Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5631–5640
Marseille, 11–16 May 2020

The Learnability of the Annotated Input in NMT
Replicating (Vanmassenhove and Way, 2018) with OpenNMT

Nicolas Ballier[1], **Nabil Amari**[2], **Laure Merat**[3], **Jean-Baptiste Yunès**[4]
Université de Paris[1,2,3,4]
CLILLAC-ARP, F-75013 Paris, France[1], UFR d'informatique[2,3] & IRIF[4], F-75013 Paris, France
{nicolas.ballier, jean-baptiste.yunes}@u-paris.fr, {nabil.amari,laure.merat}@etu.univ-paris-diderot.fr

Abstract

In this paper, we reproduce some of the experiments related to neural network training for Machine Translation as reported in (Vanmassenhove and Way, 2018). They annotated a sample from the EN-FR and EN-DE Europarl aligned corpora with syntactic and semantic annotations to train neural networks with the Nematus Neural Machine Translation (NMT) toolkit. Following the original publication, we obtained lower BLEU scores than the authors of the original paper, but on a more limited set of annotations. In the second half of the paper, we try to analyze the difference in the results obtained and suggest some methods to improve the results. We discuss the Byte Pair Encoding (BPE) used in the pre-processing phase and suggest feature ablation in relation to the granularity of syntactic and semantic annotations. The learnability of the annotated input is discussed in relation to existing resources for the target languages. We also discuss the feature representation likely to have been adopted for combining features.

Keywords: NMT, replication study, learnability, corpus annotation

1. Introduction

To validate a scientific experiment, its reproducibility is one of the bases of the scientific process because it may be proven wrong (Popper, 2005). As is well-known, if a repeated experiment gives different results, its validity is questioned as it fails to be generalised. The preoccupation for reproducibility when using Neural Networks (NN) is a frequent issue raised in their different applications. For example, (Laje and Buonomano, 2013) have analysed robustness against noise in Recurrent Neural Networks (RNN) used to investigate the stability of complex spatiotemporal motor patterns.

For neural networks used in Neural Machine Translation, reproducibility seems even more problematic as NN are fed with variable samples of the training data. To this end, the translation toolkit OpenNMT (Klein et al., 2017) in its PyTorch implementation, now has access to pseudorandom number generators, which offer better control on experiments, even though "completely reproducible results are not guaranteed across PyTorch releases".[1] We have used the version OpenNMT-py v1.0.0.rc1 (released on 1st of Oct 2019) of OpenNMT (PyTorch 1.2) to try to reproduce the experiments described in (Vanmassenhove and Way, 2018) as part of an on-going research project on the possibility to improve quality translation in NMT with linguistically annotated input in the training phase.

The remainder of the paper is organised as follows[2]: section 2 presents the method of the original paper (Vanmassenhove and Way, 2018). Section 3 delineates our replication process. Section 4 details how our experiments may further the original analysis. Section 5 explains our directions for future research.

2. Characterisation of the Original Approach

Vanmassenhove and Way (2018) incorporated semantic supersense tags and syntactic supertag features to their training datasets. By incorporating these features (particularly when combined) they found out that not only the model training converged faster but the features improved the model quality according to the BLEU scores (Papineni et al., 2002).

2.1. The Nematus Implementation Followed in the Original Paper

We found that there were no computational details or processing time reported for the Nematus Toolkit (Sennrich et al., 2017) training phase, but the required parameters for the training were provided: "[o]ur model was trained with the following parameters: vocabulary size: 45000, maximum sentence length: 60, vector dimension: 1024, word dimension: 500, learning optimizer: adadelta" (Vanmassenhove and Way, 2018). The number of operations for the byte-pair encoding, BPE (Sennrich et al., 2015) was made explicit (89,500 operations), even though we had to assume it applied to both source and target texts. We retained the same number of operations, even though previous papers using OpenNMT for English and French report 30,000 operations (Servan et al., 2017). The BPE documentation also suggests to optimise the process using two corpora at the same time, but this is not discussed in the original paper.

2.2. Data Sources

The original paper used a well-established dataset as the training set, so that we can assume that we retrieved the same Europarl dataset (Koehn, 2005). The validation sets and test sets proved more challenging: the repository for

[1] `https://pytorch.org/docs/stable/notes/randomness.html`

[2] We have been inspired by the previous editions of the workshops on replicability, especially (Branco et al., 2018). We found (Repar et al., 2018) outline particularly convincing and this outline closely follows theirs.

MT competitions[3] provides the archives of the test corpora used for Machine Translation competitions using Europarl. The test sets were thus defined in the original paper: "[w]e test the systems on 5K sentences (different from the training data) extracted from Europarl and the newstest2013"(Vanmassenhove and Way, 2018). This last sentence is potentially ambiguous: is it 5K in total for the two test sets? Since two figures were produced corresponding to two test sets, we decided to use two test sets of 5K sentences each. Several tests sets are available, but of 2,000 sentences, when the original paper seems to report a 5K test corpus. Similarly, the 2005 first competition using Europarl used 2K for validation tests and test sets. [4] Since the original paper did not report any subtraction from the Europarl (2005) data, we used the full 2005 Europarl corpus for the training and used the datasets provided in competitions from 2005 to 2007 to reach the 5K sentences of the validation and test sets. For the 5K test set, we concatenated the 2K sentences from the 2005 test set, [5] the 2K sentences from the 2006 test set [6] and the 1,000 first sentences from test set 2007.[7] For validation we compiled the 2K sentences from the 2005 Development Test Data [8] and from the 2006 competition, [9] and the 1,000 last sentences from the 2007 test set. [10]The second test set, referred to as "news2013", [11] seemed easier to identify. We assumed news2013 was used in classical papers for Phrase-Based Statistical Machine Translation (Wang et al., 2016) and was necessarily a multilingual aligned dataset. We used newsread in the STM competitions around 2013 and 2014. [12] It remains to be seen whether the test set used in the 2014 competition was the one used in the original paper. [13]

[3] http://matrix.statmt.org/test_sets/list

[4] http://www.statmt.org/wpt05/mt-shared-task/#TEST

[5] http://www.statmt.org/wpt05/mt-shared-task/realtest2000.en.gz

[6] http://matrix.statmt.org/test_sets/test2006.tgz?1504722372

[7] http://matrix.statmt.org/test_sets/test2007.tgz?1504722372

[8] http://www.statmt.org/wpt05/mt-shared-task/test2000.en.gz

[9] http://www.statmt.org/wmt06/shared-task/ namely the files http://www.statmt.org/wmt06/shared-task/dev2006.en.gz, http://www.statmt.org/wmt06/shared-task/dev2006.fr.gz, and http://www.statmt.org/wmt06/shared-task/dev2006.de.gz

[10] http://matrix.statmt.org/test_sets/test2007.tgz?150472237

[11] We used http://matrix.statmt.org/test_sets/newstest2013.tgz?1504722373.

[12] http://statmt.org/wmt15/translation-task.html

[13] We did not contact their authors in the submission phase as we assumed the replicability test needed to make the most of what was made explicit in the paper and was part of the challenge. In the second phase, we got access to part of the data thanks to Eva Vanmassenhove.

2.3. Features

We share a similar goal (provide better NMT models with linguistic annotation), but when we replicated the experiments, we found that the most specific annotation features were insufficiently characterised in the original paper. In this sense, reproducibility proved ruthless for annotation specifications.

2.3.1. The Workflow

For clarity's sake, we spell out the annotation workflow, which can partly be reconstructed from examples (1) to (6) in section 3 of the original paper. The original paper aimed at "a combination of both syntactic and semantic features" and present a four-fold set of results in Table 1 & 2 and Figure 2 & 4: BPE (baseline) / CCG / SST / SST–CCG (combined). With the concept of supertags, they subsumed syntactic (CCG) and semantic (SST) supertags and their combinations. They begin with the presentation of semantic features, which is questionable as (i) it is more complex in its implementation and (ii) some of the semantic features actually pre-suppose a preliminary syntactic annotation. These supertags combine several operations that need to be detailed when replicating the annotation. An example of CCG is given in (6), reproduced below.

The semantic supertag (SST) combines two features:

```
It|NP is|(S[dcl]\NP)/NP a|NP/N modern|N/
N form|N/PP of|PP/NP colonialism|N .|.
```

Figure 1: CCG annotation

a semantic annotation of nouns and verbs inspired by Wordnet (Miller, 1995) and a labelling of multi-word expressions (MWE) trained on web-based data (Schneider et al., 2014) incorporated in the version 2.0 of AMALGrAM (A Machine Analyzer of Lexical Groupings and Meanings (Schneider and Smith, 2015)). Lexical units perceived as MWE are joined by an underscore, semantic labels for nouns are capitalised and added after a pipe, semantic labels for verbs are not capitalised). Here is an example: It presupposes pos-tagging as ver-

```
I was|stative particularly horrified|
cognition by the idea|COGNITION of
Mrs_Randzio-Plath|GROUP that we should
actually enrol|cognition schoolchildren|
PERSON as the stormtroopers|GROUP of this
euro propaganda|COMMUNICATION blitz|ACT .
```

Figure 2: CCG annotation

tical tabulated input as in 3:

2.3.2. Combining features

To be fair, we did run into trouble of our own making by mismanaging the computational power, time and space required for the CCG and STT annotation when dealing with

```
It          PRP
is          VBZ
a           DT
modern      JJ
form        NN
of          IN
colonialism  NN
.           .
```

Figure 3: Pos-tagging assumed by AMALGrAM

1M sentences. We only managed to annotate 578,865 sentences for CCG in 19h and crashed our machine with the SST annotation due to space allocation. Splitting the Europarl into smaller files solved our problem, but too late for proper combinations of NN training phases. We underestimated the memory and time requirements for the final completion of the CCG and SST annotations, however, some aspects of the combining features remain sufficiently undocumented to cast doubt on our ability to have reproduced the proper training data in the final stage of the annotation which is supposed to combine features.

- **computational resource**: the semantic annotation is very greedy as it goes through several lexicons and also requires pos-tagged data as input. If pos-tagging was done at about 10,0930.48 tokens per second, SST annotation was much slower (about 10 tokens per second, 471.77 seconds to annotate 4800 tokens).

- **status of the separator**: we assume OpenNMT takes only one separator when the combination of syntactic and semantic supertags lead to potentially two separators for the same token.

- **post-processing and BPE**: the original paper begins with the semantic tagging, discussing the supertag, then adds a layer for multiword expressions and then explains how to integrate BPE. Examples (3) to (5) suggest they first copied the semantic tags, then the MWE labels and re-introduced the BPE encoding. This post-processing and re-integration of BPE is even more striking when combined with the syntactic annotation.

3. Reimplementing the approach

Our experiments were carried out with the PyTorch version of OpenNMT. We used the following functions: onmt_preprocess, onmt_train and onmt_translate. The BPE pre-processing was left at 89,500 operations. OpenNMT is a 2-layer BiLSTM translation toolkit and we trained the models with hidden size 500 for 20 epochs. We ran the training and calculation on a computer with Intel® Core™i7-7700K using a Debian GNU/Linux 10 distribution and equipped with an NVIDIA GeForce GTX 1080 Ti GPU. Each training phase took about 4 hours for each experiment and the translation phase (two test tests of about 5K sentences) took an hour. The detailed parameters and corresponding scripts can be found on the corresponding gitlab (https://gitlab.com/nballier/reprolang2020).

3.1. Experimental Setup

The original paper did not report performance differences for the training phase but speed of convergence. OpenNMT being a different ecosystem for neural machine translation than Nematus, we needed to map the parameters used in the initial paper with Nematus on OpenNMT. The specifications of the training were clearly laid out in the original paper, which allowed us to reproduce them with OpenNMT, prompting us to use the functions onmt_preprocess for the preprocessing step and onmt_train for training.

Nematus	OpenNMT
Vocabulary Size	src_vocab_size
Maximum Sentence Length	src_seq_length
Vector Dimension	feat_vec_size
Word Embedding Layer	src_word_vec_size
Learning Optimizer	optim

Table 1: Parameters of the original paper and their equivalent in OpenNMT

3.2. Problems with Reimplementation

Two main issues arose when reading the original paper. First, no code was provided, which led to make decisions when the quoted linguistic examples did not cover the cases we encountered for annotation, especially for post-processing. Having the codes would have helped, a point already made in the replicability literature (see for instance Rahmandad and Sterman (2012)). By contrast, we have made the code we used for post-processing available on Github. Second, the test data was not detailed in the paper, which proved highly challenging.

3.2.1. Data Incertainty

We explained how we produced our data. For Europarl, we retrieved the English examples given in the original paper text, so that we are vindicated in choosing to extract the first million sentences from the Europarl repository. For the newstest2013 and Europarl test set, we are much less assertive.

3.2.2. The pre-processing phase

What is presented as the baseline is the BPE segmentation, which we take to be a form of annotation, not the raw text. Since the BPE is also presented a secondary step on supersense tags in example (3), this is a bit counter-intuitive for a naive reconstruction of the steps of the experiments. The post-processing phase, which adds $|mwefordetectedMulti-WordExpressions(MWE)reliesonalistofassumedMWEssuchas$

3.2.3. The annotation features

The range of annotation features was not covered with sufficient examples to reconstruct the fully annotated datasets (separately and in combination with the semantic ones), especially for pos-tags and CCG supertags in what they call the "ALL combined" configuration. No linguistic example was provided

of an input sentence with all the tags combined. Some detailed requirements of the annotation procedures were not covered in the description of the annotation phase. For example, supersense tagger takes as input pos-tagged files in vertical format. The AMALGrAM supersense tagger potentially returns a tabular format or requires the 'cut' functionality to present data like in example (5) of the original paper. Similarly, EasySRL can take pos-tagged texts as input and these specifications were not indicated in the original paper.

3.3. Results

This subsection gives an overview of the results, with a specific emphasis on the comparables. The REPROLANG call for papers described the expected figures and results. We used python scripts to plot the major reproduction comparables. We created datasets with the number of iterations used in the figures of the original paper and the corresponding BLEU scores and plotted them.

3.3.1. Feature Ablation Procedures

Even though they have a very interesting section 2 on previous experiments and make the point that the results of the annotations are not necessarily cumulative, the authors of the original paper mostly reported the score obtained with all the features combined against their baseline. We felt the need to distinguish BPE from raw texts and to distinguish between pos-tagging and syntactic supertags (CCG). Similarly, their semantic annotation (SST) does not distinguish the Wordnet-derived labels and the MWE-tagging. Though it multiplies the number of training phases, this analysis of features might shed light on what has actually been learnt with each level of annotation.

3.3.2. The Baseline

The original paper describes the PBE pre-processing as their baseline (without reporting the number of operations used for the target language). Because we believe BPE is a form of annotation, at some point, the authors refer to "the best BPE-ed baseline model", we first plotted BPE against raw texts, which we deem to be strictly the real baseline in terms of annotation procedures. We assumed the BLEU scores were obtained in translating from English to French and English to French, but maybe they averaged scores, whereas other papers report BLEU scores in both directions (Belinkov et al., 2017). We decided to average BLEU scores. Predictably enough, the BLEU scores are below the ones visible in the original paper (between 21.5 and 22,5), probably because of many UNK (unknown words) subside as out-of-vocabulary words due to the absence of any procedure aiming at reducing them. Nevertheless, it should be noted that a similar stabilisation of the BLEU scores can be observed between 60,000 and 80,000 iterations. The scores are much lower on newstest13, which is more dissimilar to the Europarl training data than the Europarl-based test set.

3.3.3. The Individual Benefit of pos-tagging

The original paper combines two operations for their "syntactic features [...]: POS tags and CCG supertags" and only report "Syntactic (CCG)" scores. We wanted to report the

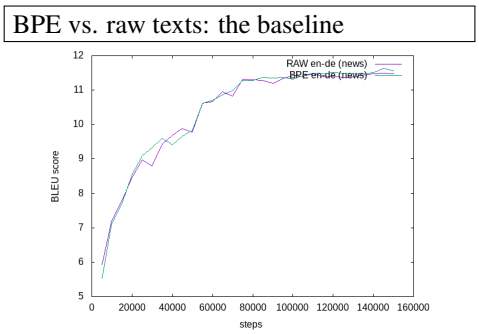

Figure 4: raw texts vs. BPE Systems for EN–DE, evaluated on the newstest2013

BLEU scores obtained with pos-tagging, before showing the difference with the syntactic "supertag" (CCG).

3.3.4. the Syntactic Supertags (CCG)

Our replication is only partial as we only trained on a half a million tokens. We were not able to train models on the pos-tagged and CCG-annotated data due to technical problems in the submission phase.

3.3.5. Semantic Supertags and Combined Features

We explained in the previous section why we did not manage to produce the comparables for the semantic annotations. We make the point that feature ablation would have required us to distinguish the MWE and the wordnet-based annotation in the supplementary plots.

4. Annotation Experiments and Projects for Improving the Reimplementation

In this section, we present several methods aiming at improving the experiments from the point of view of annotation. With the same aim to prove the point that more adequately trained data provided better BLEU scores, we discuss alternative enrichment of the training data, making suggestions with the target texts as well, when comparable resources are available.

4.1. Linguistic Input in the Pre-processing Phase

Discussing learnability of the input with BPE may challenge the possibility of exploiting character-based language models. The sub-word analysis carried out by BPE is puzzling to the human eye. We analysed some of the BPE encoding output for French. Although this seems to be current practice for Neural Machine Translation (Senellart et al., 2017), the precision and recall seems highly questionable for the sub-word analysis of French from a linguistic point of view. The first sentence we ran for a test contained "is@@ o@@ lée ni s'isol@@ er", where only the latter is correct for the flectional boundary and the isolation of the infinitive morpheme er. We nevertheless trained the model with the two pre-processsed texts (for French, (Servan et al., 2017) have 30,000 operations, not 89,500) and this parameter does not seems to be discussed in the original paper. It may well be the case that NN learn better with the BPE pre-processing phase. Two experiments appeal to us.

One way to solve this dispute might be to plot vocabulary growth curves, plotting types against BPE-ed tokens. For incremental analysis of the lexical input, vocabulary growth curves as plotted in R with the ZipfR package (Evert and Baroni, 2007) could be tried to measure the actual limitation of vocabulary entailed by the Byte Pair Encoding (see section 5.3 below). The other alternative would be to do sub-word division on morphological grounds. Small-scale tests seem to suggest that pre-processing for French would be more efficient with specialised morphological parsers designed for French such as FLEMM (ATILF, 2008). Alternatively, specialised morphological taggers may provide better results with subword units. This could be tested with tools such as chipmunk (Cotterell et al., 2015), which provides "labeled morphological segmentation" for languages like German and English with F1 scores of 86.31 and 87.85, respectively, (Cotterell et al., 2015) .

4.2. Competing Tagsets for the Input

The paper chose the Stanford tool (Toutanova et al., 2003) but did not detail the model used for the tagging, whereas at least five models are available on the Stanford website. Comparing the efficiency of tagsets would definitely be a paper of its own, bearing in mind that precision depends on the training model. It is an empirical question, nevertheless triggered by competing tagsets and differences in the training data. The performances of the tools are not reported in the original paper (see next section) but the training corpora play a role for the relevance of the training.

4.3. Annotating the Target Texts in the Training Phase

The second question that came to mind is the status of the target training files, which we supposed were not annotated. Because several language models were created for the Stanford pos-tagger, it could be possible to give as input a source text and a target text that are both pos-tagged, with the hope than some 'mapping' between pos-tags could be learned by the NN. We used the corresponding French and German models to annotate the target training files. For pos-tagging with Stanford for English we used the Stanford POS Tagger.[14] It should be noted that the German component of Europarl is not quite equivalent to the English/French translation and its pos-tagging raise different issues with tagging numbers (phone number extensions). We cleaned the data as the pos-tagger generated many mistakes with figures. In many cases, only the first part of the number was tagged. 300 000 gets to be tagged as 200—CD 000. With the inconvenient that a date may be concatenated to a figure, we did some post-processing for pos-tagging with a clean.py script that underscores numbers.

[14]Java Version: `https://nlp.stanford.edu/software/stanford-postagger-full-2018-10-16.zip`
Python Version: `http://www.nltk.org/_modules/nltk/tag/stanford.html#CoreNLPPOSTagger`
For German: `https://nlp.stanford.edu/software/stanford-postagger-full-2016-10-31.zip`
For French: `https://nlp.stanford.edu/software/stanford-postagger-full-2014-06-16.zip`

4.4. Using Upos as Tagset for the Input

Cross-linguistic annotation aiming at consistency is the general project for the upos (universal part of speech) (Nivre et al., 2016) in Universal Dependency. One of the great strengths of this project is that the pos-tagging of the target and source languages are not based on a language-dependent tagset. This suggests an experiment we deem important for learnability: using upos annotation for target and source language. Re-tagging the pos-tags of the target and source texts (as well as the validation sets) with upos universal part of speech) would also require to specify the tagging models. Again, several models are available, and we feel the need to specify them. For English, we would used the Partut model and for French the Sequoia model, which was partly trained on French Europarl (Candito and Seddah, 2012).

4.5. MWE Annotation

Vanmassenhove and Way (2018) refer to standard definitions of Multi-Word Expressions (MWE) "a group of tokens in a sentence that cohere more strongly than ordinary syntactic combinations" and make room for the terminological diversity in the field ("fixed expression, formulaic sequence, fossilized idiom, phraseological unit, and prefabricated pattern"). We would like to make three points:

- **copy tag format**: as a result of the post-processing scripts that copy the MWE features to the components of the MWE, MWE are tagged by a form of an "adjacency tag". The tag —mwe is just copied to every adjacent token of the mwe so that it seems unlikely that the boundaries of the MWE are learnt, since they are not coded. This could perhaps be done with the IOB format. This IOB (Inside, Outside, Beginning) format was first introduced by (Ramshaw and Marcus, 1999) and a similar IOB feature representation was adopted for dependency tags in the input, as experimented for Nematus by (Sennrich and Haddow, 2016) for translations between German and English. Their feature representation was clarified by a diagram showing the correspondence between the dependency annotation and a list of features assigned to tokens. They reported the improvement "of 1.5 BLEU for German→English, 0.6 BLEU for English→German, and 1.0 BLEU for English→Romanian" (Sennrich and Haddow, 2016, p. 87).
- **MWE and pos-tags mismatch**: MWE may be seen as a mix of syntactic and semantic properties, so much so that the actual pos-tag for the MWE might be different from the individual tags of the words they are made of. In *In fact*, which is a MWE, *in* is PREP and *fact* is N, but we should label the MWE *in_fact* as ADV. This property of MWE does not seem to have been controlled in the original paper.
- **Phraseology and Specialised Texts**: The MWE found in the training sets are the one detected with the original training corpus of the tool, and may not take into account the phraseology found in the Europarl data (see, in contrast, (Granger and Lefer, 2016; Ustaszewski, 2019).

4.6. More Annotation Features

The original paper resorted to several annotation tools but not the full range of annotations available in each tool they used. For example, Figure 1 illustrates some of the many annotation possibilities of EasySRL. This "html option" develops the dependency parsing (the arrows), the syntactic and logical form decomposition of sentence (6) used in the original paper: "It is a modern form of colonialism". The original paper has addressed semantic tags and pos

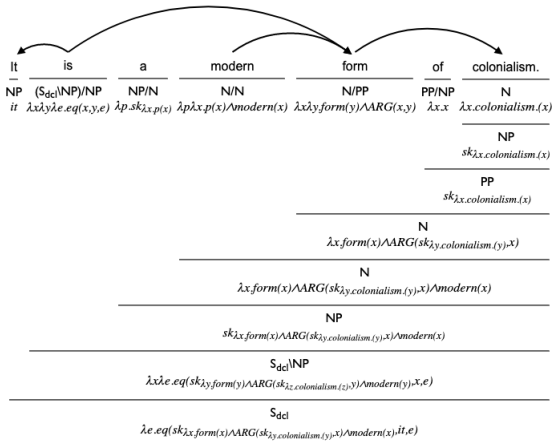

Figure 5: The multi-layered annotation possibilities of EasySRL (after Lewis, 2015)

tags but has eschewed parsing tags. Pos-tagging only consists in "shallow parsing", syntactic structures are annotated at a surface level, whereas parsing provides more syntactic structures. It could be possible to use a parser to provide the training sets with deep parsing information, especially with the SpaCy python library to annotate the data. Sennrich and Haddow (2016) experimented with the parsing features with the Nematus toolkit. More practically, the authors do not seem to have considered Named Entity Recognition among the annotated features. This feature has been analysed for automatic detection across languages from more than fifteen years (Tjong Kim Sang and De Meulder, 2003). Although the accuracy rates reported for Named Entities Recognition are much lower (80% according to spaCy) than for pos-tagging, it could be interesting to annotate the training sets with NER tags. This could result in more reliable annotations than in the current AMALGrAM annotation for proper nouns. Sequences like *Mr X* have been tagged in the English corpus as either |GROUP or |PERSON.

4.7. Manual Evaluation/Human Evaluation?

Oddly enough, the original paper did not compare the BLEU scores they obtained with the scores obtained with the same data in the 2005 MT competition. [15] More generally, the precision of the tools used during the annotation phases is neither reported nor discussed. Manual evaluation is evoked for future research in the original paper. Now, what if the system performed better but the labels were wrong? A subsection detailing the performance of the tools might be in order here. For pos-tagging, Stanford taggers seem to perform above 96% precision. For parsing models for Combinatory Categorial Grammar (CCG) (Clark and Curran, 2004) reported F-scores ranging from 84.8 to 92.5. Semantic labelling is more recent and more complex and we are not aware of evaluation campaigns on these two semantic embedded tasks (Wordnet-derived hypernym labelling and MWE labelling) which correspond to the semantic supertags. We can only surmise the relevance of the process by looking at the size of the lexicons we used in the annotation phase. For an estimation of the covered vocabulary, we give the number of lines of the lexicon used to do the supersense annotation (147,306 entries for wordnet_supersenses.json). The authors did not report the precision scores of the supersense evaluation (though a script allows to test the results against the golden standard corpus for MWE, STREUSLE (Supersense-Tagged Repository of English with a Unified Semantics for Lexical Expressions). [16] For the detection of multiword expressions, users may retrain the AMALGrAM tool and the model is stored in a lexicon, which by default is empty (said.json). For this baseline, with an empty said.json, the AMALGrAM documentation indicates that 'A user who trained and tested a model without SAID features (said.json) reports obtaining F1 scores of 61.37% for MWEs and 70.12% for supersenses'.

The more general question is: do supertags translate? It should be noted that supersense tagging was successfully used in relation with Machine Translation, in order to annotate Arabic using English supersenses and back translation (Schneider et al., 2013). A possible experiment would consist in trying to train the target texts with a similar system for semantic annotation. For French, a free version of Wordnet has been developed, WOLF (Wordnet Libre du Français), which reported in its early stages a precision rate of 83% for nouns and 75% for verbs (Sagot and Fišer, 2008). For German, a GermaNet (lexical-semantic net for German) has been developed with very similar categories to Wordnet, except maybe for stative verbs and state and process nouns (Hamp and Feldweg, ; Henrich and Hinrichs, 2010). A possible way to analyse the translatability of supertags would be to tag the target texts when the resources exist. As to using multingual tagging of MWE, we are not aware of fully-fledged system, though progress have been made for verbal MWE (Savary et al., 2018). This point is debatable since the test sets are comparable but not identical to the original sets, nevertheless, it would have been useful to run the semantic parsers on the two test sets. How many Multi-Word units were present in the two test sets and does this correlate with the results noted on Figure 2 and 4 of the original paper? Intuitively, the annotation of the MWE gold corpus used to train the MWE annotation tool was made of web-based material, so that it is probable that the news2013 test set includes MWE more likely to be detected. This would explain why SST fares

[15] The scores of the best systems are listed on http://www.statmt.org/wpt05/mt-shared-task/#TEST.

[16] https://github.com/nert-nlp/streusle/

better in Figure 2 with news2013 in the original paper.

5. Further Research

This subsection gives the gist of what we learnt from this replication task for neural networks. We have not tried reciprocal replicability: test our re-generated data with the Nematus toolkit, in the same way that the test sets can be used to be translated in the two directions. We used the PyTorch implementation and it is tempting to replicate the experiments with the Tensorflow implementation of OpenNMT, possibly resorting to Tensorboard to monitor some aspects of the training phase, especially the Tensorboard logging parameters for further analysis.

5.1. What is Learned with the Annotation?

We have tested the annotated data with the pipe as a unique tag separator (a possibility hinted at in the OpenNMT documentation). It should be noted that the separator in the training data is not the standard pipe symbol but Unicode FFE8. From the point of view of the learnability of the input, there is great uncertainty about the input as to the separation and interpretation of tags. We have decided to use this separator as a generic tag for every feature, which begs the question of the learnability of several tags assigned to a token? Most systems do not seem to cater for possible asymmetries, so that when only nouns and verbs are granted a semantic supersense tag, this has some consequences in feature representation as dummy tags need to be added for the other categories and these adjacent successive dummy tags may not be distinguished from MWE tags.

5.2. Cumulative Features or Hybrid Features?

We are uncertain as to the configuration of the tagging retained for the combined features, probably a double tagging with a grammatical layer followed by a semantic layer.

The post-processing phase for the Wordnet-based analysis assigns a default tag ($|none$) to all the tokens that are not nouns and verbs and the post-processing of MWE copies this feature across the components, to the detriment of the Inside, Outside, Beginning (I,O,B) labels that are actually actually captured by AMALGrAM annotation. The generalisability of these default and copied tags can be questioned. The default tag multiplies uninformative tags that are not as specific as pos-tags. For example, articles are either assigned a default tag ($|none$) in example (3) in the original paper, or a copied tag ($|mwe$) in example (5) for the MWE: a$|mwe$ number$|mwe$ of$|mwe$. A solution to this kind of contradictory tagging would be to use a hybrid system assigning a single tag. Assuming the semantic categories for verbs assigned by AMALGrAM/wordnet are more relevant than indirect aspectual information encoded in the pos-tagging (VBD, VBG, VBZ), here is a potential procedure (or roadmap for future experiments): define an algorithm that best captures the specificities of features to be learned by the NN: 1. Apply Named Entity Recognition for consistent NNP tagging; 2. Annotate Multiword Expression using I,O,B tags; 3. CCG tagging (provided disambiguation is really effective); 4. pos-tagging; 5. replace noun and verb pos-tags with their semantic labels provided by AMALGrAM/wordnet.

5.3. Monitoring the Input Quantitatively

LNRE (Large Number of Rare Events) models (Baayen, 2001; Evert, 2004) can be used to compute the number of observed types when the size of the corpus is increased. We could use this as a proxy for the quantification of out-of-vocabulary words when the size of the training set increases. We found that above 400,000 tokens, the number of hapaxes decreases in the BPE format as compared to raw data.

5.4. Gradient Descent Algorithm Optimization

A potential parameter to investigate in relation to the variability of the annotation of the input is the Learning Optimizer. In the original paper, the learning optimizer used is Adadelta (Zeiler, 2012), an extension of the Adagrad algorithm.

Gradient descent is a way to minimize an objective function $J(\theta)$ parameterized by a model's parameters $\theta \in \mathbb{R}^d$ by updating the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$ with respect to the parameters. The learning rate η determines the size of the steps we take to reach a (local) minimum. In other words, we follow the direction of the slope of the surface created by the objective function downhill until we reach a valley. Adagrad (Ruder, 2016) is an algorithm for gradient-based optimization that adapts the learning rate to the parameters, performing smaller updates for parameters associated with frequently occurring features, and larger updates for parameters associated with infrequent features. For this reason, it is well-suited for dealing with sparse data. Pennington et al. (2014) used Adagrad to train GloVe word embeddings, as infrequent words require much larger updates than frequent ones. Adagrad's main weakness is its accumulation of the squared gradients in the denominator: since every added term is positive, the accumulated sum keeps growing during training. This, in turn, causes the learning rate to shrink and eventually become infinitesimally small, at which point the algorithm is no longer able to acquire additional knowledge. Adadelta aims to solve this flaw. Adadelta is an extension of Adagrad that seeks to reduce its monotonically decreasing learning rate instead of accumulating all past squared gradient. Adadelta restricts the window of accumulated past gradients to some fixed size ω. To improve the BLEU scores, it could be possible to tweak the Gradient Descent Algorithm Optimization by testing alternative algorithms in relation to the variation of the linguistic input.

6. Conclusion

In this paper, we tried to reimplement the approach to training neural networks for translation with annotated data by (Vanmassenhove and Way, 2018). We first replicated it with a different Neural translation toolkit, to check some assumptions/facts about NN architecture. We then tried to improve the quality of the translation by enriching the linguistic annotation of the input, taking into account several layers of annotation. Prompted by one reviewer to rate the reproducibility difficulty, we would give a 4, rating from 1 to 5, 5 being the most difficult. This is true for reproducing the paper 'as is', but with the

data we received from the first author, 2 would be more accurate. Despite some shortcomings in the description of the annotation workflow we underlined, we were able to generate the (rare) example sentences given in the original paper with the same annotations. A point to be noted is that linguistic examples proved crucial in the replication phase. Another take home message for replicability is that a tool is not a tagset, so that for annotation, relying on tools for the description of an annotation procedure is insufficient and potentially misleading. Tagsets (and hopefully models or training corpora) should be specified. Their assumed reliability should be mentioned, for example by reporting F-scores or at least precision of the tool.

To be *FAIR* to the authors of the original paper, scrutinizing their paper in terms of reproducibility needs to be contextualised within the FAIR Paradigm (Mons, 2018), which suggests that research should be

- **Findable**: URLs are rarely eternal, but short of indications, the data we could retrieve is not guaranteed to be the one used in the original paper for the test sets. This volatility proved to be true for data and tools alike. The easySRL tool is still on github, but some of the subcomponents (the lexicon files) were found on another github which had been set for production. [17]
- **Accessible**: Undeniably, working with materials made public helped in this replicability study. There is no need to insist on how the availability of datasets helped from Natural Language Processing or Machine learning. It should nevertheless be seen as a source of satisfaction that the 2005 Europarl material can be found and re-used fifteen years later. However re-annotating a corpus several times with several tools also made us aware of some issues in the original release. [18] For French, it seems that some utf8 encoding issues seem to remain, judging by some of the quirks in the data in Figure 6:

```
"La proposition actuelle prévoit d'ouvrir
le droit ▯ la retraite des députés
européens ▯ l'âge de 63▯ ans, alors que le
groupe confédéral de la gauche unitaire
européenne pense que cet âge devrait être
abaissé ▯ 60▯ ans.
```

Figure 6: Some encoding issues spotted in Fr.Europarl

  The "EF BF BD" UFT8 has been used in lieu of "C3 A0" UTF8. Perhaps a more normalised version of Europarl, with more metadata, could be envisaged for the 15th anniversary of the resource. Similarly, there are probably in-house cleaner pos-tagged versions of this corpus that could be shared.

- **Interoperable**: the core of our paper is to reproduce the experiment done on Nematus with OpenNMT. In this sense, neural network architecture push the limits of the interoperability concept even further. Using several tools (across different platforms) questioned dependencies. The semantic supersensetagger relies on a version of python 2.7 which is no more compatible with the required nltk library (Loper and Bird, 2002), not to mention the status of python 2.7 in years to come. This point about the potential frailty of tools ("project survivability") was already made by (Pedersen, 2008).
- **Reusable**: The current workshop added an additional constraint with the Docker requirement on gitlab. [19] What should be reproducible, the scores (as suggested in the call for papers) or the whole process? We took the view that annotated corpora were an input (we would have liked to find the annotated corpora to just run the training and translating experiments) and our Docker image processes the annotated input up to the plotted figures, but it would make sense to design a whole Docker for the complete workflow from raw texts to BLEU scores. We have settled for a middle ground policy that included on another gitlab the scripts we used to process the annotation phases, some of them allow users to chose their specific desired models for pos-tagging, to emphasise the importance of tagsets.

The next step in our analysis of the learnability of the input for NMT should be combined with techniques visualising the activity of neural networks, in order to get closer to to the desired interpretability (Taylor, 2006) of some of the processes of the training and translation phases. (Montavon et al., 2018) make a distinction between interpretation ("the mapping of an abstract concept (e.g. a predicted class) into a domain that the human can make sense of" and explanation ("the collection of features of the interpretable domain, that have contributed for a given example to produce a decision (e.g. classification or regression)"). It seems to us that reproducibility of results across machine translation toolkits is a first step in the exploration of interpretation and that annotated features are a good candidate for this methodology. Even more so as pioneering attempts at visualising the annotation activity in neural networks has been successful in the recent Blackbox series of workshops (Linzen et al., 2018; Linzen et al., 2019) or as (Belinkov et al., 2017) have managed to shed light on the apparent division of labour between lower layers of the NMT encoder (word structure) and higher layers (word meaning). The models we trained with these different annotations could be tested with the system developed for the paper to observe what has been learnt for each type of syntactic or semantic input.

7. Acknowledgements

Thanks are due to the three reviewers for their suggestions. We thank Benoit Crabbé for advice on parsing and Marie Candito for advice on the annotation of MWEs and Eva

[17] https://github.com/brombach/ss_http/

[18] See also examples of "inconsistent and incorrectly encoded source language identifiers in Europarl source files" spotted in (Ustaszewski, 2019)

[19] See our https://gitlab.com/jbyunes/repro (Hash Commit 1a22f377a4a1840f5db1e939f6695eb0a4235b04, Tag : Release-1.2

Vanmassenhove for sharing her data after the submission phase. We also thank Maria Zimina and Chris Gledhill for comments on an earlier version of this paper.

8. Bibliographical References

ATILF. (2008). Flemm. ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

Baayen, R. (2001). Word frequency distributions.

Belinkov, Y., Durrani, N., Dalvi, F., Sajjad, H., and Glass, J. (2017). What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada, July. Association for Computational Linguistics.

Branco, A., Calzolari, N., and Choukri, K. (2018). *4REAL 2018 Workshop on Replicability and Reproducibility of Research Results in Science and Technology of Language*. European Language Resources Association.

Candito, M. and Seddah, D. (2012). Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *TALN 2012 - 19e conférence sur le Traitement Automatique des Langues Naturelles*, Grenoble, France, June.

Clark, S. and Curran, J. R. (2004). Parsing the wsj using ccg and log-linear models. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 103. Association for Computational Linguistics.

Cotterell, R., Müller, T., Fraser, A., and Schütze, H. (2015). Labeled morphological segmentation with semi-markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 164–174.

Evert, S. and Baroni, M. (2007). zipfR: Word frequency distributions in R. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 29–32. Association for Computational Linguistics.

Evert, S. (2004). A simple LNRE model for random character sequences. In *Proceedings of JADT*, volume 2004, pages 411–422.

Granger, S. and Lefer, M.-A. (2016). From general to learners' bilingual dictionaries: Towards a more effective fulfilment of advanced learners' phraseological needs. *International Journal of Lexicography*, 29(3):279–295.

Hamp, B. and Feldweg, H.). Germanet-a lexical-semantic net for German. In *Automatic information extraction and building of lexical semantic resources for NLP applications*, pages 9–15.

Henrich, V. and Hinrichs, E. (2010). Gernedit-the germanet editing tool. In *Proceedings of the ACL 2010 System Demonstrations*, pages 19–24.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. M. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Laje, R. and Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature neuroscience*, 16(7):925–933.

Tal Linzen, et al., editors. (2018). *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, November. Association for Computational Linguistics.

Tal Linzen, et al., editors. (2019). *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy, August. Association for Computational Linguistics.

Loper, E. and Bird, S. (2002). NLTK: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics -*, volume 1, pages 63–70. Association for Computational Linguistics.

Miller, G. A. (1995). Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Mons, B. (2018). *Data stewardship for open science: Implementing FAIR principles*. Chapman and Hall/CRC.

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pedersen, T. (2008). Empiricism is not a matter of faith. *Computational Linguistics*, 34(3):465–470.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Popper, K. (2005). *The logic of scientific discovery*. Routledge.

Rahmandad, H. and Sterman, J. D. (2012). Reporting guidelines for simulation-based research in social sciences. *System Dynamics Review*, 28(4):396–411.

Ramshaw, L. A. and Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Repar, A., Martinc, M., and Pollak, S. (2018). Machine learning approach to bilingual terminology alignment: Reimplementation and adaptation. In António Branco, et al., editors, *4REAL 2018 Workshop on Replicability*

and Reproducibility of Research Results in Science and Technology of Language, pages 1–8.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Sagot, B. and Fišer, D. (2008). Construction d'un wordnet libre du français à partir de ressources multilingues. In *Proceedings of TALN2008,Traitement Automatique des Langues Naturelles, Jun 2008, Avignon, France.*

Savary, A., Candito, M., Mititelu, V. B., Bejček, E., Cap, F., Čéplö, S., Cordeiro, S. R., Eryiğit, G. C., Giouli, V., Van Gompel, M., et al. (2018). PARSEME multilingual corpus of verbal multiword expressions. In *Multiword expressions at length and in depth: Extended papers from the MWE 2017 workshop*. Zenodo.

Schneider, N. and Smith, N. A. (2015). A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1537–1547, Denver, Colorado, May–June. Association for Computational Linguistics.

Schneider, N., Mohit, B., Dyer, C., Oflazer, K., and Smith, N. A. (2013). Supersense tagging for Arabic: the MT-in-the-middle attack. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 661–667, Atlanta, Georgia, June. Association for Computational Linguistics.

Schneider, N., Onuffer, S., Kazour, N., Danchik, E., Mordowanec, M. T., Conrad, H., and Smith, N. A. (2014). Comprehensive annotation of multiword expressions in a social web corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 455–461, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Sennrich, R. and Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, August. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), Berlin, Germany*.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A. V. M., Mokry, J., et al. (2017). Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65—-68.

Servan, C., Crego, J., and Senellart, J. (2017). Adaptation incrémentale de modèles de traduction neuronaux. In *Proceedings of TALN2017,Traitement Automatique des Langues Naturelles, Jun 2017, Orléans, France.*

Taylor, B. J. (2006). *Methods and procedures for the verification and validation of artificial neural networks*. Springer Science & Business Media.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for computational Linguistics.

Ustaszewski, M. (2019). Optimising the europarl corpus for translation studies with the europarlextract toolkit. *Perspectives*, 27(1):107–123.

Vanmassenhove, E. and Way, A. (2018). Supernmt: Neural machine translation with semantic supersenses and syntactic supertags. In *Proceedings of ACL 2018, Student Research Workshop*, pages 67–73.

Wang, R., Zhao, H., Lu, B.-L., Utiyama, M., and Sumita, E. (2016). Connecting phrase based statistical machine translation adaptation. pages 3135–3145.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.