This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644402.

Health in my Language

HimL

# D2.3: Final report on employing semantic role labelling and shallow proxies for negation and fidelity checking in MT

**Author(s):** Ondřej Bojar, Philip Williams, David Mareček, Martin Popel, Rudolf Rosa, Josef Jon, Michal Kašpar

**Dissemination Level:** Public

**Date:** January, 31st 2018

Version 1.0

| | |
|---|---|
| Grant agreement no. | 644402 |
| Project acronym | HimL |
| Project full title | Health in my Language |
| Funding Scheme | Innovation Action |
| Coordinator | Barry Haddow (UEDIN) |
| Start date, duration | 1 February 2015, 36 months |
| Distribution | Public |
| Contractual date of delivery | January, 31st 2018 |
| Actual date of delivery | January, 31st 2018 |
| Deliverable number | D2.3 |
| Deliverable title | Final report on employing semantic role labelling and shallow proxies for negation and fidelity checking in MT |
| Type | Report |
| Status and version | 1.0 |
| Number of pages | 24 |
| Contributing partners | CUNI, UEDIN |
| WP leader | CUNI |
| Task leader | CUNI |
| Authors | Ondřej Bojar, Philip Williams, David Mareček, Martin Popel, Rudolf Rosa, Josef Jon, Michal Kašpar |
| EC project officer | Tünde Turbucz |
| The Partners in HimL are: | The University of Edinburgh (UEDIN), United Kingdom |
| | Univerzita Karlova V Praze (CUNI), Czech Republic |
| | Ludwig-Maximilians-Universitaet Muenchen (LMU-MUENCHEN), Germany |
| | Lingea SRO (LINGEA), Czech Republic |
| | NHS 24 (Scotland) (NHS24), United Kingdom |
| | Cochrane (COCHRANE), United Kingdom |

For copies or reports, updates on project activities and other HimL-related information, contact:

Barry Haddow bhaddow@staffmail.ed.ac.uk
University of Edinburgh Phone: +44 (0) 131 651 3173

# Contents

# 1 Introduction

WP2 Semantically Motivated MT of the HimL project aims at improving the semantic correctness of machine translation. Of the many possible aspect of the meaning of sentences, we decided in the project proposal to focus on the following tasks:

**Task 2.1 Modelling Semantic Role Labelling in Machine Translation** aimed at preserving the key predicate-argument structure in the sentence (who did what to whom).

**Task 2.2 Enforcing Negation through Shallow Semantics** was devoted to analysis of errors in negation and to automatic detection of negation in text – to support correct handling of negation during training or translation.

**Task 2.3 Improving Core Fidelity of Shallow Models** attempted to prevent errors in phrase-based translation that are introduced by bad phrase table entries.

**Task 2.4: Employing High-Quality Large-Scale Dictionaries** experimented with the applicability of large manually-revised dictionaries of Lingea in MT.

This deliverable represents our final report on these tasks, describing our activities and the final results during the third year of the project. Activities of year 1 and 2 were already covered in the previous deliverables:

- D2.1 Initial report on semantics in MT (year 1),
- D2.2 Intermediate report on employing semantic role labelling and error checking in machine translation (year 2).

During the project lifetime, the field of machine translation has seen a major paradigm shift towards neural MT (NMT), an approach based on deep learning. In HimL, we responded accordingly and moved to the new state of the art. Some shift of topics was therefore inevitable. We decided to structure this deliverable according to the topics we actually addressed and we summarize this structure and its link to the original tasks here:

Section 2 focuses on the handling of syntax and semantic roles in NMT and thus falls under Task 2.1. While this task was not planned for year 3 according to the original description of work, we decided to continue with this work when we observed that, despite the big improvement in adequacy and even more in fluency, NMT still easily mishandles the structure of the sentence.

Section 3 concludes our efforts dedicated to handling negation. The first works on the respective Task 2.2, originally planned to start in year 2, were actually somewhat ahead of time and we reported on problems with negation in statistical MT and on methods for automatic negation detection already in D2.1. During year 2 of the project, the paradigm shift happened and we focussed on adopting the neural MT methods in HimL setting, putting Task 2.2 essentially on hold. With neural models employed, it was necessary to re-evaluate the extent of errors in negation and we bring this analysis here, in Section 3.

Section 4 can be seen as vaguely related to the topic of Task 2.3: Core Fidelity. Task 2.3 was geared towards shallow (i.e. non-syntactic) methods of classical (i.e. non-neural) statistical machine translation. As planned, Task 2.3 ended in year 2 and its results were described in D2.2. Here in Section 4, we experiment with a promising technique of so-called reconstruction: the neural model is trained to both translate and also reconstruct the original sentence from the translation. By doing so, the risk of accidentally dropping some part of the input, or otherwise seriously damaging its meaning is slightly reduced.

In Section 5, we report on the experiments with using hand-crafted dictionaries to avoid errors and improve translation quality in both classical statistical MT as well as neural MT, thereby concluding Task 2.4.

Section 6 then provides a detailed overview of our experiments with a very novel and promising model of NMT, so-called Transformer, on the HimL data. This model was proposed by Google in June 2017 and given the timeline for detailed evaluation in WP5 (see Deliverable 5.6), it was no longer possible to consider it for the post-editing evaluation and user survey. We nevertheless managed carry out the necessary experimenting in time for the manual ranking, where it performed generally very well and even won for Czech and Romanian. Since this Transformer setup was not our foreseen model for deployment, we did not include its description in D4.6. This is why we include all the details here.

The deliverable is concluded in Section 7.

# 2 Syntax and Semantic Role Labelling in NMT

In the previous deliverables D2.1 and D2.2, we described our experiments with syntactic models of MT (primarily D2.1) and with incorporating semantic roles of various kinds into both classical statistical MT and neural MT (primarily D2.2).

During year 3, we focused solely on neural MT and documented, that explicit syntactic annotation improves translation quality, as described in Section 2.1. We also continued our experiments with adding semantic role labels to neural MT, see Section 2.2.

## 2.1 Syntax-Aware Neural Machine Translation using CCG

Part of the appeal of neural models is that they can learn to implicitly model phenomena which underlie high quality output, and some syntax is indeed captured by these models. In a detailed analysis, Bentivogli *et al.* (2016) show that NMT significantly improves over phrase-based SMT, in particular with respect to morphology and word order, but that results can still be improved for longer sentences and complex syntactic phenomena such as prepositional phrase (PP) attachment. Another study by Shi *et al.* (2016) shows that the encoder layer of NMT partially learns syntactic information about the source language, however complex syntactic phenomena such as coordination or PP attachment are poorly modeled. These problems with the basic predicate argument structure of the sentence represent exactly the kinds of errors which the HimL project is trying to overcome.

Recent work which incorporates additional linguistic information in NMT models (Luong *et al.*, 2016; Sennrich and Haddow, 2016) shows that even though neural models have strong learning capabilities, explicit features can still improve translation quality. In this work, we report on a thorough investigation of rich syntactic features in NMT. We examined the benefit of adding syntactic information in the source, as an extra feature in the embedding layer following the approach of Sennrich and Haddow (2016). We also propose a method for generating syntactic information in the target: tightly coupling words and syntax by interleaving target syntactic representation with the word sequence. We compare this to loosely coupling words and syntax using multitask solutions, where the shared parts of the model are trained to produce either a target sequence of words or supertags in a similar fashion to Luong *et al.* (2016).

We use CCG syntactic categories (Steedman, 2000), also known as *supertags*, to represent syntax explicitly. Supertags provide global syntactic information locally at the lexical level. They encode subcategorization information, capturing short and long range dependencies and attachments, and also tense and morphological aspects of the word in a given context. Consider the sentence in Figure 1. This sentence contains two PP attachments and could lead to several disambiguation possibilities (*"in"* can attach to *"Netanyahu"* or *"receives"*, and *"of"* can attach to *"capital"*, *"Netanyahu"* or *"receives"*). These alternatives may lead to different translations in other languages. However the supertag S\NP/PP/NP of *"receives"* indicates that the preposition *"in"* attaches to the verb, and the supertag NP\NP/NP of *"of"* indicates that it attaches to *"capital"*, thereby resolving the ambiguity.

**Source-side**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| BPE: | Obama | receives | Net+ | an+ | yahu | in | the | capital | of | USA |
| IOB: | O | O | B | I | E | O | O | O | O | O |
| CCG: | NP | S\NP/PP/NP | NP | NP | NP | PP/NP | NP/N | N | NP\NP/NP | NP |

**Target-side**

NP Obama S\NP/PP/NP receives NP Net+ an+ yahu PP/NP in NP/N the N capital NP\NP/NP of NP USA

**Figure 1:** Source and target representation of syntactic information in syntax-aware NMT. The IOB annotation indicates whether a particular input token is a word on its own (O) or whether it is a subword from the beginning (B), middle (I) or end (E) of a word.

The conclusions of these experiments are as follows:

- We compare three novel approaches to integrating target syntax at word level in the decoder, by *serializing* CCG supertags in the target word sequence and by multitasking with either a shared or distinct attention model and decoder.
- We show that Syntax-aware NMT (SNMT) improves translation quality for English↔German, English↔Romanian as measured by BLEU.
- Our results suggest that a tight coupling of target words and syntax (by serializing) improves translation quality more than the decoupled signal in multitask training.

For further details please refer the to published paper (Nadejde *et al.*, 2017).

## 2.2 Neural MT with PDT Tectogrammatical Semantic Role Labels

In this section, we describe the experiments with neural MT which use semantic roles labels as defined the in the tectogrammatical layer, see the Prague Dependency Treebank (Hajič *et al.*, 2006). This work follows on the work presented in the previous Deliverable 2.2 in Section 1.3.

The tectogrammatical layer is available for both Czech and English, however we focus on the English side only.

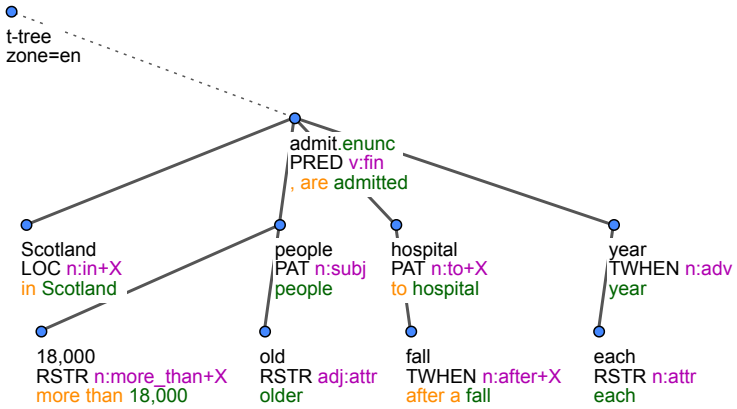


**Figure 2:** Tectogrammatical representation of sentence "In Scotland, more than 18,000 older people are admitted to hospital after a fall each year". The SRL factors we extract are *functors* (in black in the second line at each tree node) and formemes (in purple in the second line). The *valframe* values are not shown, however they are only identifiers pointing to the valency lexicon.

### 2.2.1 Data Selection and Annotation

For training, we used parallel data from UMC[1] and synthetic parallel data (synth).[2] Similarly as in previous year, the corpora were automatically parsed up to the deep syntactic representation (tectogrammatical layer of Prague Dependency Treebank) using the Treex toolkit.[3] One example of an automatically analyzed sentence is in Figure 2.

This year, we experimented with four possible types of labels that represent semantic roles in some way. All of them were automatically extracted from the tectogrammatical trees:

- *Functor* represents the semantic value of a deep-syntactic dependency relation; it expresses the function of an individual modifier in the sentence. Examples: ACT (Actor), PAT (Patient), ADDR (Addressee), LOC (Location), DIR3 (Direction to), TWHEN (Temporal), RSTR (Restrictive attribute), ...
- *Formeme* is a string representation of selected morpho-syntactic features of the content word and selected auxiliary words that belong to the content word. Examples: adj:attr (attributive adjective), n:obj (nominal object), n:after+X (noun with preposition *after*), v:fin (finite verb), ...
- *Valency Frame ID (relevant for semantic verbs only)* is a pointer to the valency lexicon, identifying the valency frame of the given verb. The pointers have forms like v-w10f1, v-w113f3, v-w202f22, ... and are not shared between verbs.
- *Verbal Valency Frame as a sequence of Functors (relevant for semantic verbs only)* is a formal description of the valency frame expressed as a list of individual functors that are obligatory in the particular frame. In contrast to Valency Frame ID, this representation will be identical for several verbs (e.g. synonyms or related verbs but also verbs with very different meaning but same syntactic pattern).

### 2.2.2 Experiments and Results

We used the Nematus[4] toolkit (a version of Nematus from June 2017) in its deep version, with batch size 80, decoder transition depth 8, encoder transition depth 4, dimension 1024, and dimension of words 500.

Semantic role labels were incorporated into the source sentences simply as additional tokens following the particular words. We performed four experiments, three of them used word forms with *Functor*, *Valency Frame ID*, and *Valency Frame as a sequence of Functors*, and their performance was compared to the baseline experiment using only word forms. Examples of the source and target sentences used in these four experiments are shown in Figure 3.

All the experiments were run for 90,000 iterations (or less if they stopped earlier). We have not performed any model ensembling, we simply took the model with the best BLEU score on the development data. The results are provided in Table 1.

[1] http://ufal.mff.cuni.cz/ufal_medical_corpus
[2] http://www-int.hink.eu/systems/nmt-training/corpora/synthetic
[3] http://ufal.mff.cuni.cz/treex
[4] https://github.com/rsennrich/nematus

| Setup | Example of Source and Target Sentence |
|---|---|
| form (baseline) | A total of 18 studies were identified and seven were included in the review .<br>Identifikovali jsme celkem 18 studií a sedm z nich zařadili do přezkumu . |
| form+functor | A total PAT of 18 RSTR studies RSTR were identified PRED and CONJ seven PAT were included PRED in the review LOC .<br>Identifikovali jsme celkem 18 studií a sedm z nich zařadili do přezkumu . |
| form+valfr.ID | A total of 18 studies were identified #w1639f2# and seven were included #w1676f1# in the review .<br>Identifikovali jsme celkem 18 studií a sedm z nich zařadili do přezkumu . |
| form+valframe | A total of 18 studies were identified [[ ACT PAT ]] and seven were included [[ ACT PAT DIR3 ]] in the review .<br>Identifikovali jsme celkem 18 studií a sedm z nich zařadili do přezkumu . |

**Figure 3:** Examples of source and target sentences used for training translation in individual setups. The target side is identical in all cases.

| Test Set | Form+Functor | Form+ValFrameID | Form+ValFrame | Form |
|---|---|---|---|---|
| dev. Cochrane+NHS24 | 27.40 | 29.26 | 29.11 | **29.72** |
| Y2 Cochrane+NHS24 | 25.60 | 27.15 | 27.57 | **28.28** |
| Y3 Cochrane | 26.63 | 26.71 | **28.04** | 25.84 |
| Y3 NHS24 | 20.56 | **20.94** | 20.30 | 20.35 |

**Table 1: Results of incorporating semantic role labels in source sentences and comparison with the baseline. The best result for each test set is in bold.**

#### 2.2.3 Discussion

From the results, it is not apparent whether the incorporation of semantic role labels into the source sentence helps translation quality. On the Y3 Cochrane data, all the three enriched setups showed better results than the baseline. However on the Y2 testing data (Cochrane together with NHS24) and even on the development data, all the setups performed worse than the baseline. On the Y3 NHS24 dataset, two of the three experiments were a little better than the baseline.

Since the improvement using the semantic role labels was questionable, we did not consider them for deployment.

# 3 Negation Errors in Machine Translation

In Task 2.2, we envisioned to tackle the well-known issue of negation in machine translation because in the state-of-the-art statistical systems (SMT) of that time, errors in translation of negation were both frequent and severe. However, the situation changed dramatically with the introduction of neural systems (NMT), with this problem practically vanishing. We review this problem, present an analysis of negation translation errors in SMT and NMT, and conclude with our eventual decision to abandon this problem as an already obsolete one.

## 3.1 Background

Negation used to be an important and difficult problem in SMT, with a source negation frequently found missing in the translation; or, less frequently, the translation would contain a surplus or misplaced negation.

It is crucial to translate negation correctly, as it strongly influences meaning – usually the meaning of the translation is inversed when negation is not translated correctly. The medical domain does not make this problem any less severe: it is of utmost importance whether e.g. a certain person *must* or *must not* take a specific medication or undergo a specific procedure, or whether certain symptoms *are* or *are not* the signs of a specific disease. Furthermore, a negation error is not easy to spot without access to the source sentence, as most sentences can plausibly exist both in their positive and their negative form – for example, a patient may be advised to take exercise under some conditions, but not to take exercise under other conditions.[5]

For SMT, the problem of negation is especially hard to handle if the source and target language use different means of expressing it in the same situation. In English, as well as in the other HimL languages, we can find negation prefixes (such as "un-" or "in-"), negation suffixes ("-less"), negation particles ("no", "not"), lexical negation ("bad" instead of "ungood"), indirect lexical negation ("not to do X" – "fail to do X"), and combinations thereof. Moreover, the languages differ in the number of negation markers used to express a negation (single negative versus double negative). This great variance in the negation markers leads

[5] Although there are, of course, cases, where the missing negation error is rather clear; e.g. doctors rarely advise patients to drink alcohol or smoke.

to severe inconsistencies already in word alignment, causing incorrect phrase extraction and subsequent generation of incorrect translation options. Furthermore, as often both the positive and the negative sentence is plausible, the language model is often unable to prefer the correct translation to the incorrect one. And, to make matters even worse, automatic evaluation metrics, such as BLEU, which are used both in training and evaluation of SMT systems, do not assign a high penalty for mistranslating negation, as to them, this is just an error in one tiny word; a missing negation typically is an error comparable e.g. to a missing determiner in the automatic scores. Thus, for an SMT system, it is very hard to translate negation correctly, as it is deficient both in the means and in the motivation to do so.

On the other hand, great improvements in the translation of negation have been achieved with source-informed post-editing approaches. In particular, the Depfix system of Rosa *et al.* (2012), which we intended to build upon in this project, has been repeatedly shown to be efficient in correcting negation errors in English-to-Czech SMT with an accuracy of approximately 90% (Bojar *et al.*, 2013), leading to a significant improvement in MT quality. Moreover, Depfix uses a rather simple solution: if negation is present in source but not present in translation, it assumes that the negation was lost in translation, and reinserts it by following alignment links from the negated part of the source (typically a verb) and negating its counterpart (typically adding a negative prefix to a target verb). We thus originally envisioned fitting Depfix to HimL by adapting it for the other HimL languages and tuning it to perform well on HimL data.

## 3.2 Current Situation

In recent years, SMT systems are being overcome by NMT systems as the new state-of-the-art, which are able to produce translations of a significantly higher quality.

NMT systems work along a different paradigm. Importantly, they do not rely on word-to-word alignment, but rather use a soft attention mechanism, which makes it easier for them to handle hard-to-align phenomena, such as negation. Moreover, the attention mechanism is trained together with the whole NMT system in a supervised way, thus fitting it not only to the task of machine translation, but even to the particular language pair and dataset. This is much more principled than the independent unsupervised SMT word-aligners and symmetrization heuristics.

Another (and possibly the most important) strength of neural systems is the way they represent input words. Apart from word clustering, which is rather rough, pre-neural systems represent each word as an independent atomic unit, and are mostly unable to generalize over individual word identities. In neural systems, continuous-space vector representations (word embeddings) are used instead, which not only can represent the similarity of close words in a fine-grained real-valued multidimensional way, but they are often able to capture even regular relations between pairs of less-similar words. While the most famous examples of relations captured are e.g. female-variant-of (king – queen, actor – actress) or capital-of (France – Paris, Germany – Berlin), grammatical relations can also be captured, such as plural-of (dog – dogs, mouse – mice) or, importantly for us, opposite-of (healthy – unhealthy, carefull – careless, new – old). As the vector representations are also typically trained together with the translation system, they can be automatically tailored to capture phenomena that are relevant for the translation task.

And finally, as a third crucial improvement, NMT systems tyically go beyond the word boundaries, splitting the input into subwords. While this is usually done independently of the MT system in an unsupervised manner, the general idea is to split words into more frequent subwords, and it can thus be expected that a negation prefix or suffix will usually get split off the rest of the word. This makes it even easier for the NMT system to handle negation properly.

It has been observed that many phenomena which used to be hard for SMT systems, including negation, are considerably easier to handle for the NMT systems. In particular, when starting preliminary experiments focused on correcting negation errors in NMT outputs, we noticed that negation errors seem to have become practically non-existent.

## 3.3 Error Analysis

To back our observations with hard data, we performed an error analysis of NMT negation-related errors. This section presents the results for three of the four HimL target languages: Czech, German and Polish.

### 3.3.1 English-to-Czech

We selected one language pair, English-to-Czech, to perform a detailed error annotation. An annotator annotated 298 English sentences and their Czech translations, marking whether the source or target sentence contains a negation, whether the meaning of the source sentence was translated correctly in such cases, and, if not, whether the translation error ocurred directly in the transfer of the negation or elsewhere.

| System | Neural Machine Translation | | | Statistical Machine Translation | | |
|---|---|---|---|---|---|---|
| | sentences | % of all | % of neg | sentences | % of all | % of neg |
| Annotated sentences | 298 | 100% | | 298 | 100% | |
| No negation present | 237 | 79.5% | | 241 | 80.9% | |
| Negation present | 61 | 20.5% | 100.0% | 57 | 19.1% | 100.0% |
| Translation correct | 55 | 18.5% | 90.2% | 41 | 13.8% | 71.9% |
| Translation incorrect | 6 | 2.0% | 9.8% | 16 | 5.4% | 28.1% |
| Error in negation | 2 | 0.7% | 3.3% | 12 | 4.0% | 21.1% |
| Other error | 4 | 1.3% | 6.6% | 4 | 1.3% | 7.0% |

**Table 2: Analysis of errors in Neural and Statistical Machine Translation.**

The error analysis was performed on HUME round 2 test set, on English-to-Czech translation, on NMT and Chimera systems. The output of the evaluation is stored in the project Subversion repository.[6]

Quantitatively, the evaluation results are summarized in Table 2. The data show that with the SMT system, nearly 30% of sentences containing a negation are translated incorrectly, while for the NMT system, this drops to only 10%. Furthermore, there is a certain proportion of sentences (approximately 7%) where the translation error is not in the translation of negation itself (although the error may be influenced by the presence of the negation in the source sentence). Thus only for 3% of sentences that contain negation, the NMT system clearly makes an error in transferring the negation, compared to 20% for the SMT system. As approximately 20% of sentences contain negation, less than 1% of all sentences produced by the NMT system contain a negation error, compared to 4% for the SMT system. In total, there were only 2 clear cases of an error in negation translation in the NMT system: 1 missing negation, and 1 incorrect negation scope (due to subject-object marking error).

Qualitatively, we saw that in many cases, difficult negation phenomena are translated correctly by the NMT system, and more complex constructions can be seen in the output of NMT compared to SMT. In Czech double negation, which is asymmetric to English negation, there was not a single error. Moreover, the NMT system also perfectly handled many cases of asymmetric/lexical negations, which are rather hard to handle correctly, and constituted a major problem for the simple correction rules in the Depfix system. We list a few examples of source phrases and NMT-produced translations (all correct), marking the regular negation markers in **bold** – it can be seen that negation is always explicitly regularly marked only in one of the languages, while the other language uses more lexical means to express the negative meaning:

- was slurring = mluvila **ne**srozumitelně (“was speaking unintelligibly”)
- recently = **ne**dávno (“not long ago”)
- enemies = **ne**přátelé (“non-friends”)
- failing to coordinate = **ne**koordinovala (“not coordinating”)
- **un**fortunately = bohužel

On the other hand, the SMT system demonstrated a range of classical well-known negation errors, especially:

- missing negation
- misplaced negation (incorrect negation scope)
- missing the whole negated verb
- incorrectly formed Czech double negation

The observed errors in SMT output are often fixable by simpler measures than for NMT; even simple rule-based solutions could be able to get rid of at least half of the cases (this has already been shown in Depfix). At the same time, the SMT system does not often use complex asymmetric negation translations that could deceive such a rule-based post-editing system to produce false positives.

### 3.3.2 English-to-German and English-to-Polish

Subsequently, we also investigated other language pairs, realizing that the situation is similar in them, and thus not performing such a detailed analysis for them. We present rough numbers from these brief analyses in Table 3, showing that incorrect translations of negation practically vanished with the introduction of neural MT.

[6] http://svn.statmt.org/repository/himl/data/hume-round2-neg-err-anot; the “header” file explains the annotation; all errors found have a textual comment.

| System | en-de PBMT | | en-de NMT | | en-pl PBMT | | en-pl NMT | |
|---|---|---|---|---|---|---|---|---|
| | abs | rel | abs | rel | abs | rel | abs | rel |
| Evaluated sentences with negation | 36 | 100% | 36 | 100% | 33 | 100% | 33 | 100% |
| Negation translation seems correct | 32 | 89% | 36 | 100% | 31 | 94% | 33 | 100% |
| Negation translation seems incorrect | 4 | 11% | 0 | 0% | 2 | 6% | 0 | 0% |

**Table 3: Rough analysis of errors in Neural Machine Translation (NMT) and Phrase-based Machine Translation (PBMT) for English-to-German (en-de) and English-to-Polish (en-pl).**

| | Fluency | | Adequacy | |
|---|---|---|---|---|
| | PBMT | NMT | PBMT | NMT |
| cs-en | 69.3 | 78.7 | 72.6 | 75.4 |
| de-en | 68.6 | 77.5 | 70.9 | 75.8 |
| ro-en | 65.6 | 71.9 | 71.0 | 71.2 |

**Table 4: Fluency and adequacy scores for the best-performing phrase-based (PBMT) and neural (NMT) systems at WMT16. The scores are average human evalation scores on a 0-100 scale where 100 represents perfect fluency / adequacy.**

### 3.4 Conclusion on Negation

The classical SMT systems are prone to frequently translate negation incorrectly, with many of the errors being fixable by simple means, such as a rule-based post-editing system (Depfix). In NMT, errors in translation of negation are present in less than 1% of all sentences, and so can be considered to be a rare problem, not deserving special treatment.

Furthermore, the correct negation translation cases clearly demonstrate that the NMT system is capable of handling complex negation constructions with a very high accuracy, without a clear systematic deficiency in treating any specific negation phenomenon. Thus, negation-related issues seem to have become marginal. Furthermore, the rare remaining issues would be very hard to tackle – as the modern NMT system does not feature the old-style errors common for SMT systems, we believe that any negation-handling system less complex than the NMT system itself is very likely to introduce more false-positives than true-positives, especially in the cases of assymetric/lexical negation.

To conclude, based on our error analysis, we do not find it constructive any more to focus specifically on fixing negation errors, at least at this stage, when there is a range of other errors in NMT that are both more frequent and more severe, and also potentially easier to handle than the intricacies of complex negation structures.

## 4 NMT with Reconstruction

The shift from statistical to neural machine translation models has seen impressive gains in translation quality and a significant reduction in some specific error types, such as negation handling. However, improvements in adequacy have generally not kept pace with improvements in fluency. Of the recent empirical studies that have been conducted, the largest scale comparison of fluency and adequacy we are aware of was at WMT16.[7] There human evaluators assessed fluency and adequacy of output from systems submitted to the shared news translation task. Table 4 gives the results for the best-performing phrase-based and neural systems for the HimL language pairs (although note that due to the greater availability of English-speaking evaluators, the systems evaluated were for the opposite into-English direction).

As a result of the increased fluency/adequacy gap, the problem of systems producing seemingly-good translations that read well but misrepresent or omit parts of the source has actually worsened. At the individual sentence level, this tendency is often visible in under-translation and over-translation, where key words of phrases are either skipped or repeatedly translated, often without affecting the fluency of the sentence. Figure 4 gives an example of under-translation drawn from the Cochrane test set when translated with a baseline NMT system. The translation has omitted an important detail, but unless the reader has access to the source, it is not obvious that anything is missing. In phrase-based systems this type of problem was less common due to the search algorithm's coverage mechanism (instead problems arose from the lack of context during phrase pair selection or from low-quality phrase table entries).

Several solutions have been proposed in the literature, including models that add a coverage-style component. We chose to reimplement the encoder-decoder-reconstructor model (Tu *et al.*, 2017), which is a comparatively straightforward extension of the attentional encoder-decoder we are already using. The authors convincingly demonstrate improvements in overall translation quality (as measured by BLEU) and that the changes to the model address adequacy in practice (according to human evaluation).

[7] http://www.statmt.org/wmt16

| | |
|---|---|
| **Source** | Dieser Zustand erhöht **vier bis fünf Mal** das Risiko, dass eine transitorische ischämische Attacke (TIA) oder Schlaganfall vorkommt. |
| **Reference** | This condition increases your risk **by about four to five times** of having a transient ischaemic attack (TIA) or stroke. |
| **Translation** | This condition increases the risk of transient ischaemic attack (TIA) or stroke. |
| **Source** | eine ausgewogene Ernährung und Einschränkung des Alkoholkonsums |
| **Reference** | eating a balanced diet and limiting how much alcohol you drink |
| **Translation** | Nutrition, nutrition, nutrition, nutrition, nutrition, nutrition, nutrition, nutrition, nutrition, nutrition, ... |

**Figure 4:** Two examples of inadequate translations from German-to-English systems trained using the HimL data. The first, from the Cochrane test set, is an example of under-translation (the untranslated text is indicated in boldface). The second is from the NHS24 test set. Note that the translation is truncated: the decoder actually repeats the word 'nutrition' 100 times, stopping only when its length limit kicks in.

We also experimented with a closely-related, but simpler, approach in which we train a model in the inverse translation direction and use its scores for reranking.

## 4.1 Model

The encoder-decoder-reconstructor model adds a 'reconstructor' component to the now-standard attentional encoder-decoder model (Bahdanau *et al.*, 2014). The reconstructor is a recurrent neural network with a similar form to the decoder, with the main differences being that its attention mechanism operates over the decoder's hidden states (instead of the encoder's) and that it outputs a softmax distribution over sequences of source words instead of target words. The basic idea is similar to that of an auto-encoder: the reconstructor's job is to reproduce the source sentence. It does this, in part, by ensuring that the decoder accumulates information about all parts of the source sentence.

During training, the model produces a reconstruction score, analagous to the standard translation likelihood score, and the optimization algorithm is given the objective of maximizing the sum of the two scores. In principle, a higher reconstruction score should correspond to a more adequate translation. During decoding, the reconstruction scores can be used for reranking: the decoder first produces a $n$-best list containing likelihood scores. The corresponding decoder state sequences are given as input to the reconstructor, which generates reconstruction scores. The translations are then reranked according to a weighted sum of the two scores. For full details, we refer the reader to the original work (Tu *et al.*, 2017).

The inverse model is much simpler: in addition to the baseline attentional encoder-decoder model, we train an identically configured model with the source and target data switched. We use the resulting system to add model scores to the $n$-best list of the baseline, which we then rerank in the same way as for the reconstruction model (i.e. according to a weighted sum of the forward and inverse model scores).

## 4.2 Experiments

We reimplemented the reconstructor in the Nematus toolkit, adding support for training, stochastic generation of reconstructed source sentences, and reranking of $n$-best output. We conducted small-scale preliminary reconstruction experiments using 4 million sentence pairs each of WMT16 Latvian→English data and HimL English→German data. We then performed larger-scale experiments using the HimL systems that were submitted to the Biomedical task at WMT17. For the larger-scale experiments, we also trained inverse models. The larger systems were evaluated as part of the final human evaluation (see D5.6 for results).

In all reconstruction experiments, we first trained a standard encoder-decoder model, stopping on convergence of the likelihood score on a held-out validation set. We then resumed training with the full reconstructor-enabled model (this is possible since the full model network is simply an extension of the original). During training Nematus saves the model parameters after a fixed number of iterations. We continued training of the model using the parameters from the point at which the BLEU score peaked on the validation set.

### 4.2.1 Small-Scale Experiments

In order to satisfy ourselves that the reconstructor was working as intended, we first used it to stochastically generate sentences in the source language and then used BLEU to compare the reconstructed sentences with the true input sentences (in this mode, the model is functioning as an auto-encoder). Figure 5 shows how the BLEU score improves during training. After sufficient training, the reconstructor is able to do a reasonably good job of reconstructing the source sentence, often with only minor, semantically-plausible changes. In other cases, the changes are less interpretable, though typically errors are localized to specific parts of the sentence. See Figure 6 for some representative examples.

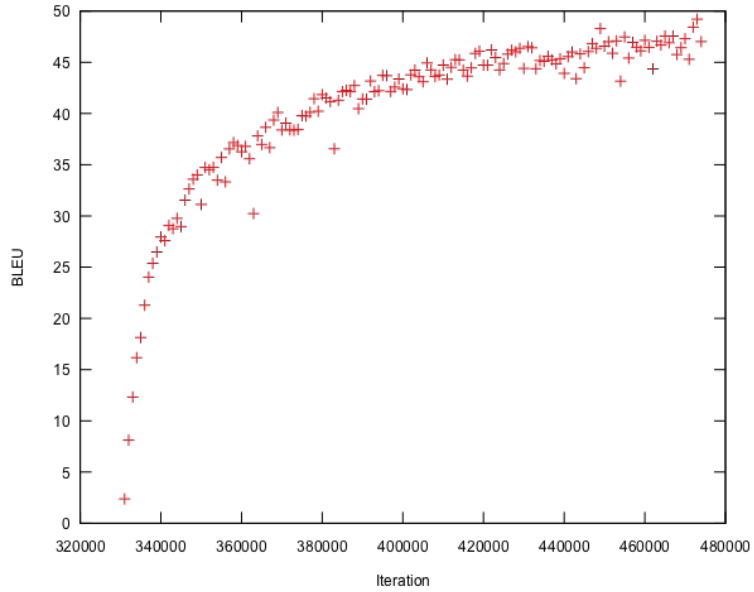


**Figure 5:** BLEU scores of reconstructed sentences (compared to original input sentences) as training progresses. In this experiment, the reconstructor was enabled at iteration 330,000.

| | |
|---|---|
| **Source** | Founded in 2006, Madara Cosmetics has several shareholders, mostly Latvian **individuals**. |
| **Reconstructed** | Founded in 2006, Madara Cosmetics has several shareholders, mostly Latvian **people**. |
| **Source** | Latvia's economic growth is **planned** at 3.5 percent in 2017 and 3.4 percent in 2018 and 2019. |
| **Reconstructed** | Latvia's economic growth is **scheduled** at 3.5 percent in 2017 and 3.4 percent in 2018 and 2019. |
| **Source** | 'Norwegians **like** to visit restaurants as well as SPA services,' she said. |
| **Reconstructed** | 'Norwegi**Spanish kind** to visit restaurants as well as SPA services,' she said . |

**Figure 6:** Examples of reconstructed source sentences.

| **lcost** | **rcost** | **Translation** | **New Rank** |
|---|---|---|---|
| 4.85 | 2.20 | this condition increases the risk of transient ischaemic attack ( TIA ) or stroke . | 7 |
| 4.93 | 2.02 | this condition increases the risk of transient ischaemic attacks ( TIA ) or stroke . | 6 |
| 5.36 | 2.28 | this condition increases the risk of a transient ischaemic attack ( TIA ) or stroke . | 9 |
| 6.67 | 0.44 | this condition increases **four to five times** the risk that transient ischemic attack ( TIA ) or stroke occurs | 1 |
| 5.13 | 2.22 | this condition increases the risk of transient ischemic attack ( TIA ) or stroke . | 10 |
| 6.95 | 0.44 | this situation increases **four to five times** the risk that transient ischemic attack ( TIA ) or stroke occurs | 2 |

**Table 5: Translation costs (lcost) and reconstruction costs (rcost) for the first six candidates from the 12-best output. The ordering reflects the ranking produced by the baseline system. The final column indicates the order of the sentences if reranked according to the weighted combination of lcost and rcost.**

| | en→ro | | | en→pl | | |
|---|---|---|---|---|---|---|
| | 12 | 50 | 100 | 12 | 50 | 100 |
| Cochrane baseline | 41.1 | 40.9 | 40.5 | 28.5 | **28.7** | 28.5 |
| +rerank | 41.0 | **41.5** | **41.5** | 28.6 | 28.6 | 28.0 |
| NHS24 baseline | 29.2 | 28.9 | 26.7 | 22.4 | 22.7 | 22.6 |
| +rerank | 29.3 | 29.6 | **29.8** | 22.5 | 22.7 | **22.9** |

**Table 6: BLEU scores for the large-scale English→Romanian and English→Polish reconstructor experiments using various $n$-best sizes ($n = 12, 50, 100$). The best result for each test set is indicated in boldface.**

| | en→ro | | en→pl | |
|---|---|---|---|---|
| | Cochrane | NHS24 | Cochrane | NHS24 |
| baseline | 40.9 | 28.9 | **28.7** | **22.7** |
| reconstructor | 41.5 | 29.6 | 28.6 | 22.7 |
| inverse | **41.8** | **30.1** | 28.6 | 22.4 |

**Table 7: BLEU scores for the large-scale English→Romanian and English→Polish inverse model experiments using a $n$-best size of 50. The best result for each test set is indicated in boldface.**

We evaluated translation quality in terms of BLEU score. For English→German, continued training with the reconstructor produced a poorer score compared to the baseline peak (32.14 BLEU, down from 32.63 BLEU), although this was better than the baseline after training for the same total number of iterations (30.50 BLEU). Results for the NHS test set and for Latvian→English followed a similar pattern, with the decreasing BLEU likely being due to the model overfitting the small data set. Inspection of the reconstruction scores for the $n$-best output indicated that increased reconstruction costs often appeared to be indicative of translation adequacy, with under-translated sentences often having high costs. Table 5 gives the translation and reconstruction costs for candidate translations from our earlier under-translation example. We obtained further improvements from reranking the $n$-best output, although the final results were only marginally better than the baseline (32.75 BLEU). However, this did raise the question of whether we could do better by reranking the output of the baseline system rather than the extended system. We explored both options in our larger-scale experiments.

### 4.2.2 Large-scale Experiments

In our large-scale experiments we started from the English→Romanian and English→Polish models that were used for UEDIN's submission to the WMT17 biomedical translation task.[8] We continued training and then used the reconstructor to score the output of the $n$-best lists from the original systems (we found that this performed slightly better than using $n$-best lists from the extended systems). For reranking we used the scoring function,

$$\text{score}(y \mid x) = \frac{logP(y \mid x; \theta)}{|y|^{\alpha}} + \lambda \frac{logR(x \mid s; \gamma)}{|x|^{\alpha}},$$

where $x$ and $y$ are the source and target sentence, respectively; $s$ represents the decoder state sequence for $y$; $\theta$ represents the model parameters for the baseline model; $\gamma$ represents the parameters for the reconstructor; and $\lambda$ and $\gamma$ are the weighting and sentence normalization hyperparameters, which were tuned on the validation set.

Table 6 gives the results for various different $n$-best list sizes. The clearest results are for English→Romanian. Note that baseline translation quality degrades with increasing beam width, but improves when the reranker is added (and surpasses the best baseline results). Further analysis revealed that the reranker is correcting a tendency of the baseline system to produce increasingly long sentences as the beam width increases.[9] For English→Polish, the results were more mixed, with the reranker having a much smaller effect on translation quality.

Table 7 gives the results for the inverse model using a beam size of 50. In this case, the inverse model is effective for English→Romanian, actually outperforming the more complex reconstructor model. However, it has little effect for English→Polish.

## 4.3 Conclusion on NMT with Reconstruction

Despite the improvement in overall translation quality that has come from the shift to neural models, adequacy remains a problem for machine translation. The reconstructor and inverse models aim to improve translation by reducing the tendency of the model

[8] http://statmt.org/wmt17/biomedical-translation-task.html

[9] At a beam width of 12, the average baseline translation length is 96.9% of the reference length. This increases to 100.4 at a beam width of 50, then 108.8% at 100. When the re-ranker is used, the average translation length is much more stable: 96.3%, 96.0%, and 96.5% for widths of 12, 50, and 100.

to omit parts of the source sentence or to deviate from the meaning.

For English→Romanian, both models led to improvements in translation quality as measured by BLEU. For English→Polish, the results were less clear. Both the reconstructor and inverse systems were included in the year three evaluation (described in Deliverable D5.6). For English→Romanian, the reconstruction model was the third best according to human evaluation, beating the inverse model (4th) and WMT17 model (6th), which is closest to the baseline setup. For English→Polish, the reconstruction model was tied for first place with WMT17 model. The inverse model was 4th.

# 5 Employing Dictionaries

As part of their business, Lingea has been developing dictionaries for over 20 years. The aim of this development and long-term curation of dictionary data is to provide content usable for printed book dictionaries as well as for various automatic language processing tools, such as morphological analyzers. In this section, we investigate how to make use of the available dictionary data to improve both SMT and NMT.

The common setup of the experiments is described in Section 5.1.

The experiments[10] go in three directions. One of them is to filter training data according to available dictionary data, and reject sentences that seem to not match well, see Section 5.2. The second one is to enrich existing training corpora with existing dictionaries, see Section 5.3. The third one is to rerank translation results based on their dictionary adherence, see Section 5.4

## 5.1 Common Settings

### 5.1.1 Datasets

We trained several baseline en-cs models until convergence on our general corpus, comprising of roughly 24M lines from the common WMT-like sources: news, Europarl, CzEng, etc. We tried several mixing and filtering techniques to create an in-domain training set.

For experiments with medical data, the HimL development set served for MERT in phrase-based MT and for validation, early stopping and choosing the best performing model during the training of NMT models. For experiments within the travel domain, we used excerpts of our translated travel guides as training, development and test data.

We tokenized all the data with standard Moses tokenizer and truecased them using our own truecaser based on morphology.

### 5.1.2 NMT Models

For all our NMT experiments, we used the Marian toolkit.[11] We call these models *model 62*, *model 72*, and *model 76*. The hyperparameters used for these models can be found in Table 8 (models 72 and 76 have the same hyperparameters, the only difference between them was training on different splits of training data during previous experiments).

In all NMT setups, we segmented the input tokens into 90000 subwords, selected with BPE from concatenation of both source and target corpora.[12]

We measured BLEU on HimL dev and test sets each 20000 minibatches, with dynamic batching, adjusting the size of mini-batches so that they fit in the GPU memory. Scores reported in the following sections are BLEU scores on the HimL test set, achieved by models with best performance on HimL dev set. We used early stopping during the training, with patience of 10.

### 5.1.3 Moses Setup for SMT

For comparison with pre-neural statistical MT (SMT), we also trained number of phrase-based models with in-domain data only.

### 5.1.4 Evaluation Details

Scores in medical experiments were computed on tokenized outputs with multi-bleu.perl.[13] For travel guides evaluation, we switched to detokenized output and SacréBLEU[14] script to produce the scores.

[10] GPU time was supported by Microsoft's donation of Azure credits to The Alan Turing Institute. This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

[11] https://github.com/marian-nmt/marian

[12] https://github.com/rsennrich/subword-nmt

[13] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[14] https://github.com/awslabs/sockeye/tree/master/contrib/sacrebleu

| Parameter | Models 62 and 72 | Model 76 |
|---|---|---|
| dropout_hidden | 0.1 | 0.1 |
| decoder | gru_cond | gru_cond |
| dropout_source | 0.1 | 0.1 |
| dec_base_recurrence_transition_depth | 3 | 3 |
| enc_depth_bidirectional | 1 | 1 |
| lrate_decay | 0.9 | 0.9 |
| dropout_target | 0.1 | 0.1 |
| dec_depth | 1 | 2 |
| decoder_deep | gru | gru |
| optimizer | adam | adam |
| encoder | gru | gru |
| dec_high_recurrence_transition_depth | 1 | 1 |
| lrate | 0.001 | 0.001 |
| dim | 448 | 768 |
| enc_recurrence_transition_depth | 1 | 3 |
| dim_word | 384 | 128 |
| enc_depth | 1 | 1 |

**Table 8: NMT models hyperparameters in Nematus format**

## 5.2 Domain Adaptation and Corpora Filtering

This section describes our experiments with filtering training corpora using our Czech-English dictionaries to improve translation quality of NMT or phrase-based models.

### 5.2.1 Filtering Script

The basic element of this experiment is a script which matches words and phrases in source and target sentences based on entries in the dictionaries.

For purposes of dictionary filtering, we match the lemmas in dictionary entries with lemmas in the considered sentence pair. Technically speaking, we encode translations from the dictionary as pairs of formulas in Conjunctive Normal Form (CNF), where the first formula represents the source language phrase and the second formula represents the target language phrase. Each clause in the formula represents one word of the corresponding phrase and each literal in the clause represents one of all the lemmas possible for the given word. Each input sentence is analyzed into a set of lemmas (literals). Then we build set of usable translations. These are formula pairs from the dictionary, where the first formula is satisfied given literals from the source sentence and the second formula is satisfied given the literals from the target language sentence. We find all such tuples. At the end we consider each word in the sentence as matching if there exists some literal within the found set of formula pairs, where the respective formula contains some of its lemmas as a literal.

Since languages often use some auxiliary words which have rather grammatical than lexical meaning, we also use monolingual dictionaries of such words for each language and these words are considered matching even when they are present just on one side. These could be for example articles, prepositions, pronouns or auxiliary verbs. There are also many words which are not present in the dictionary, but are actually identical or very similar in both languages. These are for example numbers (written using digits) and proper names and they are also considered matching. Since Czech is heavily inflected, we also tried to come up with some heuristics to cover inflections of words unknown to the lemmatizer. For instance, we consider the words matching, if the Czech one has one of the possible possessive affixes and the English one ends with "*'s*" or "".

The tunable parameters of the filter are the maximum allowed out of vocabulary ratio, maximum percentage of non-matching tokens in both sentences and minimal sentence length. During the following experiments, we set minimal length to 2, maximum OOV ratio to 0.33 and varied non-matching tokens ratio to obtain corpora filtered at different levels of strictness.

### 5.2.2 Mixing with General Data

The first approach was to filter the in-domain dataset (which seemed to contain a considerable number of non-matching lines) with variable strictness and mix the resulting sentences with lines from the general corpus (which seemed to have higher quality, so it wasn't filtered) to get 1M sentences to get training data for adaptation.

There were several reasons for choosing this approach:

- The domain data seemed to have many problematic examples – so we tried to improve its quality by filtering.

| Filtering | Medical lines | General lines | 62 | 72 | 76 |
|---|---|---|---|---|---|
| No adaptation | - | - | 21.56 | 21.42 | 20.67 |
| 0 | 28600 | 971400 | 20.99 | 21.25 | 20.5 |
| 0.1 | 141896 | 858104 | 22.99 | 23.18 | 23.28 |
| 0.2 | 393594 | 606406 | **24.71** | **24.23** | 23.67 |
| 0.3 | 459529 | 540471 | 23.47 | 23.44 | **23.98** |
| 0.4 | 600666 | 399334 | 23.75 | 23.47 | 23.83 |
| 0.5 | 628849 | 371151 | 23.66 | 23.48 | 23.54 |
| 0.6 | 642903 | 357097 | 24.39 | 23.24 | 23.27 |
| 0.7 | 653199 | 346801 | 23.71 | 23.38 | 22.9 |
| 0.8 | 659503 | 340497 | 23.24 | 23.41 | 23.34 |
| 0.9 | 662127 | 337873 | 24.07 | 23.98 | 23.27 |
| No filter | 846189 | 153811 | 24.02 | 23.70 | 23.75 |

**Table 9: BLEU scores on HimL test set on three different models adapted using dictionary filtered training data mixed with common data.**

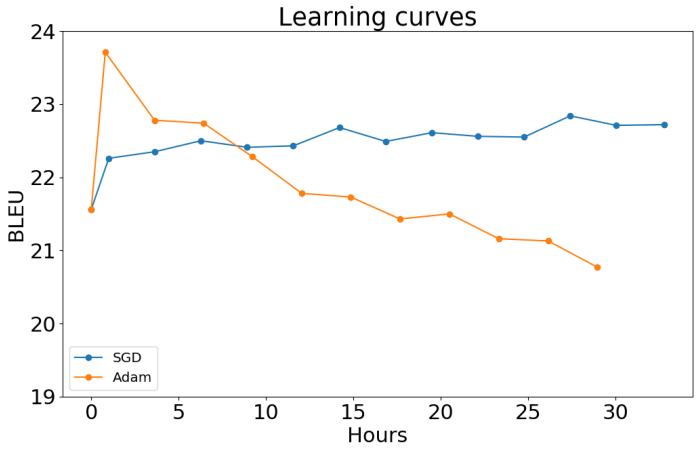


**Figure 7:** Different behavior of SGD and Adam during adaptation on in-domain data only.

- The domain data was relatively small, so we decided to add general data in amount which would make the results comparable by size and sufficient for training translation model and avoiding quick overfitting.
- The general data seemed to have a higher quality, so we didn't filter them to avoid introducing some systematic data sparsity caused by imperfect filtering.

We evaluated the model in terms of BLEU and cross-entropy every 20000 minibatches (roughly 900000 seen training examples, we used adaptive batching) on both dev set and test set and trained until cross-entropy on dev set did not improve over currently best value for 10 consecutive evaluations. BLEU scores reported are scores on HimL test set, achieved by models with best score on dev set. See Table 9 for the results.

We did not perform any SMT experiments with mixed corpora.

### 5.2.3 Domain Data Alone

In this section, we examine the performance of models that do not consider the general corpus in the domain adaptation phase any more.

**NMT** NMT is known to perform badly when trained on small data. Therefore, we used the general model to train a baseline system and then switch to training on the in-domain data, results are shown in Tables 10 and 11. Just for comparison, we also trained one of the models (same parameters as model 62) from scratch on in-domain and mixed data. Surprisingly, in terms of BLEU, these models did not perform dramatically worse than the adapted ones, as you can see in Table 12. The models were again trained until best value of cross-entropy on HimL dev set did not improve for 10 consecutive evaluations and the scores reported are on HimL test set, using model with best dev BLEU score.

| Filtering | Corpus size | 62 | 72 | 76 |
|---|---|---|---|---|
| No adaptation | - | 21.56 | 21.42 | 20.67 |
| 0 | 28600 | 15.64 | 15.83 | 15.96 |
| 0.1 | 141896 | 21.49 | 21.79 | 21.87 |
| 0.2 | 393594 | 23.1 | **24.26** | **23.13** |
| 0.3 | 459529 | **23.71** | 23.35 | 22.25 |
| 0.4 | 600666 | 22.89 | 22.74 | 22.56 |
| 0.5 | 628849 | 21.74 | 22.4 | 22.14 |
| 0.6 | 642903 | 22.54 | 22.96 | 21.76 |
| 0.7 | 653199 | 23.04 | 23.14 | 22.28 |
| 0.8 | 659503 | 23.06 | 23.14 | 22.58 |
| 0.9 | 662127 | 22.93 | 22.28 | 22.34 |
| No filter | 846189 | 23.03 | 23.62 | 22.25 |

**Table 10: Results of domain adaptation on in-domain data alone, HimL test set, Adam. Best scores are written in bold.**

| Filtering | Corpus size | 62 | 72 | 76 |
|---|---|---|---|---|
| No adaptation | - | 21.56 | 21.42 | 20.67 |
| 0 | 28600 | 22.33 | 22.05 | 21.38 |
| 0.1 | 141896 | 22.5 | 22.39 | 22.05 |
| 0.2 | 393594 | **22.91** | **22.8** | 22.2 |
| 0.3 | 459529 | 22.69 | 22.53 | **22.21** |
| 0.4 | 600666 | 22.54 | 22.03 | 22.01 |
| 0.5 | 628849 | 22.35 | 22.26 | 22.01 |
| 0.6 | 642903 | 22.6 | 22.12 | 22.06 |
| 0.7 | 653199 | 22.47 | 22.2 | 21.73 |
| 0.8 | 659503 | 22.58 | 22.36 | 22.01 |
| 0.9 | 662127 | 22.82 | 22.07 | 21.97 |
| No filter | 846189 | 22.70 | 22.39 | 21.93 |

**Table 11: Results of domain adaptation on in-domain data alone, HimL test set, SGD**

As our experiments have shown, the small size of the adaptation corpus can lead to overfitting very quickly: we saw big drops in BLEU scores after a few thousand mini-batches when using the default optimizer Adam. We therefore experimented also with vanilla SGD with learning rate of 0.001 and learning rate decay for the domain adaptation. The difference in training dynamics can be seen in Figure 7. Even though the quality starts decaying rapidly with Adam, in the first few thousand minibatches it outperforms SGD by a margin of 1–2 BLEU. While SGD is improving steadily for some time, it never reached the same BLEU in our settings, so, overall, Adam outperformed SGD by about 0.5–1 BLEU.

Detailed results of domain adaptations runs based on the in-domain corpus filtered at various filtering thresholds are presented in Tables 10 and 11 for SGD and Adam, respectively. Adam, due to the quick overfitting, can damage the performance, see the BLEU drop from the baseline ~21 to ~16. We did not observe similar drops of quality when we trained on mixed data (previous section) or with SGD (Table 11).

If the overfitting is avoided, the performance of adapted NMT is always better than the baseline, sometimes by up to 2.15 BLEU points.

**Moses** For comparison, we also trained a number of phrase-based models with in-domain data only. SMT can handle small data conditions better than NMT so we do not include the general data at all in these runs.

The evaluation was performed on two domains – medical and travel guides. Results are shown in Table 13. Surprisingly, these models did not perform much worse in terms of BLEU score than the adapted NMT models, even though they were trained on relatively small corpora. However, when we manually inspected the output, we found out that most of the translations were much worse than that of NMT model trained on the same data.

### 5.2.4 Summary of Dictionary Filtering

In most NMT experiments the best BLEU score was obtained using filter with strictness of either 0.2 or 0.3, which suggests that, at least with our dataset and settings, dictionary-based filtering can be used to improve NMT performance. This finding may be of course dataset specific. On the other hand, in phrase-based models, filtering the training corpora does seem to harm the

| Filtering | Domain corpus size | Domain only | Mixed |
|---|---|---|---|
| 0 | 28600 | 6.09 | 18.38 |
| 0.1 | 141896 | 14.23 | **24.33** |
| 0.2 | 393594 | 17.97 | 22.27 |
| 0.3 | 459529 | **19.03** | 21.67 |
| 0.4 | 600666 | 18.44 | 21.46 |
| 0.5 | 628849 | 18.83 | 21.38 |
| 0.6 | 642903 | 19.0 | 21.96 |
| 0.7 | 653199 | 18.49 | 21.27 |
| 0.8 | 659503 | 18.95 | 21.92 |
| 0.9 | 662127 | 18.41 | 21.28 |
| No filter | 846189 | 18.55 | 21.33 |

**Table 12: Results of training on domain / mixed data from scratch, model 62, corpus size was 1M sentences for mixed data.**

| Filtering | Corpus size | BLEU |
|---|---|---|
| 0.3 | 142130 | 54.68 |
| 0.5 | 174287 | 59.22 |
| 0.9 | 190202 | 61.59 |
| 1.0 (no filter) | 244778 | **69.38** |

Travel guides

| Filtering | Corpus size | BLEU |
|---|---|---|
| 0.0 | 28601 | 11.96 |
| 0.2 | 393595 | 23.16 |
| 0.9 | 662128 | 23.24 |
| 1.0 | 850530 | **23.46** |

Medical

**Table 13: Effects of training set filtering on phrase-based translation performance.**

performance significantly. This may be due to the fact that random errors are likely to have low probability, thus they are rarely used to prepare final result of translation, on the other hand, sentence pairs which were filtered out may contain rare examples, which would be very useful for building phrase table.

## 5.3 Adding Dictionary Data to Training Corpora

Another way to make use of our dictionary data is to incorporate them directly into the training data. In this section, we look into effects of this enhancement of training corpora both in NMT and phrase-based translation.

### 5.3.1 Data Preparation

We took entries from the dictionary and expanded them to cover all morphology forms. We checked tags assigned to these forms and filtered out those that weren't matching across languages.

For en-cs translation we first checked for an exact tag match. Since English does not have grammatical cases, we used nominative case for Czech side in all instances. This turned out to be a rather strict requirement, leading to a weak expansion, but nevertheless helping a lot.

For ru-cs, we tried both the strict matching, as above, and a more tolerant one. In the softer approach, we added the word pair into the training data even when there were no tags matching, and then again each time part-of-speech tags matched, part-of speech, case and grammatical number tags matched or the whole exact tag matched. This means more matching words are added multiple times into the training data, to reflect that they should be more probable translation.

### 5.3.2 Experiments

For phrase-based models, we tried two approaches of incorporating dictionary data – creating one phrase table with all the data simply concatenated (corpus + lexicon), or creating a back-off phrase table solely from dictionary data for cases when a token from a source sentence is not found in general phrase table. We ran the experiments on our general translator from Russian to Czech, and three of the systems used in previous section, medical (reported score is on HimL test set) and two travel guides systems. The results are presented in Table 14. We see that adding expanded dictionary entries was helpful for each of the tested domains as well as for the general domain. It is also apparent that more tolerant version of tag matching performed better, by about 0.9 BLEU in Russian translation. The simpler method of adding the dictionary to the training corpus seems more successful than the separate phrase table as a fall-back. There are number of possible reasons for this outcome:

| Model | BLEU |
|---|---|
| Guides, en-cs, no filtering, +lex | **70.91** |
| Guides, en-cs, no filtering | 69.38 |
| Guides, en-cs, filter 0.3, +lex | **60.37** |
| Guides, en-cs, filter 0.3 | 59.92 |
| Medical, en-cs, no filter, +lex | **24.58** |
| Medical, en-cs, no filter | 23.46 |
| General, ru-cs, no filter, +lex, tolerant matching | **22.79** |
| General, ru-cs, no filter, +lex, exact matching | 21.9 |
| General, ru-cs, no filter | 21.57 |

Adding dictionary data to training corpus

| Model | BLEU |
|---|---|
| Guides, en-cs, no filtering, +lextable | 69.38 |
| Guides, en-cs, no filtering | 69.38 |
| Guides, en-cs, filter 0.5, +lextable | 59.90 |
| Guides, en-cs, filter 0.5 | 59.92 |

Creating back-off phrase table from dictionary data

**Table 14: Impact of dictionary data on phrase-based model performance**

| Model | BLEU |
|---|---|
| General en-cs, +lex | 19.67 |
| General en-cs | **21.20** |

**Table 15: Addition of dictionary data in NMT**

- It may help with word alignment: short segment pairs coming from the dictionary are a very clear indication for the word-alignment model which words are translations of which.
- It helps to avoid OOV words by simply adding them to the training corpus together with translation which is correct for some contexts.
- It has a minor impact on probabilities in the translation model and this impact is rather positive. All known translations get one more occurrence in training data, which should not affect the order of translation options covered well by the corpus already. Previously unknown translations, on the other hand, get a small but non-zero probability.

The second approach of a separate fall-back phrase table just solves OOV but it doesn't introduce realistic probabilities because in the dictionary all translation options have an equal probability.

For NMT, we only concatenated the dictionary data with a general purpose corpus. The results in Table 15 indicate a considerable loss in performance. This is most likely due to nature of the added training segments: dictionary entries are much shorter than regular sentences and the NMT system does not see the words in a real context.

### 5.3.3 Summary of Adding Dictionary Entries

Adding dictionary data to training sets has proved to be beneficial to BLEU scores for phrase based translation systems. Creating a back-off phrase table did not affect overall performance significantly, but it can be used to achieve more understandable translation by translating tokens that were not present in the training data.

On the other hand, for an NMT model, adding dictionary data in this form seems to have negative effect on the performance.

We assume that this different behavior is caused by the fact that PBMT has a separate translation model (affected by adding of dictionary data) and language model (not affected by adding of dictionary data) whereas NMT has one joint model which then learns to translate incomplete sentences.

## 5.4 Reranking According to Dictionary Correspondence

We also tried to rerank n-best lists produced by an NMT model using our dictionary data. We computed a score for each sentence pair. We used Nematus toolkit for the following experiments.

### 5.4.1 Reranking Score

We took these numbers as the input to our reranking score:

[14] https://github.com/EdinburghNLP/nematus

- the number of source words with missing translation ($N_t$) – these source words are known to the dictionary but none of their known translations is present in the translation,
- the number of target words with missing source ($N_s$) – these target words are known to the dictionary but none of their known translations is present in the source,
- the number of source words with unknown translation ($U_s$) – source word is missing in the dictionary,
- the number of target words with unknown translation ($U_t$) – target word is missing in the dictionary,
- the number of input tokens ($n$).

When combining these numbers into the final score, we should not penalize input for unknown words and we should accept their translations although these are unknown. A reasonably permissive score seems to be $M = \frac{N_t+|N_s+U_t-U_s|}{n}$.

To get overall metric, we normalize (divide) the combined counts of problematic words by length of input sentence. This keeps the result always non-negative and for good results under 1. For our baseline tests we simply added the result of the metrics to the entropy as produced by Nematus. We also trained a simple feed-forward neural network, which input consisted of the metrics mentioned above, score from Nematus and a position of the sentence in n-best list. Based on these features, we tried to predict BLEU score of the translation and rerank the sentences accordingly.

### 5.4.2 Results of Reranking

The results of reranking experiments are so far only preliminary. These techniques seemed to help mainly with clearly mistranslated sentences like those with repeating tokens at the end. On the other hand, circumstances of its deployment (for example bigger beamsize) may result in worse performance of the model itself, thus in our case it merely compensated its deployment drawbacks. Without changing other properties of model and search, it may work as a safety test for detecting weird translations. We know this should be tested with more models and we plan to further investigate this in the future.

# 6 Transformer NMT

We have also trained the Transformer model (Vaswani *et al.*, 2017) implemented in the Tensor2tensor (T2T) framework[15] version 1.2.1.

## 6.1 Common settings

For training, we used parallel data from UMC[16] and synthetic parallel data (synth).[17] Both resources were preprocessed using MorphoDita[18] tool. Data sizes are summarized in Table 16. For development, we used the HimL *dev* set, which is a concatenation of Cochrane and NHS24 2014 tuning datasets.[19] For the final evaluation in Table 17, we used the HimL 2015 test sets (Cochrane and NHS24 separately). See D5.6 for a full evaluation including human ranking.

| | sent. pairs | EN tokens | non-EN tokens |
|---|---|---|---|
| EN-CS UMC | 50 M | 467 M | 389 M |
| EN-CS synth | 7 M | 149 M | 135 M |
| EN-DE UMC | 48 M | 707 M | 651 M |
| EN-PL UMC | 40 M | 402 M | 316 M |
| EN-RO UMC | 62 M | 557 M | 484 M |
| EN-RO synth | 10 M | 178 M | 172 M |

**Table 16: Training data sizes.**

We have used the default T2T setup with the following specifics (unless stated otherwise in Section 6.2):

- We built a subword vocabulary for each language pair, always using the joint vocabulary for both languages. For this purpose, we increased the `file_byte_budget`, that is the amount of training data used for building subword units (with the T2T internal subword algorithm (Wu *et al.*, 2016), which is similar to BPE (Sennrich *et al.*, 2015)). In the English-Polish experiment, with the default value 700 KiB for English and 700 KiB for Polish, we got a vocabulary of only 26k subwords, although our goal was 32k.[20] This vocabulary was based also on singletons (words with a single occurrence within the first 700 KiB of the training data), so we did not consider it reliable. We thus increased `file_byte_budget` to 10 MiB.
- We trained all models with 8 GPUs (GeForce GTX 1080 Ti with 11 GiB memory) and thus used `--worker_gpu=8`. The largest batch size (in subwords) we could use was 1500.
- We used the `transformer_big_single_gpu` hyper-parameter settings. According to our experience, it is better than `transformer_big` even for multi-GPU training, at least with batch size 2048 or lower.
- We disabled the internal evaluation with `--local_eval_frequency=0` because it is currently not compatible with the multi-GPU training in T2T.
- Unlike Nematus, T2T currently does not support ensembling, only checkpoint averaging. For the final models, we average last 8 or 16 checkpoints with the `utils/avg_checkpoints.py` script.
- We used `--save_checkpoints_secs=3600` to save checkpoints each hour. According to our experience, this is better than the default 10-minutes interval for the purpose of checkpoint averaging (given a fixed number of checkpoints we can average models from a 6 times longer time span, covering more diverse models and achieving a higher final BLEU).

## 6.2 Language-specific settings

- For English-Czech, we used the common settings described above. As described in Section 6.3, we used UMC+synth training data for the final run. The final model was obtained by averaging 8 checkpoints after 3 days of training. Training for one more day did not bring any improvement according to BLEU on the dev set.

[15] https://github.com/tensorflow/tensor2tensor
[16] http://ufal.mff.cuni.cz/ufal_medical_corpus
[17] http://www-int.hink.eu/systems/nmt-training/corpora/synthetic
[18] http://ufal.cz/morphodita
[19] http://data.statmt.org/data/himl/himl-test-v2.tgz
[20] Our hypothesis is that there was not enough diversity in the beginning of the English-Polish training data. This is also influenced by the fact that the data was already tokenized. T2T can build subwords from untokenized data, but except for German-English we used the tokenized data for better comparability with the other MT systems (which used the same tokenized training data).

| | EN-CS | | EN-DE | | EN-PL | | EN-RO | |
|---|---|---|---|---|---|---|---|---|
| | cochr | NHS24 | cochr | NHS24 | cochr | NHS24 | cochr | NHS24 |
| Y3 system | 33.34 | 26.36 | 39.09 | 33.66 | 22.47 | **28.32** | **39.05** | **34.02** |
| Transformer | **39.31** | **28.59** | **39.83** | **34.00** | **22.53** | 24.78 | 33.82 | 28.13 |
| training time | 72 h | | 216 h | | 97 h | | 72 h | |

**Table 17: Transformer BLEU results on the 2015 HimL test sets**

- For English-Romanian, we were able to use `batch_size=2048` (instead of 1500), but we had only 7 GPUs available (instead of 8). As described in Section 6.3, we used UMC training data only for the final run. The final model was obtained by averaging 8 checkpoints after 3 days of training. Training for up to 7 days led to minor BLEU improvements only (< 1 BLEU) for the model without averaging and almost no improvement (±0.1 BLEU) with averaging (trying 8, 16 and 32 checkpoints).
- For English-Polish, the final model was obtained by averaging 16 checkpoints after 4 days and one hour of training. We have not tried longer training, but the training seemed to be plateaued as there has been just +0.2 BLEU improvement on the dev set over the last two days.
- For English-German, we decided to exploit a previously trained model and subword vocabulary within the `translate_ende_wmt32k` problem class in T2T (with 4.5M sentence pairs from WMT). For this reason, we used untokenized UMC training data for English-German (unlike for the other language pairs).

  Our first experiment with taking the pretrained model and continuing training on UMC was not successful because the training diverged (probably due to too high learning rate).

  In a second experiment, we tried multi-task learning on the WMT data and on the UMC data using options `--problems=himl_ende_umc-translate_ende_wmt32k --hparams='batch_size=1024,learning_rate_warmup_steps=30000,shared_embedding_and_softmax_weights=0'`. We achieved BLEU=39.64 on the dev set after 6 days of training.

  Concurrently, we ran a third experiment, where we trained on UMC data only (without multi-task with WMT), using options `--problems=himl_ende_umc --hparams='batch_size=1500,learning_rate_warmup_steps=30000'`. We achieved BLEU=40.06 on the dev set after 9 days of training with just a small (0.6 BLEU) but steady improvements over the last 6 days. We choose this experiment for the final model after averaging the last 16 checkpoints.

### 6.3 Effect of synth data

For Czech and Romanian we had access to the synth training data, so we did an experiment with adding it to the UMC training data. Note that T2T always shuffles all data before training.

In English-Czech, the addition of 7M synth sentence pairs to the training data was helpful. It increased BLEU on the dev set from 31.64 (UMC only, 4 days of training) to 35.99 (UMC+synth, 3 days of training).

For English-Romanian, the addition of 10M synth sentence pairs was not helpful. It decreased BLEU on the dev set from 30.27 (UMC only, 3 days of training) to 26.01 (UMC+synth, 3 days and 16 hours of training).

We were surprised by this deterioration, so we inspected the translation output and noticed that many "normal" English sentences are translated either as *"( 3/3/2015 8 : 10 : 02 PM )"* (or a similar data&time expression) or as one particular sentence *"Pagina 1 - afișăm firmele de la 1 până la 1 ."*.[21] Afterwards, we confirmed it is caused by the synth training data, where many English sentences are mistranslated (mis-aligned) in the same way. These two types of mis-alignment are easy to filter out, but we noticed many other problems in the EN-RO synth training data.

## 7 Conclusion

This Deliverable 2.3 described our activities on semantics in machine translation in the third year of the HimL project. We managed to quickly respond to the shift towards neural MT and the vast majority of our experiments this year were already performed in this framework.

[21] Out of the 1961 sentences in the dev set, 60 were translated as "Pagina 1…" and 273 were translated with one of 21 date&time expressions. None of these translations was correct. We could not spot any pattern among the source sentences which were mistranslated this way. They varied in length and topic. For example one of them was *"NHS inform is a new national health information service"*.

We documented the improvements in performance thanks to dedicated handling of syntax and semantic roles. We re-evaluated errors in negation and concluded that NMT suffers almost no errors of this kind, when large data and all state-of-the-art tricks and techniques are used.

We experimented with better models of neural MT, specifically the reconstruction technique and the Transformer model, getting best results with these (see D5.6), and we also got improvements from large manually edited dictionaries.

# References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. "Neural machine translation by jointly learning to align and translate." *CoRR*, abs/1409.0473.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. "Neural versus phrase-based machine translation quality: a case study." *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 257–267.

Bojar, Ondřej, Rudolf Rosa, and Aleš Tamchyna. 2013. "Chimera – three heads for english-to-czech translation." *Proceedings of the Eight Workshop on Statistical Machine Translation*, 92–98. Sofija, Bulgaria.

Hajič, Jan, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková Razímová. 2006. "Prague Dependency Treebank 2.0." LDC2006T01, ISBN: 1-58563-370-4.

Luong, Minh-Thang, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. "Multi-task sequence to sequence learning." *Proceedings of International Conference on Learning Representations (ICLR 2016)*.

Nadejde, Maria, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. 2017. "Predicting Target Language CCG Supertags Improves Neural Machine Translation." *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*. Copenhagen, Denmark.

Rosa, Rudolf, David Mareček, and Ondřej Dušek. 2012. "DEPFIX: A system for automatic correction of czech MT outputs." *Proceedings of the Seventh Workshop on Statistical Machine Translation*, 362–368. Montréal, Canada.

Sennrich, Rico and Barry Haddow. 2016. "Linguistic input features improve neural machine translation." *Proceedings of the First Conference on Machine Translation*, 83–91. Berlin, Germany.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2015. "Neural machine translation of rare words with subword units." *CoRR*, abs/1508.07909.

Shi, Xing, Inkit Padhi, and Kevin Knight. 2016. "Does string-based neural mt learn source syntax?" *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1526–1534. Austin, Texas.

Steedman, Mark. 2000. *The syntactic process*, vol. 24. MIT Press.

Tu, Zhaopeng, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. "Neural machine translation with reconstruction." *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 3097–3103.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention is all you need." *CoRR*, abs/1706.03762.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. "Google's neural machine translation system: Bridging the gap between human and machine translation." *CoRR*, abs/1609.08144.