# Machine Translation of Restaurant Reviews: New Corpus for Domain Adaptation and Robustness

## Abstract

We share a French-English parallel corpus of Foursquare restaurant reviews (this https URL), and define a new task to encourage research on Neural Machine Translation robustness and domain adaptation, in a real-world scenario where better-quality MT would be greatly beneficial. We discuss the challenges of such user-generated content, and train good baseline models that build upon the latest techniques for MT robustness. We also perform an extensive evaluation (automatic and human) that shows significant improvements over existing online systems. Finally, we propose task-specific metrics based on sentiment analysis or translation accuracy of domain-specific polysemous words.

## Introduction

Very detailed information about social venues such as restaurants is available from user-generated reviews in applications like Google Maps, TripAdvisor or Foursquare. Most of these reviews are written in the local language and are not directly exploitable by foreign visitors: an analysis of the Foursquare database shows that, in Paris, only 49% of the restaurants have at least one review in English. It can be much worse for other cities and languages (e.g., only 1% of Seoul restaurants for a French-only speaker).

Machine Translation of such user-generated content can improve the situation and make the data available for direct display or for downstream NLP tasks (e.g., cross-lingual information retrieval, sentiment analysis, spam or fake review detection), provided its quality is sufficient.

We asked professionals to translate 11.5k French Foursquare reviews (18k sentences) to English. We believe that this resource will be valuable to the community for training and evaluating MT systems addressing challenges posed by user-generated content, which we discuss in detail in this paper.

We conduct extensive experiments and combine techniques that seek to solve these challenges (e.g., factored case, noise generation, domain adaptation with tags) on top of a strong Transformer baseline. In addition to BLEU evaluation and human evaluation, we use targeted metrics that measure how well polysemous words are translated, or how well sentiments expressed in the original review can still be recovered from its translation.

## Related work

Translating restaurant reviews written by casual customers presents several difficulties for NMT, in particular robustness to non-standard language and adaptation to a specific style or domain (see Section SECREF7 for details).

Concerning robustness to noisy user generated content, BIBREF0 stress differences with traditional domain adaptation problems, and propose a typology of errors, many of which we also detected in the Foursquare data. They also released a dataset (MTNT), whose sources were selected from a social media (Reddit) on the basis of being especially noisy (see Appendix for a comparison with Foursquare). These sources were then translated by humans to produce a parallel corpus that can be used to engineer more robust NMT systems and to evaluate them. This corpus was the basis of the WMT 2019 Robustness Task BIBREF1, in which BIBREF2 ranked first. We use the same set of robustness and domain adaptation techniques, which we study more in depth and apply to our review translation task.

BIBREF3, BIBREF4 and BIBREF5 propose to improve robustness by training models on data-augmented corpora, containing noisy sources obtained by random word or character deletions, insertions, substitutions or swaps. Recently, BIBREF6 proposed to use a similar technique along with noise generation through replacement of a clean source by one obtained by back-translation.

We employ several well-known domain adaptation techniques: back-translation of large monolingual corpora close to the domain BIBREF7, BIBREF8, fine-tuning with in-domain parallel data BIBREF9, BIBREF10, BIBREF11, domain tags for knowledge transfer between domains BIBREF12, BIBREF2.

Addressing the technical issues of robustness and adaptation of an NMT system is decisive for real-world deployment, but evaluation is also critical. This aspect is stressed by BIBREF13 (NMT of curated hotel descriptions), who point out that automatic metrics like BLEU tend to neglect semantic differences that have a small textual footprint, but may be seriously misleading in practice, for instance by interpreting available parking as if it meant free parking. To mitigate this, we conduct additional evaluations of our models: human evaluation, translation accuracy of polysemous words, and indirect evaluation with sentiment analysis.

## Task description

We present a new task of restaurant review translation, which combines domain adaptation and robustness challenges.

## Task description ::: Corpus description

We sampled 11.5k French reviews from Foursquare, mostly in the food category, split them into 18k sentences, and grouped them into train, valid and test sets (see Table TABREF6). The French reviews contain on average 1.5 sentences and 17.9 words. Then, we hired eight professional translators to translate them to English. Two of them created the training set by post-editing (PE) the outputs of baseline NMT systems. The other six translated the valid and test sets from scratch. They were asked to translate (or post-edit) the reviews sentence-by-sentence (to avoid any alignment problem), but they could see the full context. We manually filtered the test set to remove translations that were not satisfactory. The full reviews and additional metadata (e.g., location and type of the restaurant) are also available as part of this resource, to encourage research on contextual machine translation.

Foursquare-HT was translated from scratch by the same translators who post-edited Foursquare-PE. While we did not use it in this work, it can be used as extra training or development data. We also release a human translation of the French-language test set (668 sentences) of the Aspect-Based Sentiment Analysis task at SemEval 2016 BIBREF14.

## Task description ::: Challenges

Translating restaurant reviews presents two main difficulties compared to common tasks in MT. First, the reviews are written in a casual style, close to spoken language. Some liberty is taken w.r.t. spelling, grammar, and punctuation. Slang is also very frequent. MT should be robust to these variations. Second, they generally are reactions, by clients of a restaurant, about its food quality, service or atmosphere, with specific words relating to these aspects or sentiments. These require some degree of domain adaptation. The table above illustrates these issues, with outputs from an online MT system. Examples of full reviews from Foursquare-PE along with metadata are shown in Appendix.

Examples 1 and 2 fall into the robustness category: 1 is an extreme form of SMS-like, quasi-phonetic, language (et quand j'ai vu ça); 2 is a literal transcription of a long-vowel phonetic stress (trop $\rightarrow

$ trooop). Example 3 falls into the domain category: in a restaurant context, cadre typically refers to the setting. Examples 4 and 5 involve both robustness and domain adaptation: pété un cable is a non-compositional slang expression and garçon is not a boy in this domain; nickel is slang for great, très is missing an accent, and pâtes is misspelled as pattes, which is another French word.

Regarding robustness, we found many of the same errors listed by BIBREF0 as noise in social media text: SMS language (é qd g vu sa), typos and phonetic spelling (pattes), repeated letters (trooop, merciiii), slang (nickel, bof, mdr), missing or wrong accents (tres), emoticons (`:-)') and emojis, missing punctuation, wrong or non-standard capitalization (lowercase proper names, capitalized words for emphasis). Regarding domain aspects, there are polysemous words with typical specific meaning carte $\rightarrow $ map, menu; cadre $\rightarrow $ frame, executive, setting), idiomatic expressions (à tomber par terre $\rightarrow $ to die for), and venue-related named entities (La Boîte à Sardines).

## Robustness to noise

We propose solutions for dealing with non-standard case, emoticons, emojis and other issues.

### Robustness to noise ::: Rare character placeholder

We segment our training data into subwords with BPE BIBREF15, implemented in SentencePiece BIBREF16. BPE can deal with rare or unseen words by splitting them into more frequent subwords, but cannot deal with unseen characters. While this is not a problem in most tasks, Foursquare contains many emojis, and sometimes symbols in other scripts (e.g., Arabic). Unicode now defines around 3k emojis, most of which are likely to be out-of-vocabulary.

We replace rare characters on both sides of the training corpus by a placeholder (<x>). A model trained on this data is typically able to copy the placeholder at the correct position. Then, at inference time, we replace the output tokens <x> by the rare source-side characters, in the same order. This approach is similar to that of BIBREF18, who used the attention mechanism to replace UNK symbols with the aligned word in the source. BIBREF2 used the same technique to deal with emojis in the WMT robustness task.

## Robustness to noise ::: Capital letters

As shown in Table TABREF11, capital letters are another source of confusion. HONTE and honte are considered as two different words. The former is out-of-vocabulary and is split very aggressively by BPE. This causes the MT model to hallucinate.

### Robustness to noise ::: Capital letters ::: Lowercasing

A solution is to lowercase the input, both at training and at test time. However, when doing so, some information may be lost (e.g., named entities, acronyms, emphasis) which may result in lower translation quality.

### Robustness to noise ::: Capital letters ::: Factored translation

BIBREF13 do factored machine translation BIBREF19, BIBREF20 where a word and its case are split in two different features. For instance, HONTE becomes honte + upper.

We implement this with two embedding matrices, one for words and one for case, and represent a token as the sum of the embeddings of its factors. For the target side, we follow BIBREF20 and have two softmax operations. We first predict the word in its lowercase form and then predict its case. The embeddings of the case and word are then summed and used as input for the next decoder step.

## Robustness to noise ::: Capital letters ::: Inline casing

BIBREF2 propose another approach, inline casing, which does not require any change in the model. We insert the case as a regular token into the sequence right after the word. Special tokens <U>, <L> and <T> (upper, lower and title) are used for this purpose and appended to the vocabulary. Contrary to the previous solution, there is only one embedding matrix and one softmax.

In practice, words are assumed to be lowercase by default and the <L> tokens are dropped to keep the factored sequences as short as possible. “Best fries EVER" becomes “best <T> _f ries _ever <U>". Like BIBREF2, we force SentencePiece to split mixed-case words like MacDonalds into single-case subwords (Mac and Donalds).

## Robustness to noise ::: Capital letters ::: Synthetic case noise

Another solution that we experiment with (see Section SECREF6) is to inject noise on the source side of the training data by changing random source words to upper (5% chance), title (10%) or lower case (20%).

## Robustness to noise ::: Natural noise

One way to make an NMT system more robust is to train it with some of the most common errors that can be found in the in-domain data. Like BIBREF2, we detect the errors that occur naturally in the in-domain data and then apply them to our training corpus, while respecting their natural distribution. We call this “natural noise generation” in opposition to what is done in BIBREF3, BIBREF4, BIBREF6 or in Section SECREF10, where the noise is more synthetic.

## Robustness to noise ::: Natural noise ::: Detecting errors

We compile a general-purpose French lexicon as a transducer, implemented to be traversed with extended edit distance flags, similar to BIBREF21. Whenever a word is not found in the lexicon (which means that it is a potential spelling mistake), we look for a French word in the lexicon within a maximum edit distance of 2, with the following set of edit operations:

We apply the transducer to the French monolingual Foursquare data (close to 1M sentences) to detect and count noisy variants of known French words. This step produces a dictionary mapping the correct spelling to the list of observed errors and their respective frequencies.

In addition to automatically extracted spelling errors, we extract a set of common abbreviations from BIBREF22 and we manually identify a list of common errors in French:

## Robustness to noise ::: Natural noise ::: Generating errors

With this dictionary, describing the real error distribution in Foursquare text, we take our large out-of-domain training corpus, and randomly replace source-side words with one of their variants (rules 1 to 6), while respecting the frequency of this variant in the real data. We also manually define regular expressions to randomly apply rules 7 to 11 (e.g., "er "$\rightarrow $"é ").

We obtain a noisy parallel corpus (which we use instead of the “clean” training data), where about 30% of all source sentences have been modified, as shown below:

## Domain Adaptation

To adapt our models to the restaurant review domain we apply the following types of techniques: back-translation of in-domain English data, fine-tuning with small amounts of in-domain parallel data, and domain tags.

### Domain Adaptation ::: Back-translation

Back-translation (BT) is a popular technique for domain adaptation when large amounts of in-domain monolingual data are available BIBREF7, BIBREF8. While our in-domain parallel corpus is small (12k pairs), Foursquare contains millions of English-language reviews. Thus, we train an NMT model in the reverse direction (EN$\rightarrow $FR) and translate all the Foursquare English reviews to French. This gives a large synthetic parallel corpus.

This in-domain data is concatenated to the out-of-domain parallel data and used for training.

BIBREF8 show that doing back-translation with sampling instead of beam search brings large improvements due to increased diversity. Following this work, we test several settings:

We use a temperature of $T=\frac{1}{0.9}$ to avoid the extremely noisy output obtained with $T=1$ and strike a balance between quality and diversity.

## Domain Adaptation ::: Fine-tuning

When small amounts of in-domain parallel data are available, fine-tuning (FT) is often the preferred solution for domain adaptation BIBREF9, BIBREF10. It consists in training a model on out-of-domain data, and then continuing its training for a few epochs on the in-domain data only.

## Domain Adaptation ::: Corpus tags

BIBREF12 propose a technique for multi-domain NMT, which consists in inserting a token in each source sequence specifying its domain. The system can learn the particularities of multiple domains (e.g., polysemous words that have a different meaning depending on the domain), which we can control at test time by manually setting the tag. BIBREF23 also use tags to control politeness in the model's output.

As our corpus (see Section SECREF28) is not clearly divided into domains, we apply the same technique as BIBREF12 but use corpus tags (each sub-corpus has its own tag: TED, Paracrawl, etc.) which we add to each source sequence. Like in BIBREF2, the Foursquare post-edited and back-translated data also get their own tags (PE and BT). Figure FIGREF27 gives an example where using the PE corpus tag at test time helps the model pick a more adequate translation.

## Experiments ::: Training data

After some initial work with the WMT 2014 data, we built a new training corpus named UGC (User Generated Content), closer to our domain, by combining: Multi UN, OpenSubtitles, Wikipedia, Books, Tatoeba, TED talks, ParaCrawl and Gourmet (See Table TABREF31). UGC does not include Common Crawl (which contains many misaligned sentences and caused hallucinations), but it includes OpenSubtitles BIBREF24 (spoken-language, possibly closer to Foursquare). We observed an improvement of more than 1 BLEU on newstest2014 when switching to UGC, and almost 6 BLEU on Foursquare-valid.

## Experiments ::: Pre-processing

We use langid.py BIBREF25 to filter sentence pairs from UGC. We also remove duplicate sentence pairs, and lines longer than 175 words or with a length ratio greater than $1.5$ (see Table TABREF31). Then we apply SentencePiece and our rare character handling strategy (Section SECREF8). We use a joined BPE model of size 32k, trained on the concatenation of both sides of the corpus, and set SentencePiece's vocabulary threshold to 100. Finally, unless stated otherwise, we always use the inline casing approach (see Section SECREF10).

## Experiments ::: Model and settings

For all experiments, we use the Transformer Big BIBREF26 as implemented in Fairseq, with the hyperparameters of BIBREF27. Training is done on 8 GPUs, with accumulated gradients over 10 batches BIBREF27, and a max batch size of 3500 tokens per GPU. We train for 20 epochs, while saving a checkpoint every 2500 updates ($\approx \frac{2}{5}$ epoch on UGC) and average the 5 best checkpoints according to their perplexity on a validation set (a held-out subset of UGC).

For fine-tuning, we use a fixed learning rate, and a total batch size of 3500 tokens (training on a single GPU without delayed updates). To avoid overfitting on Foursquare-PE, we do early stopping according to perplexity on Foursquare-valid. For each fine-tuned model we test all 16 combinations of dropout in

$\lbrace 0.1,0.2,0.3,0.4\rbrace $ and learning rate in $\lbrace 1, 2, 5, 10\rbrace \times 10^{-5}$. We keep the model with the best perplexity on Foursquare-valid.

## Experiments ::: Evaluation methodology

During our work, we used BLEU BIBREF28 on newstest[2012, 2013] to ensure that our models stayed good on a more general domain, and on Foursquare-valid to measure performance on the Foursquare domain.

For sake of brevity, we only give the final BLEU scores on newstest2014 and Foursquare-test. Scores on Foursquare-valid, and MTNT-test (for comparison with BIBREF0, BIBREF2) are given in Appendix. We evaluate “detokenized” MT outputs against raw references using SacreBLEU BIBREF29.

In addition to BLEU, we do an indirect evaluation on an Aspect-Based Sentiment Analysis (ABSA) task, a human evaluation, and a task-related evaluation based on polysemous words.

## Experiments ::: BLEU evaluation ::: Capital letters

Table TABREF41 compares the case handling techniques presented in Section SECREF10. To better evaluate the robustness of our models to changes of case, we built 3 synthetic test sets from Foursquare-test, with the same target, but all source words in upper, lower or title case.

Inline and factored case perform equally well, significantly better than the default (cased) model, especially on all-uppercase inputs. Lowercasing the source is a good option, but gives a slightly lower score on regular Foursquare-test. Finally, synthetic case noise added to the source gives surprisingly good results. It could also be combined with factored or inline case.

## Experiments ::: BLEU evaluation ::: Natural noise

Table TABREF44 compares the baseline “inline case” model with the same model augmented with natural noise (Section SECREF17). Performance is the same on Foursquare-test, but significantly better on newstest2014 artificially augmented with Foursquare-like noise.

## Experiments ::: BLEU evaluation ::: Domain adaptation

Table TABREF46 shows the results of the back-translation (BT) techniques. Surprisingly, BT with beam search (BT-B) deteriorates BLEU scores on Foursquare-test, while BT with sampling gives a consistent improvement. BLEU scores on newstest2014 are not significantly impacted, suggesting that BT can be used for domain adaptation without hurting quality on other domains.

Table TABREF47 compares the domain adaptation techniques presented in Section SECREF5. We observe that:

Concatenating the small Foursquare-PE corpus to the 50M general domain corpus does not help much, unless using corpus tags.

Foursquare-PE + tags is not as good as fine-tuning with Foursquare-PE. However, fine-tuned models get slightly worse results on news.

Back-translation combined with tags gives a large boost. The BT tag should not be used at test time, as it degrades results.

Using no tag at test time works fine, even though all training sentences had tags.

As shown in Table TABREF54, these techniques can be combined to achieve the best results. The natural noise does not have a significant effect on BLEU scores. Back-translation combined with fine-tuning gives the best performance on Foursquare (+4.5 BLEU vs UGC). However, using tags instead of fine-tuning strikes a better balance between general domain and in-domain performance.

## Experiments ::: Targeted evaluation

In this section we propose two metrics that target specific aspects of translation adequacy: translation accuracy of domain-specific polysemous words and Aspect-Based Sentiment Analysis performance on MT outputs.

### Experiments ::: Targeted evaluation ::: Translation of polysemous words

We propose to count polysemous words specific to our domain, similarly to BIBREF31, to measure the degree of domain adaptation. TER between the translation hypotheses and the post-edited references in Foursquare-PE reveals the most common substitutions (e.g., “card” is often replaced with “menu”, suggesting that “card” is a common mistranslation of the polysemous word “carte”). We filter this list manually to only keep words that are polysemous and that have a high frequency in the test set. Table TABREF58 gives the 3 most frequent ones.

Table TABREF59 shows the accuracy of our models when translating these words. We see that the domain-adapted model is better at translating domain-specific polysemous words.

### Experiments ::: Targeted evaluation ::: Indirect evaluation with sentiment analysis

We also measure adequacy by how well the translation preserves the polarity of the sentence regarding various aspects. To evaluate this, we perform an indirect evaluation on the SemEval 2016 Aspect-Based Sentiment Analysis (ABSA) task BIBREF14. We use our internal ABSA systems trained on English or French SemEval 2016 data. The evaluation is done on the SemEval 2016 French test set: either the original version (ABSA French), or its translation (ABSA English). As shown in Table TABREF61, translations obtained with domain-adapted models lead to significantly better scores on ABSA than the generic models.

Experiments ::: Human Evaluation

We conduct a human evaluation to confirm the observations with BLEU and to overcome some of the limitations of this metric.

We select 4 MT models for evaluation (see Table TABREF63) and show their 4 outputs at once, sentence-by-sentence, to human judges, who are asked to rank them given the French source sentence in context (with the full review). For each pair of models, we count the number of wins, ties and losses, and apply the Wilcoxon signed-rank test.

We took the first 300 test sentences to create 6 tasks of 50 sentences each. Then we asked bilingual colleagues to rank the output of 4 models by their translation quality. They were asked to do one or more of these tasks. The judge did not know about the list of models, nor the model that produced any given translation. We got 12 answers. The inter-judge Kappa coefficient ranged from 0.29 to 0.63, with an average of 0.47, which is a good value given the difficulty of the task. Table TABREF63 gives the results of the evaluation, which confirm our observations with BLEU.

We also did a larger-scale monolingual evaluation using Amazon Mechanical Turk (see Appendix), which lead to similar conclusions.

## Conclusion

We presented a new parallel corpus of user reviews of restaurants, which we think will be valuable to the community. We proposed combinations of multiple techniques for robustness and domain adaptation, which address particular challenges of this new task. We also performed an extensive evaluation to measure the improvements brought by these techniques.

According to BLEU, the best single technique for domain adaptation is fine-tuning. Corpus tags also achieve good results, without degrading performance on a general domain. Back-translation helps, but only with sampling or tags. The robustness techniques (natural noise, factored case, rare character placeholder) do not improve BLEU.

While our models are promising, they still show serious errors when applied to user-generated content: missing negations, hallucinations, unrecognized named entities, insensitivity to context. This suggests that this task is far from solved.

We hope that this corpus, our natural noise dictionary, model outputs and human rankings will help better understand and address these problems. We also plan to investigate these problems on lower resource languages, where we expect the task to be even harder.