Correcting Keyboard Layout Errors and Homoglyphs in Queries

Derek Barnes
debarnes@ebay.com

Mahesh Joshi
mahesh.joshi@ebay.com

Hassan Sawaf
hsawaf@ebay.com

eBay Inc., 2065 Hamilton Ave, San Jose, CA, 95125, USA

Abstract

Keyboard layout errors and homoglyphs in cross-language queries impact our ability to correctly interpret user information needs and offer relevant results. We present a machine learning approach to correcting these errors, based largely on character-level n-gram features. We demonstrate superior performance over rule-based methods, as well as a significant reduction in the number of queries that yield null search results.

1 Introduction

The success of an eCommerce site depends on how well users are connected with products and services of interest. Users typically communicate their desires through search queries; however, queries are often incomplete and contain errors, which impact the quantity and quality of search results.

New challenges arise for search engines in cross-border eCommerce. In this paper, we focus on two cross-linguistic phenomena that make interpreting queries difficult: **(i) Homoglyphs**: (Miller, 2013): Tokens such as "саse" (underlined letters Cyrillic), in which users mix characters from different character sets that are visually similar or identical. For instance, English and Russian alphabets share homoglyphs such as c, a, e, o, y, k, etc. Although the letters are visually similar or in some cases identical, the underlying character codes are different. **(ii) Keyboard Layout Errors (KLEs)**: (Baytin et al., 2013): When switching one's keyboard between language modes, users at times enter terms in the wrong character set. For instance, "чехол шзфв" may appear to be a Russian query. While "чехол" is the Russian word for "case", "шзфв" is actually the user's attempt to enter the characters "ipad" while leaving their keyboard in Russian language mode. Queries containing KLEs or homoglyphs are unlikely to produce any search results, unless the intended ASCII sequences can be recovered. In a test set sampled from Russian/English queries with null (i.e. empty) search results (see Section 3.1), we found approximately 7.8% contained at least one KLE or homoglyph.

In this paper, we present a machine learning approach to identifying and correcting query tokens containing homoglyphs and KLEs. We show that the proposed method offers superior accuracy over rule-based methods, as well as significant improvement in search recall. Although we focus our results on Russian/English queries, the techniques (particularly for KLEs) can be applied to other language pairs that use different character sets, such as Korean-English and Thai-English.

2 Methodology

In cross-border trade at eBay, multilingual queries are translated into the inventory's source language prior to search. A key application of this, and the focus of this paper, is the translation of Russian queries into English, in order to provide Russian users a more convenient interface to English-based inventory in North America. The presence of KLEs and homoglyphs in multilingual queries, however, leads to poor query translations, which in turn increases the incidence of null search results. We have found that null search results correlate with users exiting our site.

In this work, we seek to correct for KLEs and homoglyphs, thereby improving query translation, reducing the incidence of null search results, and increasing user engagement. Prior to translation and search, we preprocess multilingual queries by identifying and transforming KLEs and homoglyphs as follows (we use the query "чехол шзфв 2 new" as a running example):

(a) Tag Tokens: label each query token

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 621–626,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

with one of the following semantically motivated classes, which identify the user's information need: (i) E: a token intended as an English search term; (ii) R: a Cyrillic token intended as a Russian search term; (iii) K: A KLE, e.g. "шзфв" for the term "ipad". A token intended as an English search term, but at least partially entered in the Russian keyboard layout; (iv) H: A Russian homoglyph for an English term, e.g. "вмw" (underlined letters Cyrillic). Employs visually similar letters from the Cyrillic character set when spelling an intended English term; (v) A: Ambiguous tokens, consisting of numbers and punctuation characters with equivalent codes that can be entered in both Russian and English keyboard layouts. Given the above classes, our example query "чехол шзфв 2 new" should be tagged as "R K A E".

(b) Transform Queries: Apply a deterministic mapping to transform KLE and homoglyph tokens from Cyrillic to ASCII characters. For KLEs the transformation maps between characters that share the same location in Russian and English keyboard layouts (e.g. ф → a, ы → s). For homoglyphs the transformation maps between a smaller set of visually similar characters (e.g. е → e, м → m). Our example query would be transformed into "чехол ipad 2 new".

(c) Translate and Search: Translate the transformed query (into "case ipad 2 new" for our example), and dispatch it to the search engine.

In this paper, we formulate the token-level tagging task as a standard multiclass classification problem (each token is labeled independently), as well as a sequence labeling problem (a first order conditional Markov model). In order to provide end-to-end results, we preprocess queries by deterministically transforming into ASCII the tokens tagged by our model as KLEs or homoglyphs. We conclude by presenting an evaluation of the impact of this transformation on search.

2.1 Features

Our classification and sequence models share a common set of features grouped into the following categories:

2.1.1 Language Model Features

A series of 5-gram, character-level language models (LMs) capture the structure of different types of words. Intuitively, valid Russian terms will have high probability in Russian LMs. In contrast, KLEs or homoglyph tokens, despite appearing on the surface to be Russian terms, will generally have low probability in the LMs trained on valid Russian words. Once mapped into ASCII (see Section 2 above), however, these tokens tend to have higher probability in the English LMs. LMs are trained on the following corpora:

English and Russian Vocabulary: based on a collection of open source, parallel English/Russian corpora ($\sim$50M words in all).

English Brands: built from a curated list of 35K English brand names, which often have distinctive linguistic properties compared with common English words (Lowrey et al., 2013).

Russian Transliterations: built from a collection of Russian transliterations of proper names from Wikipedia (the Russian portion of guessed-names.ru-en made available as a part of WMT 2013[1]).

For every input token, each of the above LMs fires a real-valued feature — the negated log-probability of the token in the given language model. Additionally, for tokens containing Cyrillic characters, we consider the token's KLE and homoglyph ASCII mappings, where available. For each mapping, a real-valued feature fires corresponding to the negated log-probability of the mapped token in the English and Brands LMs. Lastly, an equivalent set of LM features fires for the two preceding and following tokens around the current token, if applicable.

2.1.2 Token Features

We include several features commonly used in token-level tagging problems, such as case and shape features, token class (such as letters-only, digits-only), position of the token within the query, and token length. In addition, we include features indicating the presence of characters from the ASCII and/or Cyrillic character sets.

2.1.3 Dictionary Features

We incorporate a set of features that indicate whether a given lowercased query token is a member of one of the lexicons described below.

UNIX: The English dictionary shipped with CentOS, including $\sim$480K entries, used as a lexicon of common English words.

BRANDS: An expanded version of the curated list of brand names used for LM features. Includes

[1] www.statmt.org/wmt13/translation-task.html#download

~58K brands.

PRODUCT TITLES: A lexicon of over 1.6M entries extracted from a collection of 10M product titles from eBay's North American inventory.

QUERY LOGS: A larger, in-domain collection of approximately 5M entries extracted from ~100M English search queries on eBay.

Dictionary features fire for Cyrillic tokens when the KLE and/or homoglyph-mapped version of the token appears in the above lexicons. Dictionary features are binary for the Unix and Brands dictionaries, and weighted by relative frequency of the entry for the Product Titles and Query Logs dictionaries.

3 Experiments

3.1 Datasets

The following datasets were used for training and evaluating the baseline (see Section 3.2 below) and our proposed systems:

Training Set: A training set of 6472 human-labeled query examples (17,239 tokens).

In-Domain Query Test Set: A set of 2500 Russian/English queries (8,357 tokens) randomly selected from queries with null search results. By focusing on queries with null results, we emphasize the presence of KLEs and homoglyphs, which occur in 7.8% of queries in our test set.

Queries were labeled by a team of Russian language specialists. The test set was also independently reviewed, which resulted in the correction of labels for 8 out of the 8,357 query tokens.

Although our test set is representative of the types of problematic queries targeted by our model, our training data was not sampled using the same methodology. We expect that the differences in distributions between training and test sets, if anything, make the results reported in Section 3.3 somewhat pessimistic[2].

3.2 Dictionary Baseline

We implemented a rule-based baseline system employing the dictionaries described in Section 2.1.3. In this system, each token was assigned a class $k \in \{\mathtt{E}, \mathtt{R}, \mathtt{K}, \mathtt{H}, \mathtt{A}\}$ using a set of rules: a token among a list of 101 Russian stopwords[3] is tagged as R. A token containing only ASCII characters is labeled as A if all characters are common to English and Russian keyboards (i.e. numbers and some punctuation), otherwise E. For tokens containing Cyrillic characters, KLE and homoglyph-mapped versions are searched in our dictionaries. If found, K or H are assigned. If both mapped versions are found in the dictionaries, then either K or H is assigned probabilistically[4]. In cases where neither mapped version is found in the dictionary, the token assigned is either R or A, depending on whether it consists of purely Cyrillic characters, or a mix of Cyrillic and ASCII, respectively.

Note that the above tagging rules allow tokens with classes E and A to be identified with perfect accuracy. As a result, we omit these classes from all results reported in this work. We also note that this simplification applies because we have restricted our attention to the Russian → English direction. In the bidirectional case, ASCII tokens could represent either English tokens or KLEs (i.e. a Russian term entered in the English keyboard layout). We leave the joint treatment of the bidirectional case to future work.

Tag	Prec	Recall	F1
K	.528	.924	.672
H	.347	.510	.413
R	.996	.967	.982

Table 1: Baseline results on the test set, using UNIX, BRANDS, and the PRODUCT TITLES dictionaries.

We experimented with different combinations of dictionaries, and found the best combination to be UNIX, BRANDS, and PRODUCT TITLES dictionaries (see Table 1). We observed a sharp decrease in precision when incorporating the QUERY LOGS dictionary, likely due to noise in the user-generated content.

Error analysis suggests that shorter words are the most problematic for the baseline system[5]. Shorter Cyrillic tokens, when transformed from Cyrillic to ASCII using KLE or homoglyph mappings, have a higher probability of spuriously mapping to valid English acronyms, model IDs, or short words. For instance, Russian car brand "ваз" maps across keyboard layouts to "dfp",

[2] As expected, cross-validation experiments on the training data (for parameter tuning) yielded results slightly higher than the results reported in Section 3.3, which use a held-out test set

[3] Taken from the Russian Analyzer packaged with Lucene — see lucene.apache.org.

[4] We experimented with selecting K or H based on a prior computed from training data; however, results were lower than those reported, which use random selection.

[5] Stopwords are particularly problematic, and hence excluded from consideration as KLEs or homoglyphs.

	Tag	Classification			Sequence		
		P	R	F1	P	R	F1
LR	K	.925	.944	.935	.915	.934	.925
	H	.708	.667	.687	.686	.686	.686
	R	.996	.997	.996	.997	.996	.997
RF	K	.926	.949	.937	.935	.949	.942
	H	.732	.588	.652	.750	.588	.659
	R	.997	.997	.997	.996	.998	.997

Table 2: Classification and sequence tagging results on the test set

a commonly used acronym in product titles for "Digital Flat Panel". Russian words "муки" and "рук" similarly map by chance to English words "verb" and "her".

A related problem occurs with product model IDs, and highlights the limits of treating query tokens independently. Consider Cyrillic query "БМВ е46". The first token is a Russian transliteration for the BMW brand. The second token, "е46", has three possible interpretations: i) as a Russian token; ii) a homoglyph for ASCII "e46"; or iii) a KLE for "t46". It is difficult to discriminate between these options without considering token context, and in this case having some prior knowledge that e46 is a BMW model.

3.3 Machine Learning Models

We trained linear classification models using logistic regression (LR)[6], and non-linear models using random forests (RFs), using implementations from the Scikit-learn package (Pedregosa et al., 2011). Sequence models are implemented as first order conditional Markov models by applying a beam search ($k = 3$) on top of the LR and RF classifiers. The LR and RF models were tuned using 5-fold cross-validation results, with models selected based on the mean F1 score across R, K, and H tags.

Table 2 shows the token-level results on our in-domain test set. As with the baseline, we focus the model on disambiguating between classes R, K and H. Each of the reported models performs significantly better than the baseline (on each tag), with statistical significance evaluated using McNemar's test. The differences between LR and RF models, as well as sequence and classification variants, however, are not statistically significant. Each of the machine learning models achieves a query-level accuracy score of roughly 98% (the LR sequence model achieved the lowest with 97.78%, the RF sequence model the highest with 97.90%).

Our feature ablation experiments show that the majority of predictive power comes from the character-level LM features. Dropping LM features results in a significant reduction in performance (F1 scores .878 and .638 for the RF Sequence model on classes K and H). These results are still significantly above the baseline, suggesting that token and dictionary features are by themselves good predictors. However, we do not see a similar performance reduction when dropping these feature groups.

We experimented with lexical features, which are commonly used in token-level tagging problems. Results, however, were slightly lower than the results reported in this section. We suspect the issue is one of overfitting, due to the limited size of our training data, and general sparsity associated with lexical features. Continuous word presentations (Mikolov et al., 2013), noted as future work, may offer improved generalization.

Error analysis for our machine learning models suggests patterns similar to those reported in Section 3.2. Although errors are significantly less frequent than in our dictionary baseline, shorter words still present the most difficulty. We note as future work the use of word-level LM scores to target errors with shorter words.

3.4 Search Results

Recall that we translate multilingual queries into English prior to search. KLEs and homoglyphs in queries result in poor query translations, often leading to null search results.

To evaluate the impact of KLE and homoglyph correction, we consider a set of 100k randomly selected Russian/English queries. We consider the subset of queries that the RF or baseline models predict as containing a KLE or homoglyph. Next, we translate into English both the original query, as well as a transformed version of it, with KLEs and homoglyphs replaced with their ASCII mappings. Lastly, we execute independent searches using original and transformed query translations.

Table 3 provides details on search results for original and transformed queries. The baseline model transforms over 12.6% of the 100k queries. Of those, 24.3% yield search results where the unmodified queries had null search results (i.e. Null → Non-null). In 20.9% of the cases, however, the

[6] Although CRFs are state-of-the-art for many tagging problems, in our experiments they yielded results slightly lower than LR or RF models.

transformations are destructive (i.e. Non-null → Null), and yield null results where the unmodified query produced results.

Compared with the baseline, the RF model transforms only 7.4% of the 100k queries; a fraction that is roughly in line with the 7.8% of queries in our test set that contain KLEs or homoglyphs. In over 42% of the cases (versus 24.3% for the baseline), the transformed query generates search results where the original query yields none. Only 4.81% of the transformations using the RF model are destructive; a fraction significantly lower than the baseline.

Note that we distinguish here only between queries that produce null results, and those that do not. We do not include queries for which original and transformed queries both produce (potentially differing) search results. Evaluating these cases requires deeper insight into the relevance of search results, which is left as future work.

	Baseline	**RF model**
#Transformed	12,661	7,364
Null → Non-Null	3,078 (24.3%)	**3,142 (42.7%)**
Non-Null → Null	2,651 (20.9%)	**354 (4.81%)**

Table 3: Impact of KLE and homoglyph correction on search results for 100k queries

4 Related Work

Baytin et al. (2013) first refer to keyboard layout errors in their work. However, their focus is on predicting the performance of spell-correction, not on fixing KLEs observed in their data. To our knowledge, our work is the first to introduce this problem and to propose a machine learning solution. Since our task is a token-level tagging problem, it is very similar to the part-of-speech (POS) tagging task (Ratnaparkhi, 1996), only with a very small set of candidate tags. We chose a supervised machine learning approach in order to achieve maximum precision. However, this problem can also be approached in an unsupervised setting, similar to the method Whitelaw et al. (2009) use for spelling correction. In that setup, the goal would be to directly choose the correct transformation for an ill-formed KLE or homoglyph, instead of a tagging step followed by a deterministic mapping to ASCII.

5 Conclusions and Future Work

We investigate two kinds of errors in search queries: keyboard layout errors (KLEs) and homoglyphs. Applying machine learning methods, we are able to accurately identify a user's intended query, in spite of the presence of KLEs and homoglyphs. The proposed models are based largely on compact, character-level language models. The proposed techniques, when applied to multilingual queries prior to translation and search, offer significant gains in search results.

In the future, we plan to focus on additional features to improve KLE and homoglyph discrimination for shorter words and acronyms. Although lexical features did not prove useful for this work, presumably due to data sparsity and overfitting issues, we intend to explore the application of continuous word representations (Mikolov et al., 2013). Compared with lexical features, we expect continuous representations to be less susceptible to overfitting, and to generalize better to unknown words. For instance, using continuous word representations, Turian et al. (2010) show significant gains for a named entity recognition task.

We also intend on exploring the use of features from in-domain, word-level LMs. Word-level features are expected to be particularly useful in the case of spurious mappings (e.g. "ваз" vs. "dfp" from Section 3.2), where context from surrounding tokens in a query can often help in resolving ambiguity. Word-level features may also be useful in re-ranking translated queries prior to search, in order to reduce the incidence of erroneous query transformations generated through our methods. Finally, our future work will explore KLE and homoglyph correction bidirectionally, as opposed to the unidirectional approach explored in this work.

Acknowledgments

We would like to thank Jean-David Ruvini, Mike Dillinger, Saša Hasan, Irina Borisova and the anonymous reviewers for their valuable feedback. We also thank our Russian language specialists Tanya Badeka, Tatiana Kontsevich and Olga Pospelova for their support in labeling and reviewing datasets.

References

Alexey Baytin, Irina Galinskaya, Marina Panina, and Pavel Serdyukov. 2013. Speller performance prediction for query autocorrection. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, pages 1821–1824.

Tina M. Lowrey, Larry J. Shrum, and Tony M. Dubitsky. 2013. The Relation Between Brand-name Linguistic Characteristics and Brand-name Memory. *Journal of Advertising*, 32(3):7–17.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tristan Miller. 2013. Russian–English Homoglyphs, Homographs, and Homographic Translations. *Word Ways: The Journal of Recreational Linguistics*, 46(3):165–168.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Adwait Ratnaparkhi. 1996. A Maximum Entropy Model for Part–of–Speech Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.

Casey Whitelaw, Ben Hutchinson, Grace Y. 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.