Impacts of Misspelled Queries on Translation and Product Search

Greg Hanneman and **Natawut Monaikul** and **Taichi Nakatani**
Amazon
{ghannema, natawm, taichina}@amazon.com

Abstract

Machine translation is used in e-commerce to translate second-language queries into the primary language of the store, to be matched by the search system against the product catalog. However, many queries contain spelling mistakes. We first present an analysis of the spelling-robustness of a population of MT systems, quantifying how spelling variations affect MT output, the list of returned products, and ultimately user behavior. We then present two sets of practical experiments illustrating how spelling-robustness may be specifically improved. For MT, reducing the number of BPE operations significantly improves spelling-robustness in six language pairs. In end-to-end e-commerce, the inclusion of a dedicated spelling correction model, and the augmentation of that model's training data with language-relevant phenomena, each improve robustness and consistency of search results.

1 Introduction

In many e-commerce settings that support users across multiple language backgrounds, machine translation (MT) is used to translate search queries from the user's preferred language into the primary language of the store in order to match those search queries against the product catalog. Search queries, however, are prone to spelling mistakes (e.g., typos or misspelled words) or, more generally, spelling variations (e.g., leaving out diacritics as shorthand or due to keyboard limitations).

Robustness of MT systems in the face of such "noisy" input has been a persistent focus of study in both statistical and neural translation. However, most work has considered only MT performance in isolation: if standard MT metric scores and/or robustness scores increase, then the MT system is more robust and the improvement is successful.

We consider by contrast in this paper the multilingual e-commerce setting: a specific application of MT where the system's output is not the end of the story. Consider a product search system that delivers search results based on a user's input that has first undergone MT. The quality of the search results then partially depends on the MT system's robustness to misspellings or typographical preferences in the original input. An additional spelling correction step can also precede MT to further mitigate the impacts of noisy input. We illustrate that targeted improvements to the MT system and the spelling correction system can improve the robustness of the product search system as a whole to spelling variations.

Our contributions in this paper are as follows:

Using test sets of rightly and wrongly spelled search queries in multiple secondary languages, along with targeted metrics of MT robustness and search result difference (Section 3), we quantify how the spelling-robustness of industrial-grade MT systems impacts the e-commerce experience. We show that MT can implicitly correct for spelling errors up to half the time, that the properties of the retrieval system also impact the search results, but that correctly spelled queries still tend to lead to better shopping outcomes (Section 4).

Via modeling experiments, we demonstrate that MT's spelling-robustness can be significantly improved by reducing the number of BPE operations (Section 5). We measure the effect of a dedicated spelling correction model, and we improve its contribution as well by augmenting its training data with language-relevant phenomena such as missing diacritics (Section 6).

2 Related Work

Our e-commerce scenario, where users' search queries are automatically translated to match the language of the product catalog, is fundamentally similar to the setups presented by Guha and Heger (2014) and Yao et al. (2020), or the QT alternative

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13907–13920

studied by Saleh and Pecina (2020).

The most straightforward method of measuring MT robustness is via standard reference-based metric scores — such as BLEU (Papineni et al., 2002), chrF (Popović, 2015), or TER (Snover et al., 2006) — on a test set of interest. In this formulation, robustness is judged by comparing the MT system's score when translating a damaged version of the input against its score on the corresponding "clean" text. The approach is popular for evaluating both SMT and NMT performance in the face of spelling errors (Bertoldi et al., 2010; Belinkov and Bisk, 2018), social-media content (Vaibhav et al., 2019), and non-parallel training data (Khayrallah and Koehn, 2018). We follow Niu et al. (2020) in dispreferring this approach.

Instead, researchers have proposed targeted robustness metrics that go beyond the standard scoring of an MT output against a reference translation (Michel et al., 2019; Niu et al., 2020; Bergmanis et al., 2020). Our analyses use several of the same metrics, though sometimes with minor differences to accommodate our particular scenario. See Section 3.2 for details.

Prior work has demonstrated how MT robustness may be improved, either via data augmentation that targets different kinds of spelling mistakes or "noise" (Heigold et al., 2018; Belinkov and Bisk, 2018; Karpukhin et al., 2019; Vaibhav et al., 2019; Bergmanis et al., 2020) or via specific modeling improvements (Bertoldi et al., 2010; Belinkov and Bisk, 2018; Heigold et al., 2018; Michel et al., 2019; Niu et al., 2020). The BPE-based case study we present in Section 5 contributes to the modeling line of work. Our work in Section 6 meanwhile follows the data augmentation approach.

Of special note to Section 5, Post and Duh (2019) tried adjusting the number of BPE operations, among other hyperparameters, in MT systems aimed at the translation of user-generated texts. Their robustness application was therefore much more general than our present study of spelling mistakes, and it was evaluated from the point of view of overall MT performance only. Our experiments are more narrowly focused, and our evaluation more targeted.

3 General Setup

3.1 Test Sets

We sourced wrongly spelled search queries by taking a sample of historical traffic from each of 25 secondary languages supported by a multilingual e-commerce store. Samples were biased towards containing spelling mistakes according to in-house automatic spelling correction models. (See Appendix A for construction details.) Note that these wrongly spelled search queries consist of both unintentional and intentional misspellings, ranging from typos or homophone replacement to "text-speak" (as in Spanish *q* instead of *que*). Some languages are represented more than once among the 25 if they are supported as secondary languages for multiple primary languages (such as English queries to be translated into German vs. Japanese), but we treat these as separate test sets: we expect the distribution of queries to be different in each case, and the MT models used are different as well.

From each traffic stream we selected 11,000 unique queries to send to human annotators. The annotators were asked to perform three main tasks for each query: first, to verify that it was in the expected secondary language and that the query's intent was understandable; second, to record whether the query was misspelled or not; and third, to provide a correctly spelled version if necessary. The result of this human annotation provided us with a variable number of rightly and wrongly spelled query pairs for each secondary traffic stream. Final test set sizes range from 875 query pairs up to 6,242, with an overall average of 3,432. See Appendix A for the complete list.

3.2 Robustness Metrics

Formally, each entry in our test sets consists of a rightly spelled input query x_r and a wrongly spelled version x_w; we also assume their corresponding machine translations y_r and y_w. Our goal is to quantify the spelling-robustness of each MT system in question.

Niu et al. (2020) define a reference-free Consistency metric based on the harmonic mean of two directional invocations of a standard MT metric. (Bidirectionality ensures that the result of comparing two translations is symmetric.) In the original work, the authors base the Consistency score on BLEU. We prefer chrF over BLEU throughout this work; the use of chrF instead in Equation 1 is the only change.[1] Consistency, designed to evaluate

[1]We implement robustness metrics (and later evaluate MT results) using chrF instead of BLEU for several reasons. First, the average length of a search query in our test sets is 3.3 raw words, making BLEU's four-gram statistics sparse and unreliable even at the corpus level. Perhaps most importantly, the

two output variants against each other, is an intuitive fit for our use case.

$$\text{Consistency} = \frac{2 \cdot \text{chrF}(y_w, y_r) \cdot \text{chrF}(y_r, y_w)}{\text{chrF}(y_w, y_r) + \text{chrF}(y_r, y_w)} \quad (1)$$

Michel et al. (2019) introduce a measurement that explicitly compares the effect of a noisy input–output pair on the source versus target side, though the metric still expects a reference translation. One term quantifies the relative fraction of the MT metric score lost by the noisy translation; it is then added to a term measuring the similarity between the clean versus noisy source side in order to form an overall Success metric. In our reference-less setting, we compute a Pseudo-Success metric by replacing the reference with y_r, which gives the form in Equation 2.

$$\begin{aligned}\text{Pseudo-Success} = &\max\left(1 - \text{chrF}(y_w, y_r), 0\right) \\ &+ \text{chrF}(x_w, x_r) \end{aligned} \quad (2)$$

The presence of the spelling mistake in the input (i.e., having to translate x_w instead of x_r) is deemed a successful adversarial attack if Success > 1.

Finally, we introduce our own simple MT robustness measure that tracks whether the two MT outputs y_w and y_r are equal. This is a binary metric (Equation 3) that we believe is particularly suitable to our search-query use case: if the two MT outputs are equal, then the MT system has successfully "corrected" or regularized out any downstream effect of the source-language spelling mistake.

$$\text{Equal} = (y_w \stackrel{?}{=} y_r) \quad (3)$$

3.3 Product Search Metric

In multilingual e-commerce, a query and its corresponding translation yield a set of product search results. We represent these results as the ranked list of the top 16 product IDs that would be displayed. Formally, we define P_r as this set of top products as a function of the rightly-spelled query x_r, its translation y_r, its input language ℓ, and the e-commerce store s — and analogously for P_w:

$$P_r = \text{Search}(x_r, y_r, \ell, s) \quad (4a)$$

$$P_w = \text{Search}(x_w, y_w, \ell, s) \quad (4b)$$

years since 2020 have led to an increased consensus in the MT field demonstrating chrF as a generally better lexical metric (Kocmi et al., 2021; Freitag et al., 2022). Finally, we believe that using chrF remains in keeping with the Consistency metric's original introduction and intent: Niu et al. (2020) note that it could be based on "any quality measurement metric."

To compare lists P_w and P_r, we use a variant of the Normalized Discounted Cumulative Gain (NDCG) score, which measures the quality of a hypothesized ranked list against a gold-standard reference list, with more weight assigned to the higher positions than lower ones. In our setting, we use P_r as the reference list and compute NDCG as

$$\text{NDCG} = \frac{\sum_{i=1}^{\min(|P_w|,|P_r|)} \frac{(P_w^{(i)} \stackrel{?}{\in} P_r)}{\log_2(i+1)}}{\sum_{i=1}^{|P_r|} \frac{1}{\log_2(i+1)}} \quad (5)$$

4 Analysis of MT Robustness

In this section we analyze the robustness of production-grade MT systems to spelling variations in the input, as exemplified by our test sets. Measures of MT robustness, difference in product search results, and difference in user behaviors track the effect of misspelled MT input through the rest of the e-commerce experience.

4.1 Procedure

We translated our 25 sets of rightly and wrongly spelled queries through two types of production-grade MT systems: the publicly available Amazon Translate service as of April 2023, and our own in-house MT systems supporting secondary-language search in e-commerce. Note that the former are generic systems aimed at a wide variety of MT applications, while the latter are trained specifically for translating search queries.

Given the MT inputs and outputs, we computed Consistency (Equation 1), Pseudo-Success (Equation 2), and Equal Output (Equation 3) for each MT system. Applicable metrics use the definition of chrF that is included with SacreBLEU 1.4.14 (Post, 2018). We also generated the top 16 search results for each query and computed the NDCG scores (Equation 5).

We apply no tokenization or normalization other than lowercasing the strings: capitalization is not a reliable signal when processing user-generated search queries. Our chrF-based robustness scores are aggregated from the segment level to the corpus level by adding up the underlying sufficient statistics (as in chrF itself) rather than by averaging the segment-level scores. The corpus-level Equal Output and NDCG scores are the mean values across all queries in the test set.

4.2 Misspelling's Effect on MT Output

We computed scores for each of our population of 50 MT systems according to each of the three spelling-robustness metrics.

It is first worth noting the extent to which these scores correlate with each other. For this we computed the Pearson linear correlation coefficient (ρ) for each pair of metrics. From most to least similar, the correlation between Consistency and Pseudo-Success is $\rho = -0.78$, between Consistency and Equal Output $\rho = 0.47$, and between Pseudo-Success and Equal Output $\rho = -0.33$. These three metrics, therefore, each seem to have a somewhat different view of spelling-robustness.[2]

The raw scores obtained by our 50 MT systems on these three metrics are shown in Figure 1, in the form of correlation plots depicting each possible pair of metrics at a time.

The MT systems illustrate a widely varying degree of robustness — a quantification that we do not believe has been presented in prior work. It is commonly assumed anecdotally that practical MT systems do learn how to cope with "some" degree of spelling variation, based on the range of misspellings illustrated in the system's training data along with the model's ability to generalize from that training. Here we find that the capability to produce the same output in the face of a spelling error ranges from 0.3% of our test cases up to 54.7%. Thus, spelling errors could be automatically "erased" up to half the time — or perhaps almost never.

Only seven of 50 MT systems score less than 1 on the Pseudo-Success metric when aggregated over all the examples in the applicable test set. That is, a spelling mistake still usually degrades the MT output more than the input and thus constitutes a "successful" attack.

4.3 Misspelling's Effect on Search Results

The fact that y_r and y_w differ does not necessarily mean that the *search results* will differ as well. Ideally, the sophisticated processes of matching and ranking relevant products will be able to implicitly "correct" or regularize translation variations where the user's shopping intent is still the same.

Indeed, there is much variation in the degree to which robustness in the MT output predicts robustness in the list of search results. Figure 2 illustrates the relationship between NDCG and each of Equal Output ($\rho = 0.96$), Consistency ($\rho = 0.62$), and Pseudo-Success ($\rho = -0.44$).

At a high level, MT systems that are more robust to spelling errors do tend to lead to search results that are more similar to the equivalent correctly spelled query, as we would expect. The search system's regularizing effect is however also clear. There is some overlap in search results even for MT systems whose output is almost always affected by a spelling error. Systems with the lowest NDCG scores interestingly tend to score near the middle of the range on Consistency and Pseudo-Success, which suggests that the spelling mistakes that most disrupt MT are not necessarily the same as the ones that most disrupt search.

4.4 Misspelling's Effect on Shopping Success

Our final analysis examines the effect of the whole misspelling–translation–search chain on online shoppers' experiences. We use customer behavior as a proxy for shopping experience and investigate if misspellings lead to changes in interaction with search results. We searched our e-commerce store logs for records containing any (x_w, y_w) or (x_r, y_r) pair from our test sets, along with whether any of the following actions was associated with each search: clicking on a result, adding a product to the shopping cart, or reformulating the query. These records are sourced over a one-year period to avoid seasonal effects.

Since our original test sets (x_w) were sourced over a different time period and then manually corrected, it is not guaranteed that the same queries appear in the logs during the analysis period — or that the machine translations made of them at different points throughout the year match the ones we created at a single point in time. Our extracted customer behavior data therefore tends to cover only a small fraction of each test set. Further, 98% of the data concerns rightly spelled queries, as spelling mistakes are relatively rare in general and the exact spelling mistakes of our test set even more so. To mitigate noise, we included only those (x_w, y_w) and (x_r, y_r) pairs that appeared at least five times in the logs.

[2] We initially included a standard MT metric in our reference-less scenario by computing the chrF of y_w against y_r as a pseudo-reference. However, we found that these "Pseudo-chrF" scores were numerically nearly identical to the Consistency metric ($\rho = 0.98$). This makes sense: our implementation of Consistency is simply the harmonic mean of two chrF scores, and chrF is itself based on the harmonic mean of character-level precision and recall. The choice of which string is used as the "hypothesis" versus "reference" has little impact.

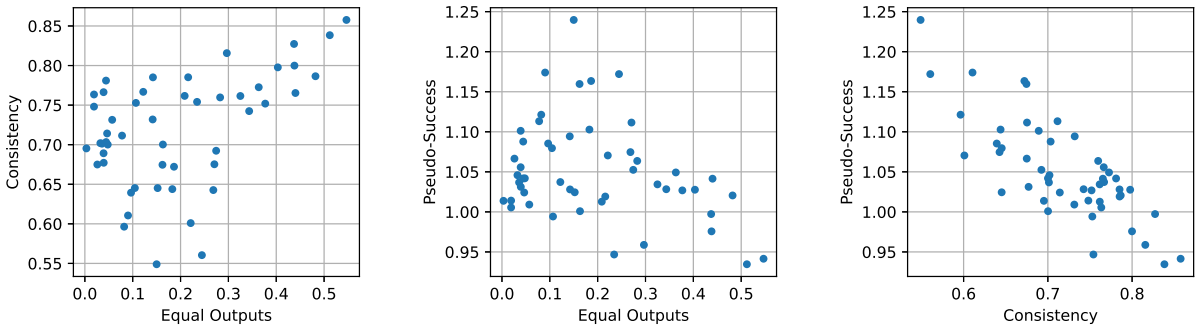

Figure 1: Scores of spelling-robustness for a population of 50 MT systems according to robustness metrics Equal Output, Consistency, and Pseudo-Success. Plots illustrate the correlation between each possible pair of metrics.

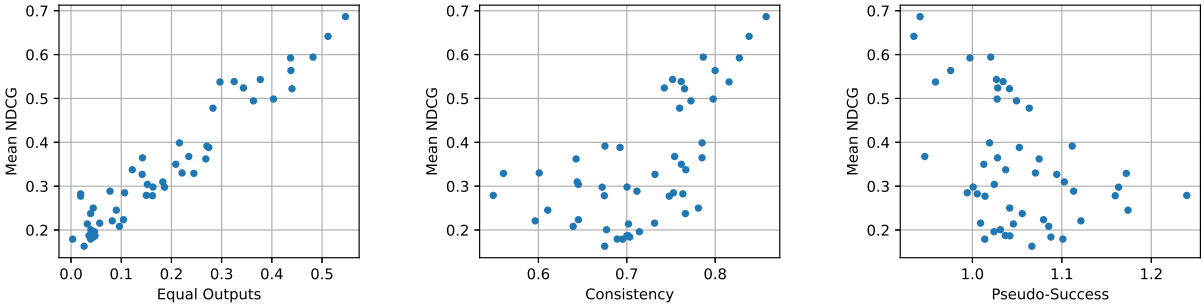

Figure 2: The power of using an MT system's spelling-robustness (Equal Output, Consistency, or Pseudo-Success) to predict the ensuing divergence in search results (NDCG) is variable.

To measure the impact of misspellings on shopping experience, we compared the rates of three customer behaviors for each available x_r and corresponding x_w — this shows, for example, how much more likely customers are to click on a product after searching x_r compared to x_w. We average these rates across all unique (x_w, x_r) pairs from our test sets appearing in the logs in each secondary language. We report in Table 1 only those secondary languages for which there were at least 90 such unique pairs.

We observed that, in most cases, shoppers are more likely to interact with search results and less likely to reformulate when the query was rightly spelled compared to when the query was wrongly spelled. For example, from the first row of Table 1, the rate at which customers shopping in French (with English as the primary language) clicked on a search result from a rightly spelled query is 2.17% higher than from a wrongly spelled query, and the rate at which customers reformulated their rightly spelled queries is 3.20% lower than their wrongly spelled queries. These results suggest that rightly spelled queries tend to generate search results that encourage customer interaction and decrease the need for reformulating searches com-

Language	Primary	Click	Add	Reform
fr-CA	en-CA	2.17	0.84	–3.20
cs-CZ	de-DE	–0.14	0.49	0.32
tr-TR	de-DE	1.44	0.36	–2.86
pt-PT	es-ES	2.08	1.80	–1.90
bn-IN	en-IN	12.48	1.01	–5.59
mr-IN	en-IN	7.07	0.93	–4.34
en-US	ja-JP	6.59	2.50	–9.86
en-GB	nl-NL	6.89	2.61	–10.82
es-MX	en-US	3.30	0.16	–5.32
he-IL	en-US	4.87	1.05	–10.89
ko-KR	en-US	8.61	2.06	–6.60
pt-BR	en-US	0.54	–0.10	–1.72
zh-TW	en-US	2.16	0.61	–10.08

Table 1: Differences in average Click, Add, and Reform(ulation) rates between rightly spelled and wrongly spelled queries from historical traffic data, in percentage points, for select secondary languages (with corresponding primary languages).

pared to wrongly spelled queries — in other words, misspellings indeed have an observable impact on the e-commerce shopping experience.

5 Improving MT Robustness with BPE

Having seen how the innate spelling-robustness of MT systems can vary widely, we now examine how a system's robustness can be explicitly improved.

Intuitively, an MT system may struggle to translate a spelling mistake because the input looks unusual compared to the bulk of the model's training data. Systems trained with byte-pair encoding (BPE) require their inputs to be broken down into a sequence of sub-word units that match the model's fixed vocabulary (Sennrich et al., 2016b). A familiar input word should be more often directly contained in the model's BPE vocabulary; a rare (e.g., misspelled) word would need to be broken down into a relatively larger number of sub-words.

Consider the English search query *samsung galaxy s21* and its misspelled counterpart *samsung gslaxy s21*. Using the same BPE model, these two strings are encoded as follows:

```
samsung galaxy s@@ 21
samsung gs@@ la@@ xy s@@ 21
```

The wrongly spelled query requires six tokens to the rightly spelled query's four.

We tested this intuition by building slates of MT system variants that differ in the number of BPE iterations ("operations") that were performed prior to model training. As the number of BPE operations — and thus the model's vocabulary — becomes smaller, rightly and wrongly spelled inputs should be expressed in more and more similar ways. In the limit, when the model vocabulary consists of single characters only, there should be no difference in the nature of sub-words needed to express the two different types of query, and almost no difference in the pre-processed query lengths.

5.1 Procedure

Our experimental MT systems are initially trained on publicly available WMT corpora before being fine-tuned on in-house data sets representing the search-query domain. We began with the most recently available WMT constrained training data for six language pairs: 2013 for Spanish–English (ES–EN); 2015 for French–English (FR–EN); 2020 for Tamil–English (TA–EN); and 2023 for Hebrew–English (HE–EN), English–German (EN–DE), and English–Japanese (EN–JA). These six cases were chosen to illustrate a range of training data sizes, language types, and directionality of English.

We limited data cleaning and corpus preparation steps since our focus is on comparing spelling-robustness instead of building competition-winning systems. Appendix B gives the details of component corpora used, clean-up applied, and the final training and dev set sizes for each language pair.

As our main experiment, we built a slate of system variants for each language pair using 32,000, 16,000, 8,000, 4,000, 2,000, and 0 BPE operations. Such diversity of vocabulary size leads to extreme differences among the systems in the *length* of the input/output strings for even the same line of original training data. We also have a 150× difference between the number of lines in the largest training corpus (EN–DE, 83M) versus the smallest (TA–EN, 553k). To keep the systems' training experiences more on par, we made the following hyperparameter adjustments:

- We scaled the maximum input length in proportion with the effect that a lower number of BPE operations had on the training data. In the baseline, 32,000 BPE operations is paired with a maximum input length of 100 tokens. Other variants used cutoffs 109 for 16,000 BPE, 122 for 8,000 BPE, 139 for 4,000 BPE, 161 for 2,000 BPE, and 374 for 0 BPE.[3]
- Our default checkpoint interval was 4,000 batches. Because of differences in the amount of training data, we increased the checkpoint interval to 8,000 in HE–EN and EN–DE and decreased it to 500 in TA–EN.

Remaining hyperparameters were set identically across all MT systems; see Appendix C for the complete configuration. Our models are Transformers (Vaswani et al., 2017), generally following the dimensions of Transformer-base except that we used 20 encoder layers and only two decoder layers (Hieber et al., 2020). All trainings were carried out with the Sockeye 3 toolkit (Hieber et al., 2022).

The minority of our fine-tuning data consists of human translations, while the larger part is made up of originally monolingual search queries collected in the target language and then automatically back-translated (Sennrich et al., 2016a). (We use for

[3] Empirically, the maximum input length for 0 BPE systems should have been 404, but we had to reduce the limit to 374 in order to avoid running out of RAM on our hardware platform.

back-translation a version of our base MT system with 32,000 BPE operations, but trained in the reverse direction, for each language pair.) The same fine-tuning training data was used for all the variant systems within a given language pair. It amounts to approximately 5.2 million lines for each pair — larger than the base training data in the case of TA–EN and EN–JA. Exact line counts are again in Appendix B. The fine-tuning dev set is a sample of 4,000 human-translated search queries.

5.2 Results

Our goal in this case study is to verify two claims: first, that varying of the number of BPE operations does *not* meaningfully reduce an MT system's general quality; second, that it *does* significantly improve the system's spelling-robustness.

Results related to the first claim are presented in column (a) of Table 2. Here we evaluate each fine-tuned system on an in-domain test set of 4,000 human-translated search queries, using chrF as a standard reference-based MT metric. We judge statistical significance relative to the baseline (32,000 BPE operations) according to paired bootstrap resampling (Koehn, 2004), using 1,000 resampled test sets equivalent in size to the original. Under this formulation, we claim that a system is better than the baseline (▲) if it scores higher on at least 95% of the resamples, that it is worse (▽) if the baseline scores higher on at least 95% of the resamples, and otherwise that it is equal (⇆).

While lowering the number of BPE operations *can* affect an MT system's general quality, we do not observe a strong overall pattern predicting if or in what direction it will do so — except on TA–EN, where smaller vocabulary may be more appropriate for modeling training data that is at least six times smaller than any of the other language pairs. Across all languages and variants, the change in BPE operations improves over the baseline in nine cases, regresses in eight, and does not statistically differ in 13.

For our second claim, we turn to the metrics of spelling-robustness previously defined in Section 3.2. For simplicity, we report only Equal Output and Consistency; the statistical significance of the Pseudo-Success scores is identical to Consistency in all but one instance. Columns (b) and (c) in Table 2 report the results.

Here we observe a much stronger pattern. According to Consistency, every variant system is significantly more spelling-robust than the baseline except for four of the five HE–EN cases. (A total of eight experimental builds are statistically equivalent according to Equal Output, including all five in HE–EN.) Robustness improvements tend to continue as the number of BPE operations is increasingly lowered: in a follow-up test, we computed statistical significance of the remaining models relative to the already improved 16,000 BPE variants. Fourteen of the smaller models remain significantly better, while two are worse and eight are tied.

Except for the HE–EN setting, we consider these results strong proof of the ability to intentionally improve the spelling-robustness of an MT system using simple hyperparameter settings.

Lang	BPE Ops	(a) chrF	(b) Equal	(c) Consist
ES–EN	32,000	67.4	0.211	0.724
	16,000	67.9 ▲	0.234 ▲	0.738 ▲
	8,000	67.1 ⇆	0.237 ▲	0.740 ▲
	4,000	66.4 ▽	0.227 ▲	0.744 ▲
	2,000	66.4 ▽	0.236 ▲	0.755 ▲
	0	69.2 ▲	0.299 ▲	0.789 ▲
FR–EN	32,000	62.6	0.291	0.737
	16,000	62.4 ⇆	0.294 ⇆	0.745 ▲
	8,000	62.4 ⇆	0.297 ⇆	0.748 ▲
	4,000	62.0 ▽	0.307 ▲	0.757 ▲
	2,000	62.1 ▽	0.330 ▲	0.766 ▲
	0	62.6 ⇆	0.376 ▲	0.803 ▲
HE–EN	32,000	70.7	0.279	0.711
	16,000	70.7 ⇆	0.287 ⇆	0.717 ⇆
	8,000	69.4 ▽	0.271 ⇆	0.708 ⇆
	4,000	68.5 ▽	0.270 ⇆	0.706 ⇆
	2,000	68.5 ▽	0.281 ⇆	0.718 ⇆
	0	70.7 ⇆	0.284 ⇆	0.729 ▲
TA–EN	32,000	48.9	0.048	0.660
	16,000	51.2 ▲	0.056 ▲	0.684 ▲
	8,000	52.5 ▲	0.068 ▲	0.702 ▲
	4,000	52.8 ▲	0.062 ▲	0.708 ▲
	2,000	52.3 ▲	0.067 ▲	0.708 ▲
	0	54.1 ▲	0.096 ▲	0.728 ▲
EN–DE	32,000	75.6	0.105	0.720
	16,000	76.0 ▲	0.122 ▲	0.734 ▲
	8,000	75.9 ⇆	0.123 ▲	0.736 ▲
	4,000	75.9 ⇆	0.130 ▲	0.741 ▲
	2,000	75.5 ⇆	0.128 ▲	0.747 ▲
	0	75.2 ⇆	0.144 ▲	0.751 ▲
EN–JA	32,000	62.1	0.160	0.584
	16,000	63.1 ▲	0.176 ▲	0.599 ▲
	8,000	62.7 ⇆	0.188 ▲	0.599 ▲
	4,000	62.6 ⇆	0.182 ▲	0.602 ▲
	2,000	62.4 ⇆	0.174 ▲	0.603 ▲
	0	59.3 ▽	0.155 ⇆	0.628 ▲

Table 2: chrF, Equal Output, and Consistency scores for fine-tuned systems with varying numbers of BPE operations. Symbols show whether each score is better than (▲), worse than (▽), or tied with (⇆) the 32,000 baseline.

6 Improving Search Robustness with Spelling Correction

While we can directly improve the robustness of an MT system to misspelled queries, in an e-commerce system, it is also possible to include an explicit spelling correction step in order to mitigate downstream impacts. We now explore the extent to which a spelling correction model can retain the performance of MT and search systems.

6.1 Procedure

We experimented with spelling correction models aimed at a Spanish-to-English search pipeline. The primary training data, composed of pairs of input queries and their target output forms, was again sourced from customer traffic guided by in-house spelling correction models, as well as from reformulated queries filtered for those that are likely to represent spelling corrections (similar to the approach taken by Hasan et al. (2015)). We extracted 183M pairs, with a mix of query pairs where the input and output are the same (i.e., the input query is already correct) and where the output is the corrected form of the input.

Additionally, to investigate if a targeted improvement to a spelling correction model can also improve the robustness of the search system it belongs to, we augment the primary training data with synthetically misspelled queries targeting a specific typographical phenomenon. In particular, diacritics in Spanish tend to be omitted in informal contexts (e.g., typing *electronico* instead of *electrónico*), so we extract all target queries in the training data that contain at least one letter with a diacritic and create input queries by removing all diacritics from the target queries. We then sampled 6.1M of these synthetic query pairs (3.3% of the primary training data size) to add to the training data. Dev sets of 50,000 query pairs were randomly sampled and extracted from training sets.

We trained BART models (Lewis et al., 2020) from scratch, one on the primary training data (our *baseline* model) and one on the augmented training data (our *augmented* model). Training was carried out with the Fairseq toolkit (Ott et al., 2019). Hyperparameters were the same for both models; see Appendix D for the complete list.

6.2 Results

The aim of the baseline spelling correction model is to mitigate the impact that misspellings have on both MT and search results, and the aim of the augmented model is to build even further on robustness. We first compared model performance on spelling correction with our test set of wrongly spelled Spanish queries (as described in Section 3.1), where a model output on a wrongly spelled input is marked as correct only if it exactly matches the target correction ignoring casing. We observed an accuracy of 39.4% for the baseline model and 41.2% for the augmented model, showing a general improvement in spelling correction ability after adding targeted synthetic data to the training data.

Test Set	Model	Equal	Consist	NDCG
wrongly spelled	none	0.211	0.724	0.401
	base	0.563	0.859	0.746
	aug	0.575	0.864	0.759
rightly spelled	none	1.000	1.000	1.000
	base	0.776	0.939	0.992
	aug	0.780	0.941	1.000

Table 3: Comparison of robustness metrics after translating wrongly and rightly spelled queries that were run through no spelling correction model, our baseline model, and our augmented model.

We then used our fine-tuned ES–EN MT model with 32,000 BPE operations (as described in Section 5.1) to translate the test-set outputs of each spelling correction model. We compared the impact each model had on MT outputs with the Equal Output and Consistency metrics, as well as the impact each model had on search results with NDCG, as described in Section 3.3. These results are given in the top half of Table 3.

We found that the addition of a spelling correction model strongly mitigates the negative impact that misspellings can have on MT and search results, as both the baseline and augmented models showed a significant increase in all three metrics compared to having no spelling correction model. This result was not guaranteed *a priori*, given the only moderate accuracy of the spelling models. We also found that adding the synthetic training data in the augmented model protected MT and search results more than the baseline model.

A spelling correction model can also incorrectly "correct" queries that are not misspelled, so introducing such a model may unintentionally have a negative impact on queries that are already spelled correctly. We thus ran the outputs of both spelling correction models on our test set of rightly spelled

queries through the same ES–EN MT model to measure their impact (bottom half of Table 3).

Results show that the incorrect corrections of both spelling correction models impact MT output, as neither achieved all Equal Outputs or perfect Consistency. However, we see that only the augmented model was able to fully retain NDCG performance. We note that because the query traffic stream is typically composed of many more rightly spelled queries than wrongly spelled queries, the effects of a spelling correction model on rightly spelled queries would be more pronounced online; thus, mechanisms to limit the downstream impacts of incorrect corrections through these targeted improvements is critical. Practical safeguards, such as adjusting the classification threshold of the model to have higher precision (at the cost of lower recall), can also be implemented to further decrease the likelihood of incorrect corrections in an actual deployment of such a model.

7 Conclusions

This work began by illustrating the surprising range of spelling-robustness in a population of MT systems. We demonstrated the utility of three targeted metrics for quantifying robustness, each offering a somewhat different view of how MT output is disrupted by the presence of spelling variations in the input. We also related the disruption in MT to the disruption in search results and user behavior in a cross-lingual e-commerce setting.

We then showed through practical experiments how reducing the number of BPE operations during MT training significantly improves spelling-robustness across five out of six language pairs while having a less systematic effect on overall translation quality. A second set of experiments demonstrated how a dedicated spelling correction model improves search robustness, furthered by a targeted improvement to the model through data augmentation.

8 Limitations

Our model-building experiments were focused only on the domain of e-commerce search queries. The successful results we report for improving the spelling-robustness of MT systems (Section 5) and the performance of spelling correction systems (Section 6) may not transfer to general-purpose models or to other domains.

Though our MT variants trained with 0 BPE operations gave the most spelling-robust results by a large margin (Table 2), we encountered several practical difficulties during their training and use. Because of the vastly increased input and output lengths, this fairly naïve implementation of character-based MT uses more RAM and is slower to run than our other variants, which may negate its benefits in certain use cases.

We did not compare our robustness-improving technique against any other previously published methods (Section 2) at the same time and under the same conditions. Thus, while our improvements in isolation remain valid, we do not know whether they successfully "stack" with others to improve robustness further still, or whether the use of one technique materially affects the performance of another.

Acknowledgements

Annotation and correction for wrongly spelled search queries was carried out by Pactera EDGE. We thank Elizabeth Milkovits and the anonymous ARR reviewers for their helpful feedback on earlier drafts of this paper.

References

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Toms Bergmanis, Artūrs Stafanovičs, and Mārcis Pinnis. 2020. Robust neural machine translation: Modeling orthographic and interpunctal variation. In *Human Language Technologies – The Baltic Perspective: Proceedings of the Ninth Annual Conference Baltic HLT 2020*, pages 3103–3114, Kaunas, Lithuania. IOS Press.

Nicola Bertoldi, Mauro Cettolo, and Marcello Federico. 2010. Statistical machine translation of texts with misspelled words. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 412–419, Los Angeles, California. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jyoti Guha and Carmen Heger. 2014. Machine translation for global e-commerce on eBay. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Users Track*, pages 31–37, Vancouver, Canada. Association for Machine Translation in the Americas.

Saša Hasan, Carmen Heger, and Saab Mansour. 2015. Spelling correction of user search queries through statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 451–460, Lisbon, Portugal. Association for Computational Linguistics.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. 2018. How robust are character-based word embeddings in tagging and MT against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 68–80, Boston, Massachusetts. Association for Machine Translation in the Americas.

Felix Hieber, Michael Denkowski, Tobias Domhan, Barbara Darques Barros, Celina Dong Ye, Xing Niu, Cuong Hoang, Ke Tran, Benjamin Hsu, Maria Nadejde, Surafel Lakew, Prashant Mathur, Anna Currey, and Marcello Federico. 2022. Sockeye 3: Fast neural machine translation with PyTorch. *Computing Research Repository*, arXiv:2207.05851. Version 1.

Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. Sockeye 2: A toolkit for neural machine translation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 457–458, Lisboa, Portugal. European Association for Machine Translation.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computing Research Repository*, arXiv:1412.6980. Version 9.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3103–3114, Minneapolis, Minnesota. Association for Computational Linguistics.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8538–8544, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Maja Popović. 2015. chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matt Post and Kevin Duh. 2019. JHU 2019 robustness task system description. In *Proceedings of the Fourth Conference on Machine Translation (Volume*

2: Shared Task Papers, Day 1), pages 552–558, Florence, Italy. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz Rojas. 2020. Bifixer and Bicleaner: Two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Shadi Saleh and Pavel Pecina. 2020. Document translation vs. query translation for cross-lingual information retrieval in the medical domain. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6849–6860, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts. Association for Machine Translation in the Americas.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Liang Yao, Baosong Yang, Haibo Zhang, Weihua Luo, and Boxing Chen. 2020. Exploiting neural query translation into cross lingual information retrieval. In *Proceedings of the SIGIR 2020 Workshop on eCommerce (ECOM '20)*, Virtual event.

A Spelling Correction Test Sets

As introduced in Section 3.1, we biased our spelling correction test sets towards search queries that an existing model flagged as misspelled. This oversampling is because misspelled queries are overall rather rare. However, the model's judgment was only used to help collect data and was never taken as ground truth — all annotations for the correct and incorrect spelling of queries in the test sets were provided by humans.

For example, we used the in-house model to select a batch of raw search queries for which the model believes 50% are correct and 50% contain a misspelling. Human judges annotated the whole batch, perhaps finding that 35% of the queries in fact have spelling mistakes — including some queries that the model judged as spelled correctly. The test set used for our experiments then consisted of only those human-identified and human-corrected queries from the 35%. In this way, we believe the biasing approach increases the size of the test set — i.e. 35% of the batch instead of the (say) 5% that might appear in a uniform sample — without losing any part of the distribution of true spelling errors.

Table 4 shows our complete list of test sets. These consist of the human-verified rightly and wrongly spelled query pairs in the relevant input language, as described above, for later use in an e-commerce experience where the queries will be automatically translated into the primary language of the product catalog. For example, one line of our test set for Canadian French (fr-CA), for use against a Canadian English (en-CA) catalog, consists of the query pair (*chaise bureau*, *chaise buraeu*).

B MT System Data Preparation

The slates of MT system builds described in Section 5 were based on the data resources released as part of each of the following WMT shared tasks:

- 2013 (news) translation task for Spanish–English (ES–EN)
- 2015 (news discussions) translation task for French–English (FR–EN)

Language	Primary	Queries
fr-CA	en-CA	5,270
cs-CZ	de-DE	2,853
en-GB		2,110
nl-NL		1,195
pl-PL		3,303
tr-TR		4,033
pt-PT	es-ES	2,841
bn-IN	en-IN	4,458
hi-IN		3,486
kn-IN		5,212
ml-IN		5,250
mr-IN		4,857
ta-IN		5,582
te-IN		6,242
en-US	ja-JP	3,647
zh-CN		875
en-GB	nl-NL	3,140
en-GB	sv-SE	3,650
de-DE	en-US	1,033
es-MX		5,076
he-IL		2,741
ko-KR		2,128
pt-BR		3,518
zh-CN		939
zh-TW		2,361

Table 4: Languages for which we created pairs of rightly and wrongly spelled search queries, showing the sizes of each test set after extraction, filtering, and human annotation. Note that these test sets are monolingual.

- 2020 news translation task for Tamil–English (TA–EN)
- 2023 general translation task for Hebrew–English (HE–EN)
- 2023 general translation task for English–German (EN–DE)
- 2023 general translation task for English–Japanese (EN–JA)

We selected the following WMT corpora for our initial MT training data. Most of the data sets were merely unpacked (or extracted) and combined together. We followed more substantial *cleaning* procedures, however, for three types of corpora. They are annotated with a * mark in the corpus lists and described in more detail below.

- **ES–EN:** Common Crawl*, Europarl v7, News Commentary v8, UN Docs
- **FR–EN:** Common Crawl*, Europarl v7, Giga-FrEn v2*, News Commentary v10, UN Docs
- **HE–EN:** Bible, CCAligned*, ELRC Wikipedia Health, GNOME, KDE4*, NeuLab TED Talks, NLLB, OpenSubtitles*, PHP*, QED, Tatoeba, TED 2020, WikiMatrix*, Wikimedia, Wikipedia, XLEnt
- **TA–EN:** CUNI Parallel Train v2, MKB v0, nlpcuom Corpus v1.0.3, nlpcuom Glossary v1.0.3, PIB v0, PMIndia v1, Tanzil, Wikimatrix v1*, Wikititles v2
- **EN–DE:** Common Crawl*, Europarl v10, News Commentary v18, Paracrawl*, Tilde Air Baltic, Tilde Czech Tourism, Tilde ECB 2017, Tilde EESC 2017, Tilde EMEA 2016, Tilde Rapid 2016, Wikimatrix v1*, Wikititles v3
- **EN–JA:** JESC Train, JParacrawl v3*, KFTT v1 Train, News Commentary v18, TED, Wikimatrix v1*, Wikititles v3

(1) Wikimatrix corpora were distributed by WMT already annotated with their language IDs and margin (parallelism) scores, as per Schwenk et al. (2021). We followed their Section 4.2 in filtering these corpora, keeping only those lines that were marked as being in the correct source and target languages and that had margin scores of at least 1.04.

(2) Common Crawl, Giga-FrEn, OpenSubtitles, and Paracrawl are large noisy corpora; the smaller

KDE4 and PHP corpora for HE–EN also appeared noisy upon manual inspection. We adopted a somewhat simpler approach for cleaning these data sets, consisting of running FastText language ID followed by computing cosine distances between the LASER segment embeddings of the source and target side of each line of data. We kept only those lines that were detected to be in the correct source and target languages and that had cosine similarities of at least 0.8. The FastText language ID model was `lid.176.bin`; the LASER embedding model was `bilstm.93langs.2018-12-26.pt`. The EN–DE Paracrawl corpus was so large that we only attempted to clean its first 93 million lines.

(3) The JParacrawl corpus was distributed by WMT already annotated with its Bicleaner scores, as per Ramírez-Sánchez et al. (2020). We followed their Section 3 in filtering this corpus, keeping only those lines that had scores of at least 0.7.

All selected corpora were concatenated together for each language pair. Their final combined line counts are listed in Table 5. As part of system training, the combined training data was *filtered* — after tokenization and BPE encoding in each system variant — to remove segment pairs consisting of too many tokens on either side, containing tokens with more than 100 characters, or where the length ratio between source and target was too unbalanced.

Development sets were sourced from WMT as well, typically from the same year that supplied the training data. We used newstest2012 for ES–EN, newsdiscussdev2015 for FR–EN, the concatenation of Flores-200 dev and devtest for HE–EN, newsdev2020 for TA–EN, reference A of wmttest2022 for EN–DE, and wmttest2022 for EN–JA.[4] Table 5 gives the line counts for these corpora.

The training corpora and development sets that we used during fine-tuning our MT systems to the search query domain were previously described in Section 5.1. Their line counts are included in Table 5 for reference.

Language	Corpus	Lines
ES–EN	Base train (cleaned)	14,563,500
	Base dev	3,003
	Fine-tune train	5,422,787
	Fine-tune dev	4,000
FR–EN	Base train (cleaned)	34,502,802
	Base dev	1,500
	Fine-tune train	5,282,996
	Fine-tune dev	4,000
HE–EN	Base train (cleaned)	52,529,536
	Base dev	3,009
	Fine-tune train	5,146,637
	Fine-tune dev	4,000
TA–EN	Base train (cleaned)	552,752
	Base dev	1,989
	Fine-tune train	5,120,423
	Fine-tune dev	4,000
EN–DE	Base train (cleaned)	82,720,693
	Base dev	2,420
	Fine-tune train	5,274,628
	Fine-tune dev	4,000
EN–JA	Base train (cleaned)	3,482,748
	Base dev	2,037
	Fine-tune train	5,171,246
	Fine-tune dev	4,000

Table 5: Line counts of our selected base training data, base dev sets, fine-tuning training data, and fine-tuning dev sets.

C MT System Training Hyperparameters

Details of the hyperparameters we set while training our MT systems are given below. (Key values configured directly as a result of our main BPE experiments were already described in Section 5.1.)

We set an initial learning rate of 0.0002, used

[4]The 2023 test set references for HE–EN, EN–DE, and EN–JA had not yet been released at the time we performed this work.

Adam (Kingma and Ba, 2014), and decayed the rate by a factor of 0.9 whenever training progressed 8 checkpoints without improving on the dev set.

Convergence during base training was defined as 60 checkpoints without improvement on the dev set, but after a minimum of one complete epoch and within a maximum of 832,500 batches. After convergence, the eight best checkpoints were averaged together. Convergence during fine-tuning, within the same min-epochs and max-batches limits as used during the base training, was defined as 10 checkpoints without improvement on the dev set. Otherwise, the maximum input lengths per BPE variant and the remaining hyperparameters are identical to the base.

The complete set of common parameters provided to Sockeye 3's `train.py` command for the base training of all systems is listed in full below.

```
average-checkpoints True
batch-size 2048
batch-type word
decode-and-evaluate 500
decoder transformer
embed-dropout 0.0:0.0
encoder transformer
gradient-clipping-threshold -1
gradient-clipping-type abs
initial-learning-rate 0.0002
keep-initializations True
keep-last-params 200
label-smoothing 0.1
learning-rate-reduce-factor 0.9
learning-rate-reduce-num-not-improved 8
learning-rate-scheduler-type plateau-reduce
learning-rate-warmup 0
length-task-layers 2
length-task-type None
length-task-weight 0.0
max-num-checkpoint-not-improved 60
max-updates 832500
metric bleu
min-num-epochs 1
min-samples 0
min-updates 0
n 8
num-embed 512:512
num-layers 20:2
optimized-metric bleu
optimizer adam
seed 1
strategy best
transformer-attention-heads 8:8
transformer-dropout-act 0.1:0.1
transformer-dropout-attention 0.1:0.1
transformer-dropout-prepost 0.1:0.1
transformer-feed-forward-num-hidden 2048:2048
transformer-model-size 512:512
transformer-positional-embedding-type fixed
transformer-postprocess dr:dr
transformer-preprocess n:n
weight-tying-type src_trg_softmax
```

D Spelling Correction System Training Hyperparameters

Details of the hyperparameters we set during the training of our spelling correction systems (Section 6.1) are given below. Hyperparameters were the same for both models.

We set an initial learning rate of 0.0001, used Adam (Kingma and Ba, 2014) with betas 0.9 and 0.98, and set a dropout of 0.3. We set the BPE vocab size to 32,000 and a maximum input length of 4,000 tokens (from which Fairseq calculates the batch size accordingly). Convergence was defined as 5 epochs without improvement on respective dev set, up to a maximum of 20 epochs. Only the best checkpoint after convergence was retained.

The complete set of common parameters we provided to Fairseq's `fairseq-train` command is listed in full below.

```
adam_betas (0.9,0.98)
arch bart_base
clip_norm 0.0
decoder_ffn_embed_dim 4096
decoder_layers 3
dropout 0.3
encoder_ffn_embed_dim 4096
encoder_layers 3
fp16 True
lr 0.0001
lr_scheduler inverse_sqrt
max_epoch 20
max_target_positions 128
max_tokens 4000
min_lr 1e-09
optimizer adam
patience 5
seed 1
share_all_embeddings True
update_freq 1
warmup_init_lr 1e-07
warmup_updates 4000
```