational linguistics*, pages 286–293.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Kenneth W Church and William A Gale. 1991. Probability scoring for spelling correction. *Statistics and Computing*, 1(2):93–103.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

Fred J Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171–176.

RE Gorin, Pace Willisson, Walt Buehring, Geoff Kuenning, et al. 1971. Ispell, a free software package for spell checking files. *The UNIX community*.

Daniel Hládek, Ján Staš, and Matúš Pleva. 2020. Survey of automatic spelling correction. *Electronics*, 9(10):1670.

Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. Neuspell: A neural spelling correction toolkit. *arXiv preprint arXiv:2010.11085*.

Daniel Jurafsky and James H Martin. 2008. *Speech & language processing (2nd ed.)*. Prentice Hall.

Karen Kukich. 1992. Techniques for automatically correcting words in text. *Acm Computing Surveys (CSUR)*, 24(4):377–439.

Eric Mays, Fred J Damerau, and Robert L Mercer. 1991. Context based spelling correction. *Information Processing & Management*, 27(5):517–522.

Chanjun Park, Kuekyeng Kim, YeongWook Yang, Minho Kang, and Heuiseok Lim. 2021. Neural spelling correction: translating incorrect sentences to correct sentences for multimedia. *Multimedia Tools and Applications*, 80(26):34591–34608.

Lawrence Philips. 2000. The double metaphone search algorithm. *C/C++ users journal*, 18(6):38–43.

Xu Sun, Jianfeng Gao, Daniel Micol, and Chris Quirk. 2010. Learning phrase-based spelling error models from clickthrough data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 266–274.

Valeriy S Vykhovanets, J Du, and SA Sakulin. 2020. An overview of phonetic encoding algorithms. *Automation and Remote Control*, 81(10):1896–1910.

Casey Whitelaw, Ben Hutchinson, Grace Chung, and Ged Ellis. 2009. Using the web for language independent spellchecking and autocorrection. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 890–899.