Reasonable Multilingual Sentiment Analysis Using a Simple Machine Translation Approach

Anonymous ACL submission

Abstract

This project began as an ambitious analysis of cross-language sentiment analysis. As reality hit, however, we discovered important lessons about machine learning, especially the importance of not re-inventing (or re-training) the wheel and, instead, trying to find general solutions to specific problems. This led us to investigate Machine Translation as a more reasonable approach to multilingual sentiment analysis. In the end, we found that, Machine Translation cannot currently be used as a substitute for language-specific training, but that it can, in some cases, be a substitute for language-specific preprocessing.

1 Introduction

Machine Translation (MT) quality has improved significantly in recent year (Hirschberg and Manning, 2015; Lample et al., 2018). This project proposes to evaluate an approach which makes use of these improvements for multilingual sentiment analysis. More specifically, we propose to use MT to translate non-English datasets to English and then apply a sentiment analysis algorithm on the resulting translated dataset. We became interested in this approach as a result of difficulties we encountered working on this project. In what follows, we discuss these difficulties, what they taught us, how they led us to focus on this approach.

This project began with ambition: we were going to collect hundreds of thousands of online product reviews from various branches of the multinational Amazon, we were going to use those reviews with a state-of-the-art sentiment analysis neural network, and we were going to evaluate the different ways in which these multilingual data could be used to predict sentiment across languages. Reality, however, quickly caught up.

We met our first obstacle during data collection. Though we were able to collect over $662,000$ reviews, 40% were in English from amazon.com, 27% in French from amazon.fr, 27% in Spanish from amazon.es, and the remaining 6% in Chinese from amazon.cn. Importantly, it was far easier to obtain data in English than in any other language, and significantly harder in Chinese than in the other languages.[1] In fact, we had originally wanted to work with Modern Standard Arabic and amazon.ae, but that branch of Amazon is newer and had very few reviews in Arabic; far too few for our purposes.

The second problem arose when we began working with non-English data, deciding on the preprocessing steps to apply to each language. On the one hand, it was unclear, for example, whether accents in French and Spanish should be removed or whether lemmatization was appropriate for those languages. On the other, the considerations for Chinese were more fundamental: contrarily to English, French, and Spanish, Chinese does not feature natural word boundaries (i.e., spaces between words); yet proper segmentation is necessary and the accuracy of this segmentation can impact the results of subsequent analyses (Chen et al., 2018). How does one, then, do proper segmentation of Chinese text?

We naturally turned to the scientific literature on sentiment analysis to look for answers to these questions, and it was there that we ran into our third and final problem: the lack of literature. Not only did we only find very few papers concerned with this topic and those languages, many of the papers we found came to a similar conclu-

[1] This is despite limitations on the number of reviews per product and on the number of product pages crawled per category; limitations put in place to constrain this very imbalance. Without these safeguards, the imbalance would have been far greater.

sion: the vast majority of work done in sentiment analysis has focused on English with very little work in other languages (Aydoğan and Akcayol, 2016; Balahur and Turchi, 2012; Ghorbel and Jacot, 2011; Korayem et al., 2016; Pang et al., 2008) and, Chen et al. (2018) argue, the results for English may not readily apply to other languages.

These three issues—difficulties collecting non-English data, language-specific text processing requirements, and the dearth of literature on non-English sentiment analysis—led us to question the very nature of our project: Does it even make sense to use a language-specific approach where each language has its own dataset, preprocessing steps, and trained machine learning algorithms? Already, research in Natural Language Processing (NLP) has tended to separate related but different tasks (e.g., sentiment analysis and opinion mining), each with its own preferred algorithm, trained neural networks, etc., the consequence of which has sometimes been that the solutions provided (e.g., the trained neural networks) do not generalize to related tasks. Worse yet, the solutions provided have often been for specific to English-language tasks and datasets, leaving future research to come up with new "solutions" (i.e., preferred algorithms, trained neural networks, etc.) for other languages. This approach—producing similar, but adapted resources and research for each new language—seems naive: there are too many languages and dialects for us to have a separate solution for each combination of language or dialect and task. An ideal approach would instead abstract away language to create language-less, concept-based datasets and algorithms. To our knowledge, however, such general language abstraction does not yet exist. Accordingly, a reasonable alternative is to use MT to translate from non-English languages to English and to apply the results of NLP research in English to translated-to-English datasets.

We thus ultimately decided to take on these new questions: Can MT be used in lieu of language-specific preprocessing? That is to say, can a model trained on a translated-to-English dataset identify sentiment in translated-to-English text? Moreover, can MT be used as a complete substitute such that a model trained on an English dataset can identify sentiment in translated-to-English text? Note that we are not proposing to train a MT model, but instead to use an already, freely available MT tool: Google Translate (Wu et al., 2016). The purpose, as argued previously, is to avoid reinventing the wheel (or unnecessarily retraining known algorithms) and, instead, to use already available mechanisms; a general solution to a specific problem.

2 Related works

Though most of the work in sentiment analysis has been done on datasets in English (Aydoğan and Akcayol, 2016; Balahur and Turchi, 2012; Ghorbel and Jacot, 2011; Korayem et al., 2016; Pang et al., 2008), there are instances of research done on other languages. Ghorbel and Jacot (2011), for example, report experiments on sentiment analysis of French movie reviews; Martín-Valdivia et al. (2013), on sentiment analysis of Spanish movie reviews; and Chen et al. (2018), on sentiment analysis of product reviews in Chinese from amazon.cn (see Aydoğan and Akcayol (2016) for a detailed survey of machine learning in sentiment analysis prior to 2016 and Korayem et al. (2016) for a survey on sentiment analysis in non-English languages prior to 2016). Our research complements this work: whereas their goal was to identify models appropriate for a specific language (English or otherwise), our goal here is to evaluate whether MT can help generalize their findings to other languages.

Our approach, using MT to translate a dataset to a target language before applying sentiment analysis, is not new. It was first applied by Bautin et al. (2008) and has since been applied by many others (including Balahur and Turchi, 2012; Brooke et al., 2009; Martín-Valdivia et al., 2013). Our work supplements the cited work by considering online product reviews from various branches of the multinational Amazon.

Finally, there has been previous work has investigating sentiment analysis on Amazon reviews (namely, Chen et al., 2018; Glorot et al., 2011; Rain, 2013), however, these studies focused on reviews in a single language (Chinese or English) whereas our work investigates four languages: English, French, Spanish, and Chinese.

3 Dataset and Experiments

3.1 Data collection

We collected our data using a Google Chrome extension written by one of the authors, Xavier. This extension consists in three parts. The first activates while browsing an Amazon website, when landing on a page listing products (e.g., `https://www.amazon.com/s?rh=n%3A16225007011%2Cn%3A1292110011`). This first part identifies all of the products with reviews on the page, chooses one at random which has not yet been visited, marks that product as visited, and instructs the browser to move to that product's page. The second part activates when the browser reaches a product page (e.g., `https://www.amazon.com/dp/B07MW159XC/`). It confirms that the product does have reviews and then instructs the browser to move on to the first page of reviews (e.g., `https://www.amazon.com/product-reviews/B07MW159XC/`). Once the browser has loaded a review page, the third and final part of the extension downloads that page of reviews and instructs the browser to move on to the next page of reviews. This process ends when there are no more reviews or when 10 pages of reviews have been downloaded; the browser is then instructed to return to the product listing page. The first part of the extension then activates anew, locates a new product to visit and instructs the browser to move to that product's page. When there are no products left unvisited, the browser is instructed to move on to the next product listing page. The extension terminates fully when there are no product listing pages left or when the tenth page is reached. Finally, the reviews were extracted from the review pages' HTML and written to a CSV file using a custom script written in Python (Van Rossum and Drake, 2009).

We chose to use a Chrome extension to collect our data rather than using a scraper for two reasons: CAPTCHAs and trustworthiness. First, we originally did try to use scrapers, but struggled to collect more than a handful of reviews: the scraper would quickly encounter a CAPTCHA which it had no way to solve. We were using a VPN and, when encountering a problem of any kind, the script would try again with a different VPN server and IP address, but we were nonetheless unable to collect much data. Compared to the scraper, when a CAPTCHA appears, the Chrome extension simply pauses and waits until a human solves the CAPTCHA before resuming scraping. Second, we do not know which methods companies like Amazon use to identify scrapers, but we hypothesized that there must be mechanisms meant to differentiate browsers from scrapers. In other words, we wanted to use the Chrome's inherent *trustworthiness* for scraping. This ultimately paid off: we ran into very few issues, for example having to solve only between 20 and 30 CAPTCHAs in the process of collecting over $662,000$ reviews.

rating	.com	.fr	.es	.cn
5	172,700	113,508	116,732	27,519
4	34,247	31,176	29,424	6,030
3	18,823	12,872	11,283	3,032
2	13,085	7,502	6,251	1,262
1	27,114	12,843	13,876	3,202
total	265,969	177,901	177,566	41,045

Table 1: Number of reviews per branch and rating.

3.2 Multilingual dataset

Table 1 presents the breakdown in terms of the number of reviews collected from each branch and for each rating. These reviews were collected from 28 product categories, the same or similar product categories across branches (e.g., infant clothing on amazon.cn, clothing for babies on amazon.es, 0-24 month clothing on amazon.fr, and both baby boy and baby girl clothing on amazon.com).

From this larger dataset, we created four smaller sets, each containing $22,000$ reviews. These smaller sets made training neural networks, translation, and running experiments more amenable than using the full datasets.

It should be noted that visual inspection of a sample of the reviews from each branch confirms that the vast majority of reviews are in the expected language; that is to say that almost all reviews from amazon.com were in English; amazon.fr, in French; amazon.es, in Spanish; and amazon.cn, in Chinese. We did not remove reviews which were not in the expected language from the datasets, but instead simply consider them a negligible source of noise.

3.3 Translated dataset

Three translated-to-English sets—French-to-English, Spanish-to-English, and Chinese-to-English—were created from their respective smaller sets. For each of these, all $22,000$ reviews were translated to English using the `GOOGLETRANSLATE()` function (Wu et al., 2016) provided through Google Sheets.

3.4 Data preparation

For the multilingual dataset, reviews in English, French, and Spanish were tokenized by splitting on white spaces, text was lowercased, and all tokens containing non-alphanumeric characters were removed (including punctuation, as well as unicode characters such as emojis). Chinese text was tokenized using the PyNLPIR (Roten, 2019), a Python wrapper library for NLPIR (Zhou and Zhang, 2003). Again, tokens with non-alphanumeric characters were removed.[2]

The translated-to-English reviews were treated as English; tokenized, lowercased, and stripped of non-alphanumeric tokens.

3.5 Classifier

Our classifier was implemented in Python (Van Rossum and Drake, 2009) using the NumPy (Oliphant, 2015) and PyTorch Python libraries (Paszke et al., 2017). It ran on Google's Colaboratory (Bisong, 2019).

We chose to use the Kim (2014) Convolutional Neural Network for sentiment analysis. This type of neural network architecture is very popular in the machine learning literature (with close to 5900 citations on Google Scholar at the time of this writing) and has previously been applied to sentiment analysis (e.g., Severyn and Moschitti, 2015; Cai and Xia, 2015).

This classifier is composed of a word embedding layer followed by several parallel convolutional units, a dropout layer and, finally, a linear classifier. Each convolutional unit is itself composed of a convolutional layer, a batch normalization layer, and a rectified linear unit. The output of these parallel units is concatenated before being passed on to the final dropout and linear classifier layers.

Our model's embedding layer was trained separately using our full datasets for each language[3] and using the Python library GenSim's (Řehůřek and Sojka) Word2Vec Continuous Bag Of Words (CBOW) implementation. Our model had three convolutional units, with filters of size three, four, and five respectively.

3.6 Experiments

We were interested in comparing the performances of models trained and tested in their original language to that of models trained and tested on translated-to-English datasets and to that of a model trained on an English language dataset and tested on translated-to-English datasets.

In order to do so, each dataset was split into three subsets: a training set of $20,000$ reviews, a validation set of $1,000$ reviews, and a test set of $1,000$ reviews. Using these subsets, seven networks were trained and tested, one for each language or translated dataset: English, French, Spanish, Chinese, French-to-English, Spanish-to-English, and Chinese-to-English. The performance of the English-trained network was also evaluated on the translated-to-English test sets.

Finally, in order to circumvent issues arising from the imbalance in ratings in our dataset (see Table 1 for more details), we created new datasets where the ratings were binarized such that a rating of "5" was considered positive and any other rating, negative, and trained seven new models. We report the results of both experiments on both binary and non-binary datasets.

4 Results and Discussion

The results of the first experiment are presented in Table 2. Strikingly, the French and Spanish translated-to-English trained models performed better than the models trained on the original language data, whereas we observe the opposite effect in the case of Chinese. This effect for French and Spanish was unexpected: we expected that the models might perform similarly, but, instead, for those languages, the translated-to-English-trained models performed much better than their counterparts. There are a few possible explanations. First, the translated-to-English models used word embedding trained on the full English dataset (see footnote 3) which was 1.5 times bigger than those for either French or Spanish. Having more examples may have allowed the models to develop a better sentiment representation. Another explanation is that translation may have constrained vocabulary in a way which helped the sentiment analysis models. For example, let us imagine two 5-star French reviews: "c'est génial" (it's great) and "c'est merveilleux" (it's wonderful). Let us assume that the MT algorithm translated both to "it's great." The French-

[2] The Python function `isalnum()` treats Chinese characters as "alpha". Visual inspection confirmed that this removed punctuation and emojis, but did not affect the text.

[3] The full English dataset was used to train the embedding layers for the translated-to-English datasets.

rating	English	French		Spanish		Chinese	
		original	*translated*	*original*	*translated*	*original*	*translated*
5	69.83	67.53	85.62	66.67	89.70	56.41	52.17
4	12.90	26.47	76.32	29.17	77.22	NaN	NaN
3	34.62	29.31	65.29	27.37	53.01	30.00	12.24
2	34.83	18.92	56.51	37.37	49.95	19.80	17.02
1	88.51	85.56	90.36	87.53	90.23	87.11	84.97
positive	79.26	73.08	86.03	71.04	80.87	72.30	65.07
negative	89.99	86.59	92.54	87.77	91.88	88.35	85.59

Table 2: F1 scores for the first experiment. Neural networks trained and tested on the original, multilingual datasets are compared to neural networks trained and tested on the translated-to-English datasets.

rating	French	Spanish	Chinese
5	57.89	61.63	40.68
4	14.81	8.89	NaN
3	11.43	23.08	18.54
2	25.73	22.94	24.43
1	85.11	85.56	82.34
positive	68.31	66.44	63.61
negative	84.74	85.96	77.51

Table 3: F1 scores for the second experiment. The neural network trained on the English dataset was tested on the translated-to-English test sets.

trained model would split the weight update between “génial” and “merveilleux,” whereas the translated-to-English model would have concentrated the update all on “great.” This is corroborated by the fact that the translated-to-English models in fact performed better than the English model did. As for the Chinese model, the lower performances may be a statement on the quality of MT from English to Chinese; unsurprisingly, translation between between Roman languages—French and Spanish—and Germanic languages—English—, that is, between Indo-European languages, is easier than between Indo-European and Sino-Tibetan languages.

The results of the second experiment are presented in Table 3. We note that, save for a few exceptions, the scores are lower than the corresponding scores reported for first experiment (see Table 2). That is to say that the scores for the English-trained model on the translated-to-English test sets are worse than those for the models trained on their own languages or trained on translated-to-English data.

The results of the first experiment suggest that MT is a reasonable and, perhaps even, desirable alternative to language-specific data preprocessing for sentiment analysis. For example, in this project, in preprocessing data from the original languages, we chose to keep accents and did not apply any kind of stemming or lemmatization. We made this choice due to the lack of sufficient literature to guide this decision. What our results suggest is that, in the future, one could simply avoid this question by translating the data to English, preprocessing the data in a way demonstrated to work for English and the task at hand, and training the model using this translated-to-English preprocessed data. This approach, at least for the time being, however, appears to work best with languages in the same family as the target language (English, in this case) and, otherwise, for language pairs for which MT is expected to produce quality translations.

The second experiment suggests, however, that MT does yet permit us to generalize sentiment analysis findings in English to other languages. That is to say that, at least for the time being, we will need to continue investigating language-specific solutions.

5 Conclusion

Machine Translation has come a long way in the last decade. As it keeps progressing, we believe that machine translation may enable researchers to generalize the results of sentiment analysis work in English to other languages. More generally, we hope that this approach will make it possible to apply the results of Natural Language Processing in English to other languages, simply by translating to English.

References

Ebru Aydoğan and M Ali Akcayol. 2016. A comprehensive survey for sentiment analysis tasks using machine learning techniques. In 2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), pages 1–7. IEEE.

Alexandra Balahur and Marco Turchi. 2012. Multilingual sentiment analysis using machine translation? In Proceedings of the 3rd workshop in computational approaches to subjectivity and sentiment analysis, pages 52–60. Association for Computational Linguistics.

Mikhail Bautin, Lohit Vijayarenu, and Steven Skiena. 2008. International sentiment analysis for news and blogs. In ICWSM.

Ekaba Bisong. 2019. Google colaboratory. In Building Machine Learning and Deep Learning Models on Google Cloud Platform, pages 59–64. Springer.

Julian Brooke, Milan Tofiloski, and Maite Taboada. 2009. Cross-linguistic sentiment analysis: From english to spanish. In Proceedings of the international conference RANLP-2009, pages 50–54.

Guoyong Cai and Binbin Xia. 2015. Convolutional neural networks for multimedia sentiment analysis. In Natural Language Processing and Chinese Computing, pages 159–167. Springer.

Huiling Chen, Shi Li, Peihuang Wu, Nian Yi, Shuyun Li, and Xiaoran Huang. 2018. Fine-grained sentiment analysis of chinese reviews using lstm network. Journal of Engineering Science & Technology Review, 11(1).

Hatem Ghorbel and David Jacot. 2011. Sentiment analysis of french movie reviews. In Advances in Distributed Agent-Based Retrieval Tools, pages 97–108. Springer.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 513–520.

Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. Science, 349(6245):261–266.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Mohammed Korayem, Khalifeh Aljadda, and David Crandall. 2016. Sentiment/subjectivity analysis survey for languages other than english. Social network analysis and mining, 6(1):75.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. arXiv preprint arXiv:1804.07755.

María-Teresa Martín-Valdivia, Eugenio Martínez-Cámara, Jose-M Perea-Ortega, and L Alfonso Ureña-López. 2013. Sentiment polarity detection in spanish reviews combining supervised and unsupervised approaches. Expert Systems with Applications, 40(10):3934–3942.

Travis E. Oliphant. 2015. Guide to NumPy, 2nd edition. CreateSpace Independent Publishing Platform, USA.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2):1–135.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In NIPS-W.

Callen Rain. 2013. Sentiment analysis in amazon reviews using probabilistic machine learning. Swarthmore College.

Radim Řehůřek and Petr Sojka. GenSim: Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.

Thomas Roten. 2019. PyNLPIR 0.6.0.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 959–962. ACM.

Guido Van Rossum and Fred L. Drake. 2009. Python 3 Reference Manual. CreateSpace, Paramount, CA.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Lina Zhou and Dongsong Zhang. 2003. NLPIR: A theoretical framework for applying natural language processing to information retrieval. Journal of the American Society for Information Science and Technology, 54(2):115–123.