GDBNMT - Reducing Gender Bias in Neural Machine Translation as a Domain Adaptation Problem

Contents

1 Introduction to the Practical

This practical is based on recent work aimed at remediating gender bias in neural machine translation as a domain adaptation problem [1]. Quoting from the paper:

> Natural language training data inevitably reflects biases present in our society. For example, gender bias manifests itself in training data which features more examples of men than of women.
>
> Gender bias is a particularly important problem for Neural Machine Translation (NMT) into gender-inflected languages. An over-prevalence of some gendered forms in the training data leads to translations with identifiable errors. Translations are better for sentences involving men and for sentences containing stereotypical gender roles. For example, mentions of male doctors are more reliably translated than those of male nurses.
>
> ... we propose treating gender debiasing as a domain adaptation problem, since NMT models can very quickly adapt to a new domain. ... We consider three aspects of this adaptation problem: creating less biased adaptation data, parameter adaptation using this data, and inference with the debiased models produced by adaptation.
>
> Regarding data, we suggest that a small, trusted gender-balanced set could allow more efficient and effective gender debiasing than a larger, noisier set. To explore this we create a tiny, handcrafted profession-based dataset for transfer learning.
>
> We find that during domain adaptation improvement on the gender-debiased domain comes at the expense of translation quality due to catastrophic forgetting. ... We can balance improvement and forgetting with a regularised training procedure, Elastic Weight Consolidation (EWC), or in inference by a two-step lattice rescoring procedure.

This Practical focuses on the two-step lattice rescoring procedure presented in sections 2.3.2, 3.3, and 3.4.4 of Saunders' ACL 2020 paper. The aim is to adapt a baseline NMT models to a gender-balanced adaptation set with the aim of improving accuracy while carrying out a two-pass decoding procedure to prevent any degradation in overall translation quality.

1.1 Baseline NMT Models

This practical is based on the Facebook-FAIR English-to-German translation systems developed for the 2018 Workshop on Machine Translation (WMT'18). The Facebook-FAIR system is an *ensemble* of six Transformer models $P_i(y|x)$, with translation hypotheses chosen as

$$\hat{y} = \underset{y}{\text{argmax}} \sum_{i=1}^{6} P(y|x;\theta_i)$$

[1] Saunders, Byrne. *Reducing Gender Bias in Neural Machine Translation as a Domain Adaptation Problem*, ACL'20, https://www.aclweb.org/anthology/2020.acl-main.690/

The Facebook-FAIR system was ranked best overall in human quality assessments [2] although the WMT'18 leaderboard [3] indicates the system was not best under BLEU - an instance of BLEU failing to predict human preferences for highly optimised systems. Note that the Cambridge system based on Felix Stahlberg's SGNMT decoder, which you will use for this Practical exercise out-performed the Facebook-FAIR ensemble system in terms of BLEU. We could use that system for this practical, except that it is based on a now out-of-date version of TensorFlow/Tensor2Tensor.

In this practical you will use Fairseq/PyTorch models distributed by Facebook-FAIR. These are provided for you in the Practical directories on the HPC. More info on the models and toolkits is available on GitHub[4]. **N.B.:** These baseline models were developed by Facebook-FAIR for the WMT'18 international evaluation. Their baseline performance is consequently very good and their baseline gender accuracy is also correspondingly very high.

1.2 Gender-Adapted NMT Models

As described in Section 2.2.1 of Saunders ACL'20,

> Our hypothesis is that the absence of gender bias can be treated as a small domain for the purposes of NMT model adaptation.
>
> We therefore construct a tiny, trivial set of gender-balanced English sentences which we can easily translate into each target language. The sentences follow the template:
>
> The [PROFESSION] finished [his | her] work.
>
> We refer to this as the *handcrafted set*. Each profession is from the list collected by Prates et al. (2019) [5] from US labour statistics. We simplify this list by removing field-specific adjectives. For example, we have a single profession 'engineer', as opposed to specifying industrial engineer, locomotive engineer, etc. In total we select 194 professions, giving just 388 sentences in a gender-balanced set.

The adapted models used in this practical simply takes one of the component models of the Facebook-FAIR WMT'18 ensemble and refines it with a few (10) iterations of training (SGD) over this small, gender-balanced set. These adapted models are provided for you.

For interest, but not for this practical, there is subsequent work by Cambridge fourth year Engineering student Rosie Sallis investigating even simpler schemes for incorporating explicit gender information [6] .

[2] Bojar et al. Findings of the 2018 Conference on Machine Translation, Table 8, `http://www.statmt.org/wmt18/results.html`

[3] `http://matrix.statmt.org/matrix/systems_list/1881`

[4] `https://github.com/pytorch/fairseq/blob/master/examples/translation/README.md`

[5] Prates et al. Assessing gender bias in machine translation: a case study with Google translate. Neural Computing and Applications, 2019

[6] Saunders, Sallis, Byrne: Neural Machine Translation Doesn't Translate Gender Coreference Right Unless You Make It, GEBNLP'20, `https://www.aclweb.org/anthology/2020.gebnlp-1.4/`

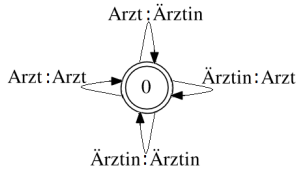
1.3 SGNMT - Syntactically Guided Neural Machine Translation

SGNMT is an open-source framework for neural machine translation (NMT) and other sequence prediction tasks. The tool provides a flexible platform which allows pairing NMT with various other models such as language models, length models, or bag2seq models. It supports rescoring both n-best lists and lattices. A wide variety of search strategies is available for complex decoding problems.

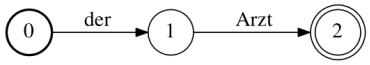
SGNMT is a decoding framework for neural machine translation which can also be used for general rescoring experiments. Visit `http://ucam-smt.github.io/sgnmt/html/` for detailed documentation to SGNMT.

The two central concepts in SGNMT are *predictors* and *decoders*. Predictors are scoring modules which can be combined to form the search space. For example, the **fst** predictor restricts the search space to a lattice, and the **fairseq** predictor can integrate a Fairseq NMT system. Decoders are search strategies for traversing the search space spanned by the predictors. The default decoder is beam search, but many other algorithms are supported. See `https://ucam-smt.github.io/sgnmt/html/command_line.html` .

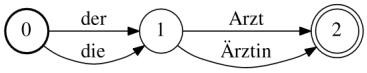
Later in this practical exercise you will build Weighted Finite State Acceptors containing syntactic gender alternatives to the words in the baseline hypotheses as described in Saunders ACL'20, and illustrated in this figure from that paper:



(a) A subset of flower transducer T. T maps vocabulary to itself as well as to differently-gendered inflections.



(b) Acceptor Y_B representing the biased first-pass translation $\mathbf{y_B}$ for source fragment 'the doctor'. The German hypothesis has the male form.



(c) Gender-inflected search space constructed from the biased hypothesis 'der Arzt'. Projection of the composition $Y_B \circ T$ contains paths with differently-gendered inflections of the original biased hypothesis. This lattice can now be rescored by a debiased model.

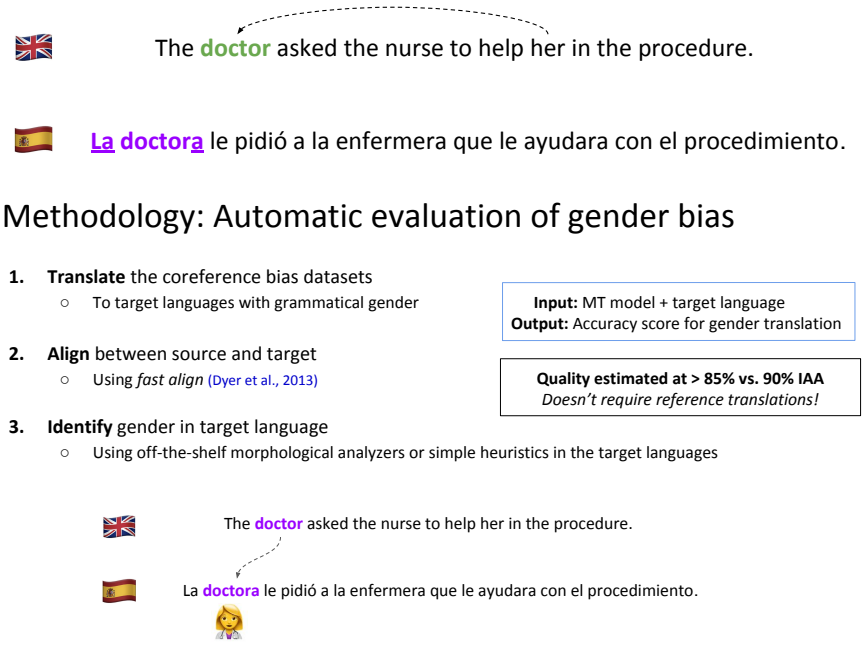
You will use SGNMT to search through these lattices with the gender-adapted NMT models to produce translation hypotheses with (hopefully) improved gender accuracy.

1.4 WinoMT Gender Accuracy

WinoMT [7] provides a suite of tools for measuring gender accuracy in translation. The core of the suite is a collection of English (source) sentences with known syntactic gender of prominent entities, e.g.

The **physician** told the nurse that **he** had been busy.
The **physician** told the nurse that **she** had been busy.

Physician should be translated as masculine for the first sentence and feminine for the second.

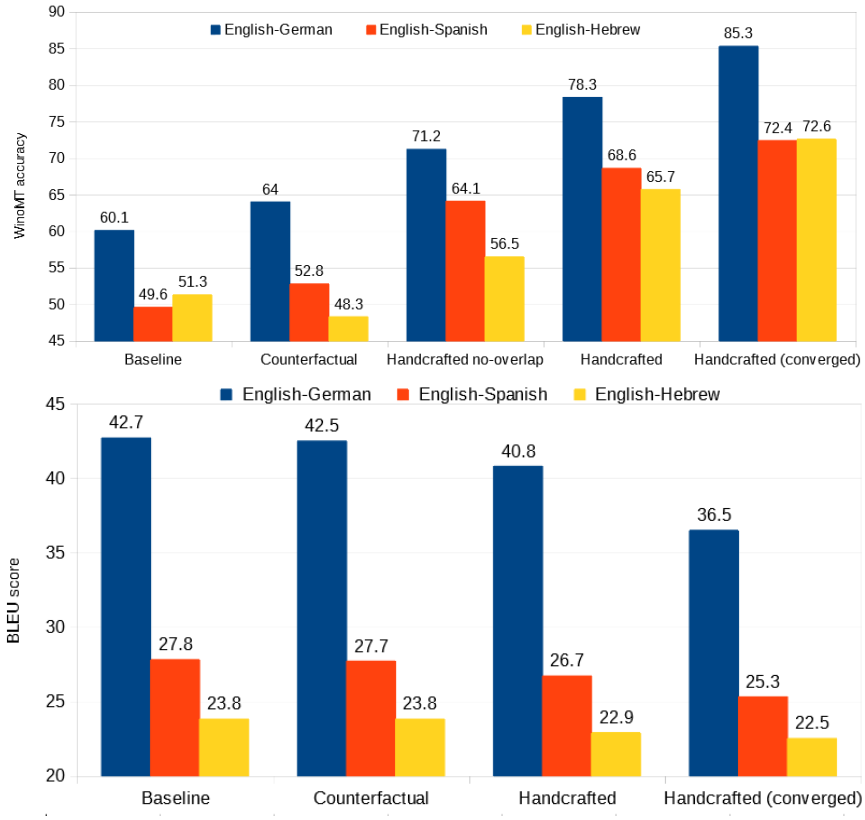


WinoMT provides a suite of tools for assessing the gender translation accuracy for systems. All that is required is for the system to supply translations of the English test set in one of the languages WinoMT supports. WinoMT then performs automatic alignment, tagging, and gender accuracy reporting.

[7] Stanovsky et al. *Evaluating Gender Bias in Machine Translation*, ACL'19 `https://arxiv.org/abs/1906.00591` , `https://github.com/gabrielStanovsky/mt_gender`

1.5 Catastrophic Forgetting

Saunders ACL'20 has shown that the simple adaptation scheme described in Section 1.2 can lead to *catastrophic forgetting*, as shown in the following figure [8]:



Fine-tuning baseline models on the English-German handcrafted set can improve WinoMT Gender accuracy from 60.1% to 85.3% (top), but in the process the system 'forgets' how to do general translation, with a BLEU score decrease from 42.7% to 36.5% (bottom).

The objective of this practical exercise is to investigate a decoding procedure that avoids catastrophic forgetting: the goal is to obtain improvement in WinoMT accuracy with little or no degradation in BLEU score relative to the baseline system.

[8] N.B. These results are not directly comparable to the systems you will study in this practical exercise.

1.6 Directories and Environments

You should activate the following environment for this exercise [9] :
`$ source /rds/project/rds-xyBFuSj0hm0/MLMI8.L2022/envs/README.MLMI8.1.activate`

Material required for this practical exercise is provided in the following directory:
`GDBNMTBDIR=/rds/project/rds-xyBFuSj0hm0/MLMI8.L2022/GDBNMT/`

Please familiarise yourself with these files and directories in `$GDBNMTBDIR/` :

- `configs/` – SGNMT configuration files
- `fairseq.pretrained/` – contains baseline translation hypotheses for the WinoMT and WMT'18 sets, as well as English and German wordmaps for the baseline systems.
- `fairseq.pretrained/adapt.0.01/` – contains models adapted as described in Section 1.2.
- `fsts/` – WinoMT and WMT'18 baseline translation hypotheses as WFSAs
- `winomt/` – WinoMT scoring scripts
- `slurm.decode, slurm.decode.mjobs.cpu` – SLURM decoding scripts
- `wordclasses` – gendered syntactic alternatives for German words

[9] Note that this environment does not support Jupyter notebooks.

2 Baseline Translations and Metrics

2.1 Byte Pair Encoded English Source Texts

The source language text for the WinoMT and WMT'18 translation sets are in Byte Pair Encoded form:

- `$GDBNMTBDIR/fairseq.pretrained/winomt.en-de.en.bpe`
- `$GDBNMTBDIR/fairseq.pretrained/wmt18.en-de.en.bpe`

The WMT'18 English source text looks like this:

```
$ head -2 $GDBNMTBDIR/fairseq.pretrained/wmt18.en-de.en.bpe
Munich 18@@ 56 : Four maps that will change your view of the city
A mental asylum , where today young people are said to meet .
```

Note that mapping from BPE's to words is done by simply deleting all all instances of '`@@ `', and detokenizing is done by the Python `sacremoses` tool. The following transforms the BPE encoded source text to its original plain text form:

```
$ head -2 $GDBNMTBDIR/fairseq.pretrained/wmt18.en-de.en.bpe | sed 's,@@ ,,g' |\
  sacremoses -l en detokenize
Munich 1856: Four maps that will change your view of the city
A mental asylum, where today young people are said to meet.
```

2.2 Baseline English-to-German Translations

Baseline translations are generated with the SGNMT decoder and the first component model of the Facebook-FAIR ensemble.

- WinoMT baseline translations:
  `$GDBNMTBDIR/fairseq.pretrained/winomt.sgnmt.wmt18ensemble.1/output.tok`
- WMT'18 baseline translations:
  `$GDBNMTBDIR/fairseq.pretrained/wmt18.sgnmt.wmt18ensemble.1/output.tok`

Note that the NMT decoders generate output in tokenized form, as follows:

```
$ head -2 $GDBNMTBDIR/fairseq.pretrained/wmt18.sgnmt.wmt18ensemble.1/output.tok
Munchen 1856 : Vier Karten , die den Blick auf die Stadt verndern
Ein psychisches Asyl , in dem sich heute Jugendliche treffen sollen .
```

which can be detokenized as follows:

```
$ head -2 $GDBNMTBDIR/fairseq.pretrained/wmt18.sgnmt.wmt18ensemble.1/output.tok |\
 sacremoses -l de detokenize
Mnchen 1856: Vier Karten, die den Blick auf die Stadt verndern
Ein psychisches Asyl, in dem sich heute Jugendliche treffen sollen.
```

2.3 WMT'18 English-German Baseline BLEU Score

The WMT'18 English-German test set is a standard test supported by the `sacrebleu` toolkit. The following command reports the BLEU score from the tokenized translations:

```
$ cat $GDBNMTBDIR/fairseq.pretrained/wmt18.sgnmt.wmt18ensemble.1/output.tok |\
 sacremoses -l de detokenize | sacrebleu --test-set wmt18 --language-pair en-de --format text
BLEU|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0 = 44.5 73.1/51.3/38.6/30.0 (BP =
    0.976 ratio = 0.976 hyp_len = 62750 ref_len = 64276)
```

In the above, `sacrebleu` reports a BLEU score of 44.5 with 1-/2-/3-/4-gram precisions of 73.1/51.3/38.6/30.0 and a Brevity Penalty of 0.976.

2.4 WinoMT English-German Baseline Gender Accuracy

The WinoMT scoring procedure is run as follows:

```
$ mkdir -p tmp/out/
$ cp $GDBNMTBDIR/fairseq.pretrained/winomt.sgnmt.wmt18ensemble.1/output.detok tmp/out
$ cd tmp/
$ $GDBNMTBDIR/winomt/winomt.sh out/output.detok
Scoring out/output.detok
....
acc = 78.3%; f1_male = 79.5% (p: 77.9/r: 81.2); f1_female = 82.8% (p: 81.1 / r: 84.6)
Gold distribution: male: 47.0% (1826), female: 46.86% (1822), neutral: 6.17% (0)
Predictions: male: 48.9%, female: 48.9%, neutral: 2.2%
```

WinoMT reports an overall accuracy of 78.3%, with male and female F1 entity scores of 79.5% and 82.8%.

2.5 German Baseline Translations – WFSAs

The baseline translations have been converted from text form to WFSAs for the `WMT'18` and `WinoMT` test sets. There is a WFSA for each translation in each set, in the files:

- `$GDBNMTBDIR/fsts/wmt18.sgnmt.wmt18ensemble.1/N.fst` for $N = 1, \ldots, 3888$
- `$GDBNMTBDIR/fsts/winomt.sgnmt.wmt18ensemble.1/N.fst` for $N = 1, \ldots, 2998$

Each WFSAs contains a single path corresponding to the tokenized baseline translation. Translations were encoded using this **wordmap**: `$GDBNMTBDIR/fsts/w+l.map.de` . Note that these correspond to the acceptors Y_B in Saunders ACL'20.

You can compare the first two WMT'18 translations:

```
$ head -2 $GDBNMTBDIR/fairseq.pretrained/wmt18.sgnmt.wmt18ensemble.1/output.tok
Munchen 1856 : Vier Karten , die den Blick auf die Stadt verndern
Ein psychisches Asyl , in dem sich heute Jugendliche treffen sollen .
```

to the contents of their corresponding WFSAs, as follows:

```
$ $GDBNMTBDIR/printstrings.py --fst $GDBNMTBDIR/fsts/wmt18.sgnmt.wmt18ensemble.1/1.fst \
    --syms $GDBNMTBDIR/fsts/w+l.map.de
<s> Munchen 1856 : Vier Karten , die den Blick auf die Stadt verandern </s>

$ $GDBNMTBDIR/printstrings.py --fst $GDBNMTBDIR/fsts/wmt18.sgnmt.wmt18ensemble.1/2.fst \
    --syms $GDBNMTBDIR/fsts/w+l.map.de
<s> Ein psychisches Asyl , in dem sich heute Jugendliche treffen sollen . </s>
```

3 Gender-Inflected Search Spaces

3.1 Introducing Inflections By Mapping Through Word Classes

A word-to-class mapping is provided for you in the file `$GDBNMTBDIR/wordclasses` . This is a two-column file with words in the first column and word classes in the second column. The file looks like:

```
Aachener Aachener.l.1
Aachenerin Aachener.l.1
Aachenern Aachenern.l.1
Aachener Aachenern.l.1
Aachenern Aachenern.l.2
Aachener Aachenern.l.2
Aachenerin Aachenern.l.2
Aachenern Aachenern.l.3
Aachenerin Aachenern.l.3
....
```

The word list should cover all the words in the WinoMT and WMT'18 test sets. Word classes are automatically derived [10] and are intended to contain differently gendered forms of the same word that are roughly similar parts of speech. For example, the class 'Aachener.l.1' contains the words 'Aachener', 'Aachenerin', 'Aachener', and 'Aachenern' referring to inhabitants of Aachen. The idea is that if any one of these words occurs in the baseline translation, all of the others should be considered as possible alternatives in a second-pass rescoring procedure intended to fix any first-pass syntactic gender errors. Note that words may belong to more than one class.

[10] See Section 3.3 of Saunders ACL'20 for a description on how classes were derived.

3.2 Exercise GDBNMT.1: Word-to-Class Transducers and Gender-Inflected Search Spaces

Make a working directory `fsts/` in which you will keep the transducers you build in this section.

3.2.1 Build a Word-to-Class Transducer

Using the word-to-class mapping file `$GDBNMTBDIR/wordclasses` and `$GDBNMTBDIR/fsts/w+l.map.de`, create a **flower transducer** `fsts/wtoc.fst` that :

1. maps every word to itself, and
2. maps every word to the wordclasses to which it belongs.

Take care to include self-mappings for the special symbols :

```
<epsilon> 0
<s> 1
</s> 2
<unk> 3
```

Applying `wtoc.fst` to the acceptors in the directories `$GDBNMTBDIR/fsts/wmt18.sgnmt.wmt18ensemble.1/` and `$GDBNMTBDIR/fsts/winomt.sgnmt.wmt18ensemble.1/` should yield transducers whose input language is the baseline translation and whose output language contains all possible class mappings, as in the following example:

```
$ fstcompose $GDBNMTBDIR/fsts/wmt18.sgnmt.wmt18ensemble.1/1.fst fsts/wtoc.fst > tmp.fst

$ $GDBNMTBDIR/printstrings.py --fst tmp.fst --syms $GDBNMTBDIR/fsts/w+l.map.de --n 2
<s> Munchen 1856 : Vier Karten , die den Blick auf die Stadt verandern </s>

$ $GDBNMTBDIR/printstrings.py --fst tmp.fst --syms $GDBNMTBDIR/fsts/w+l.map.de --n 2
    --project_output
<s> Munchen 1856 : Vize.l.1 Kaue.l.12 , diese.l.3 diese.l.3 Blick auf diese.l.3 Stade.l.4
    verndern </s>
<s> Munchen 1856 : Viertel.l.9 Kaue.l.12 , diese.l.3 diese.l.3 Blick auf diese.l.3 Stade.l.4
    verndern </s>
```

The example above shows the word 'Vier' in the baseline translation being replaced by the classes 'Vize.l.1' and 'Vize.l.12'. Only two strings form the output language of `tmp.fst` are printed - note that the complete output language is much, much larger than that.

3.2.2 Build a Gender Mapping Transducer

Using only your transducer `wtoc.fst`, make a transducer `fsts/T.fst` that maps all words in the baseline translations to their gendered alternatives. Applying `T.fst` via FST composition should generate a transducer whose target language contains all the alternative gendered versions of the input baseline translation, as follows:

```
$ fstcompose $GDBNMTBDIR/fsts/wmt18.sgnmt.wmt18ensemble.1/1.fst fsts/T.fst > tmp.fst

$GDBNMTBDIR/printstrings.py --fst tmp.fst --n 2 --syms $GDBNMTBDIR/fsts/w+l.map.de
<s> Mnchen 1856 : Vier Karten , die den Blick auf die Stadt verndern </s>

$GDBNMTBDIR/printstrings.py --fst tmp.fst --n 4 --syms $GDBNMTBDIR/fsts/w+l.map.de
    --project_output
<s> Munchen 1856 : Vize Katen , dieser diesen Blick auf dieser Stab verndern </s>
<s> Munchen 1856 : Vize Karteien , dieser diesen Blick auf dieser Stabes verndern </s>
<s> Munchen 1856 : Vikars Kauen , dieser diesen Blick auf die Stadt verndern </s>
<s> Munchen 1856 : Vize Kauen , dieser diesen Blick auf dieser Stabes verndern </s>
```

3.3 Exercise GDBNMT.2: Build a Word-to-BPE Transducer

A BPE dictionary is provided for you `$GDBNMTBDIR/fairseq.pretrained/word_bpe.dict`. This contains a mapping for all the words in the test sets to their subword (BPE) form. For example,

```
$ grep Aachen $GDBNMTBDIR/fairseq.pretrained/word_bpe.dict
Aachen A@@ a@@ chen
Aachener    A@@ ach@@ ener
Aachenerin  A@@ ach@@ en@@ erin
Aachenern   A@@ ach@@ en@@ ern
```

Build a flower transducer `fsts/wtobpe.fst` that maps word sequences to subword sequences.
Note you should use the wordmap `$GDBNMTBDIR/fsts/w+l.map.de` for the input (word level) encoding and the BPE map `$GDBNMTBDIR/fairseq.pretrained/wmap.bpe.de` for the output encoding. Applying `wtobpe.fst` and `T.fst`, followed by output projection, should yield a WFSA with gendered alternatives in their subword form:

```
$ fstcompose $GDBNMTBDIR/fsts/wmt18.sgnmt.wmt18ensemble.1/1.fst fsts/T.fst |\
  fstcompose - fsts/wtobpe.fst |fstproject --project_type=output > tmp.fst

$ $GDBNMTBDIR/printstrings.py --fst tmp.fst --n 2 \
  --syms $GDBNMTBDIR/fairseq.pretrained/wmap.bpe.de
<s> Munchen 18@@ 56 : Vier Kar@@ teien , diesen diesen Blick auf diesen Stau@@ es verandern </s>
<s> Munchen 18@@ 56 : Vier Karten , diesen diesen Blick auf diesen Stau@@ e verandern </s>
```

3.4 Remapping the Start Symbol for Fairseq/PyTorch

The Fairseq/PyTorch libraries assume that the index 0 indicates a start-of-sentence marker (the symbol <s>). This conflicts with OpenFST, which reserves 0 for the <epsilon> special symbol. The final step in preparing BPE lattices for rescoring with SGNMT is therefore to map the symbol '1' in the BPE lattice to '0'.

Note: that this mapping must be done for the Fairseq/PyTorch to assign the correct score to BPE sequences; but the mapping can only be done after ALL OpenFST optimisations are completed; otherwise OpenFST will treat the start-of-sentence symbol as an epsilon.

A special transducer `$GDBNMTBDIR/fsts/remapstartsym.fst` is provided for you to do this remapping of start symbols: it simply maps 1's to 0's, and all other indices to themselves. The effect can be seen by adding a final composition with this transducer to the above example; note that it is necessary to project on the output language following composition with `remapstartsym.fst`:

```
$ fstcompose $GDBNMTBDIR/fsts/wmt18.sgnmt.wmt18ensemble.1/1.fst fsts/T.fst |\
  fstcompose - fsts/wtobpe.fst |\
  fstcompose - $GDBNMTBDIR/fsts/remapstartsym.fst |\
  fstproject --project_type=output> tmp.fst

$GDBNMTBDIR/printstrings.py --fst tmp.fst --n 2 --syms $GDBNMTBDIR/fairseq.pretrained/wmap.bpe.de
Mnchen 18@@ 56 : Vier Kar@@ teien , diesen diesen Blick auf diesen Stau@@ es verndern </s>
Mnchen 18@@ 56 : Vier Karten , diesen diesen Blick auf diesen Stau@@ e verndern </s>
```

3.5 Exercise GDBNMT.3: Gender-Inflected WFSAs for Rescoring with Debiased Models

Using your transducers `T.fst` and `wtobpe.fst`, and the `remapstartsym.fst` transducer provided for you, create a set of gender-inflected WFSAs (as acceptors) for rescoring. You should do this for the WMT'18 and WinoMT sets. Consider whether to optimize your automata (via determinization and minimization) for rescoring (although make sure to apply `remapstartsym.fst` to your automata only after all OpenFST optimisations are completed).

Create your acceptors in your working directory with file format (here: `ga` means gendered-alternatives):
- `fsts/winomt.sgnmt.wmt18ensemble.1.ga/N.fst` for $N = 1, \ldots, 3888$
- `fsts/wmt18.sgnmt.wmt18ensemble.1.ga/N.fst` for $N = 1, \ldots, 2998$

These should be as developed in earlier sections, e.g. verify as follows:

```
$ $GDBNMTBDIR/printstrings.py --fst fsts/wmt18.sgnmt.wmt18ensemble.1.ga/1.fst \
   --syms $GDBNMTBDIR/fairseq.pretrained/wmap.bpe.de --n 2
Mnchen 18@@ 56 : Vier Kar@@ teien , diesen diesen Blick auf diesen Stau@@ es verndern </s>
Mnchen 18@@ 56 : Vier Karten , diesen diesen Blick auf diesen Stau@@ e verndern </s>
```

4 SGNMT Rescoring of Gender-Inflected Lattices with Debiased Models

The SGNMT decoder can be run as follows [11] (make the directory `tmp/`):

```
$ python $SGNMT/decode.py \
  --config_file=$GDBNMTBDIR/configs/wmt18.1.ende.wfsa.adapt.1.ini \
  --range=19:19 \
  --output_path=tmp/winomt \
  --src_test=$GDBNMTBDIR/fairseq.pretrained/winomt.en-de.en.bpe \
  --fst_path=fsts/winomt.sgnmt.wmt18ensemble.1.ga/%d.fst
2021-03-02 18:30:58,373 INFO: Setting up fairseq library...
2021-03-02 18:30:58,373 INFO: Done fairseq lib set up.
| [en] dictionary: 35662 types
| [de] dictionary: 35662 types
2021-03-02 18:30:58,450 INFO: Loading fairseq model(s) from
    /rds/project/rds-xyBFuSj0hm0/MLMI8.L2021/GDBNMT/fairseq.pretrained/adapt.0.01/checkpoint90.pt
2021-03-02 18:31:04,101 INFO: Initialized predictor fairseq (weight: 1.0)
2021-03-02 18:31:04,102 INFO: Initialized predictor fst (weight: 1.0)
2021-03-02 18:31:04,110 INFO: Start time: 1614709864.110095
2021-03-02 18:31:04,110 INFO: Next sentence (ID: 19): The physi@@ cian told the b@@ aker that
    she tried the best .
2021-03-02 18:31:06,144 DEBUG: Read fst from fsts/winomt.sgnmt.wmt18ensemble.1.ga/19.fst
2021-03-02 18:32:51,963 INFO: Decoded (ID: 19): Die rztin sagte der Bckerin , dass sie alles
    versucht habe .
2021-03-02 18:32:51,963 INFO: Stats (ID: 19): score=-0.434451 num_expansions=59 time=107.85
2021-03-02 18:32:51,963 INFO: Decoding finished. Time: 107.85
```

In the above,

- `config_file` : The config file `$GDBNMTBDIR/configs/wmt18.1.ende.wfsa.adapt.1.ini` provides a path to the debiased models along with a decoder constellation for decoding with Fairseq and FST predictors. The decoder will map BPE output sequences to tokenized words, in text form.
- `src_test` : the decoder should translate the BPE source text file for WinoMT
- `fst_path` : where the decoder can find WFSAs for each sentence to be translated
- `range` : specifies that the decoder should translate item 19 from the source file
- `output_path` : text file where tokenized translations are to be written

The decoder reports the Fairseq models and FSTs it will use as predictors, as well as the English BPE sentence to be translated:

> The physi@@ cian told the b@@ aker that she tried the best .

You can compare the output under the adapted model, `$ cat tmp/winomt` :

> Die Ärztin sagte der Bäckerin , dass sie alles versucht habe .

to the baseline translation (line 19 of `fairseq.pretrained/winomt.sgnmt.wmt18ensemble.1/output.tok`):

> Die Ärztin sagte dem Bäcker , dass sie alles versucht habe .

to see that in this instance that rescoring generates an alternative to the baseline translation.

[11] When running on a CPU you can ignore messages about missing NVIDIA drivers

4.1 SLURM Rescoring Scripts

A SLURM decoding script is provided for submitting SGNMT rescoring runs to the cluster.

- `$GDBNMTBDIR/slurm.decode.mjobs.cpu`

Decoding parallelises well over multiple CPUs and is often quicker than waiting for a GPU to become available. This script will translate the WinoMT and WMT18 sets, in succession, by constrained decoding of the automata you have built :

- Make a local copy of `$GDBNMTBDIR/slurm.decode.mjobs.cpu`
- Edit the entry `MLMI-[mycsid]-SL2-CPU` to replace `[mycsid]` with your CRSID
- Create a `logs/` directory (`mkdir -p logs/`)
- Run `sbatch slurm.decode.mjobs.cpu`
- Translations and logs are written to `wmt18.0.01.90.cpu/` and `logs/`

4.2 Exercise GDBNMT.4 WinoMT and BLEU Scores

Using the WFSAs you have constructed and the adapted models provided for you, perform rescoring of the WMT'18 and WinoMT test sets. Measure translation quality via BLEU scores on WMT'18 and gender accuracy on the WinoMT set.

5 Report Organization

Please organise your report into sections as follows:

- Overview - project objectives, description of modelling approaches and results expected
- Exercise GDBNMT.1
- Exercise GDBNMT.2
- Exercise GDBNMT.3
- Exercise GDBNMT.4
- Discussion and Conclusion

Points to address:

- Exercises 1, 2, 3: Explain how you construct and apply the WFSTs asked for in the practical. You can include pseudo-code and small examples of transducers (e.g. snippets of transducers), but please do not include extensive code listings beyond what you need to illustrate how you construct and use the transducers. For Exercise 1, give the FST operations used to create the transducer `T.fst` from your transducer `wtoc.fst`. For Exercise 3, describe any optimisation done to the lattices for rescoring.
- Exercise 4: Report the effect of constrained rescoring with adapted models in BLEU and WinoMT relative to the baseline system.
- Discussion and Conclusion: Summarise your findings. In your discussion you could:
  - Discuss whether WinoMT is a suitable metric for syntactic gender accuracy and comment on whether it has any inherent weaknesses or whether it relies on any hidden assumptions.
  - Discuss what would be required in terms of data and modelling procedures to extend the approach presented in this practical exercise to a new target language such as French or Spanish. Comment on whether you think this approach is feasible and what challenges it might face in real-world deployment.

There is no word count limit, but marking will consider both precision and recall.
You should aim for a presentation that is clear, concise, and correct.

Contributors
Danielle Saunders, Lent 2021
Bill Byrne, Lent 2022