qtleap

quality
translation
by deep
language
engineering
approaches

Report on the third MT pilot and its evaluation

DELIVERABLE D2.11

VERSION 1 | 2016-10-28

QTLeap

Machine translation is a computational procedure that seeks to provide the translation of utterances from one language into another language.

Research and development around this grand challenge is bringing this technology to a level of maturity that already supports useful practical solutions. It permits to get at least the gist of the utterances being translated, and even to get pretty good results for some language pairs in some focused discourse domains, helping to reduce costs and to improve productivity in international businesses.

There is nevertheless still a way to go for this technology to attain a level of maturity that permits the delivery of quality translation across the board.

The goal of the QTLeap project is to research on and deliver an articulated methodology for machine translation that explores deep language engineering approaches in view of breaking the way to translations of higher quality.

The deeper the processing of utterances the less language-specific differences remain between the representation of the meaning of a given utterance and the meaning representation of its translation. Further chances of success can thus be explored by machine translation systems that are based on deeper semantic engineering approaches.

Deep language processing has its stepping-stone in linguistically principled methods and generalizations. It has been evolving towards supporting realistic applications, namely by embedding more data based solutions, and by exploring new types of datasets recently developed, such as parallel DeepBanks.

This progress is further supported by recent advances in terms of lexical processing. These advances have been made possible by enhanced techniques for referential and conceptual ambiguity resolution, and supported also by new types of datasets recently developed as linked open data.

The project QTLeap explores novel ways for attaining machine translation of higher quality that are opened by a new generation of increasingly sophisticated semantic datasets and by recent advances in deep language processing.

www.qtleap.eu

Funded by

QTLeap is funded by the 7th Framework Programme of the European Commission.

Supported by

And supported by the participating institutions:

Faculty of Sciences, University of Lisbon

German Research Centre for Artificial Intelligence

Charles University in Prague

Bulgarian Academy of Sciences

Humboldt University of Berlin

University of Basque Country

University of Groningen

Higher Functions, Lda

Revision history

Version	Date	Authors	Organisation	Description
0.1	Sep 2, 2016	Ondřej Dušek, Martin Popel	CUNI	first draft
	Sep 27, 2016	Eleftherios Avramidis, Aljoscha Burchardt, Jindřich Helcl, Vivien Macketanz	DFKI	description of German Pilot and manual evaluation
	Sep 30, 2016	Gertjan van Noord	UG	description of Dutch components
	Sep 30, 2016	Gorka Labaka	UPV/EHU	description Basque and Spanish components
	Sep 30, 2016	João Rodrigues	FCUL	description of Portuguese components
	Sep 30, 2016	Gorka Labaka	UPV/EHU	description of Basque and Spanish Chimera system’s deviations
	Oct 3, 2016	Nora Aranberri	UPV/EHU	description of Basque and Spanish qualitative analysis
	Oct 3, 2016	António Branco	FCUL	description of Portuguese system
	Oct 4, 2016	Andreia Querido	FCUL	description of Portuguese qualitative analysis
0.2	Oct 6, 2016	Kiril Simov	IICT-BAS	description of the Bulgarian system and qualitative analysis
0.3	Oct 7, 2016	Petya Osenova	IICT-BAS	internal review
0.4	Oct 11, 2016	Martin Popel	CUNI	integrated comments from the internal review
1.0	Oct 28, 2016	Kiril Simov	IICT-BAS	improved description of the Bulgarian system

Statement of originality
This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Report on the third MT pilot and its evaluation

DOCUMENT QTLEAP-2016-D2.11
EC FP7 PROJECT #610516

DELIVERABLE D2.11

completition
FINAL
status
SUBMITTED
dissemination level
PUBLIC

responsible
Jan Hajič (WP2 Coordinator)
reviewer
Petya Osenova
contributing partners
CUNI, FCUL, DFKI, IICT-BAS, UPV/EHU, UG, HF

authors
Ondřej Dušek, Martin Popel, Vivien Macketanz, Aljoscha Burchardt, Eleftherios Avramidis, Gertjan van Noord, João Rodrigues, António Branco, Gorka Labaka, Kiril Simov, Aleksandar Popov

Contents

List of Abbreviations

BDT	Basque Dependency Treebank
FGD	Functional Generative Description
HMM	Hidden Markov Model
HMTM	Hidden Markov Tree Model
LM	language model
MERT	minimum error rate training
MRS	minimal recursion semantics
MT	machine translation
NED	named entity disambiguation
NERC	named entity recognition and classification
NLP	natural language processing
NMT	neural machine translation
PB-SMT	phrase-based statistical machine translation
PDT	Prague Dependency Treebank
SMT	statistical machine translation
SRL	semantic role labeling
STSG	synchronous tree substitution grammar
TM	translation model
TreeLM	target-language tree model
UD	Universal Dependencies
WER	word error rate
WMT	workshop/conference on statistical machine translation
WSD	word sense disambiguation

1 Overview – What's new in Pilot 3

This deliverable D2.11 describes and evaluates Pilot 3 (the third MT pilot system, enhanced with deep semantics), which was released as deliverable D2.10. We focus on the changes and improvements done since finalizing Pilot 2 (the second MT pilot system, enhanced with lexical semantics, delivered in D2.7 and described in D2.8 [Popel et al., 2015]). As in previous pilots, we experimented with translations from English (for helpdesk answers) and into English (helpdesk questions). The languages involved are: Basque (EU), Bulgarian (BG), Czech (CS), Dutch (NL), German (DE), Portuguese (PT) and Spanish (ES).

The main improvements of Pilot 3 consist on adding deep semantics and advanced semantic linking and resolving as described in detail in D4.13 and D5.11. This deliverable focuses on the remaining improvements (listed below) and the intrinsic evaluation (detailed in Section 4).

- In addition to Pilot 3, we have developed another system called Chimera for 5 language pairs: en→cs, en→es, en→eu, en→nl and en→pt, that is all English-to-X language pairs which use TectoMT. Chimera combines Pilot 3 with Moses phrase-based system, trying to improve upon both of them. We use the name P3-Chimera to emphasize it is based on Pilot 3. See Section 3 for the full description.

- The analysis, synthesis and transfer of TectoMT has been improved in many aspects for language-specific phenomena. These improvements were driven also by the error analysis of Pilot 2 (MQM and post-editing as presented in Section 3.2 of D2.8). One notable improvement is a new statistical module for definiteness assignment in en→cs transfer[1] (Section 2.4.2).

- Section 2.8 describes the improved DeepFactoredMoses system, which includes transfer of linguistic knowledge from the source to the target language in the postprocessing phase.

- Section 2.9 describes the extended and improved Qualitative system combination, which now includes a new Neural Machine Translation system.

- Translation models trained with Vowpal Wabbit were integrated into TectoMT and tested in en→cs. See Section 2.1.5.

Pilot 3 is the final pilot of the QTLeap project. Its intermediate version has been manually evaluated also within the IT domain translation shared task of the First Conference on Statistical Machine translation (WMT 2016).[2] This task was organized by QTLeap and attracted the attention of both researchers and companies. Systems submitted by QTLeap partners [Gaudio et al., 2016, Rosa et al., 2016b] competed with other systems submitted by researchers or companies. See Bojar et al. [2016a, Section 4] for the successful results of the manual and automatic evaluation and a further discussion.

[1] Czech does not have the grammatical category of definiteness, so the information cannot be initialized with the source-side values as in other source-side languages.

[2] http://www.statmt.org/wmt16/it-translation-task.html

2 Pilot 3 Systems

2.1 General structure of TectoMT-based Systems

TectoMT is a structural machine translation system with deep transfer, first introduced by Žabokrtský et al. [2008], with continuous enhancements up to today, including work on the QTLeap project. TectoMT uses two layers of structural description, the shallow *a-layer* (see Section 2.1.1) and the deep *t-layer* (see Section 2.1.2).

The system is composed of a pipeline with 3 main phases:

1. Source language analysis, proceeding over the a-layer to the t-layer (see Section 2.1.3).
2. Transfer on the t-layer, based on Hidden Tree Markov Models [Žabokrtský and Popel, 2009] and context-sensitive translation models, either the original Maximum Entropy models [Mareček et al., 2010] (Section 2.1.4) or the novel VowpalWabbit models (Section 2.1.5).
3. Synthesis/generation that gradually changes the t-layer representation into surface target language string (cf. Section 2.1.6).

All of the above mentioned sections describe the trained system in operation; system training is summarized in Section 2.1.7.

2.1.1 The *a-layer*: surface structural description

The *a-layer* (analytical layer) is a layer of surface syntactic description which includes all tokens of the sentence, organized as nodes (*a-nodes*) into a labeled dependency tree (*a-tree*).

Each a-layer carries the following information (among others):

- *word form* – the inflected surface word form as it appears in the sentence (including capitalization).
- *lemma* – the base form of the word; e.g., infinitive for verbs, nominative singular for nouns.
- *part-of-speech tag* and *morphological information* – all possible morphological categories (e.g., gender, case, tense). Interset [Zeman, 2008] is used to facilitate language-independent rules in TectoMT.
- *afun* – surface dependency label. The labels largely correspond to commonly commonly used grammatical functions such as subject, predicate, object, and attribute (`Sb`, `Pred`, `Obj`, `Atr`).

2.1.2 The *t-layer*: deep structural description for transfer

The *t-layer* (tectogrammatical layer) is a deep syntactic/semantic layer describing the linguistic meaning of the sentence according to the Functional Generative Description (FGD) theory of Sgall et al. [1986]. The t-layer is also represented as dependency trees (t-trees), but these only include content words (nouns, full verbs, adjectives, adverbs) as nodes (*t-nodes*).

Auxiliary words (prepositions, articles, auxiliary verbs) are not present on the t-layer as separate nodes but can influence other t-nodes' attributes. There are also nodes on the t-layer that do not correspond to any surface words, e.g., nodes representing pro-dropped subject personal pronouns.

Coreference is marked on the t-layer using special coreference links (non-tree edges).

A *t-node* has the following main attributes:

- *t-lemma* – "deep lemma" (mostly identical to surface lemma).
- *functor* – a semantic role label. There are over 60 different semantic role labels based on the FGD theoretical framework, such as `ACT` (actor/experiencer), `PAT` (patient/deep object), `TWHEN` (time adverbial), `RSTR` (modifying attribute) etc.
- *grammatemes* – a set of deep linguistic features relevant to the meaning of the given sentence (e.g., person, number, tense, modality).
- *formeme* – morpho-syntactic form information [Žabokrtský, 2010], composed of coarse-grained part-of-speech based on syntactic behavior, prepositions or subordinate conjunctions, and coarse-grained syntactic form (e.g., `v:to+inf` for infinitive verbs or `n:into+X` for a prepositional phrase).

2.1.3 Analysis

The analysis in TectoMT first uses standard dependency parsers trained on treebanks to reach the a-layer Popel et al. [2011]. A-layer parsing is preceded by preprocessing steps which include sentence segmentation, tokenization, lemmatization, and morphological tagging.

The a-tree is then gradually transformed into a t-tree by modules that build the t-tree by removing auxiliary words, changing surface lemmas to t-lemmas, and assigning formemes, functors, and grammatemes to each node. Final stages of the t-layer analysis pipeline involve reconstructing deep subjects (for pro-drop languages, imperatives, and passive) and coreference resolution.

2.1.4 Transfer: translation factorization

The transfer on the t-layer is separated into three relatively independent simpler subtasks: the translation of t-lemmas, formemes and grammatemes [Žabokrtský, 2010]. This approach makes a strong assumption that topology changes to t-trees are rarely needed as t-trees representing the same content in different languages should be similar. This allows us to model each of these three subtasks by a symmetric source-target one-to-one mapping.

The t-lemma and formeme transfer is treated jointly in the following main steps:

1. Producing an n-best list of translation variants using t-lemma translation model(s)
2. Producing an n-best list of translation variants using formeme translation model(s)
3. Joint re-ranking of the n-best lists using Hidden Markov Tree Models (HMTM)

For each t-lemma/formeme in a source t-tree, the translation model (TM) assigns a score to all possible translations observed in the training data, resulting in an n-best list of most probable translations. This score is a probability estimate of the translation variant given the source t-lemma/formeme and other context, and it is calculated as a linear combination of several components:

- Discriminative TMs – prediction is based on features extracted from the source tree using a maximum entropy (MaxEnt) model [Mareček et al., 2010] or using VowpalWabbit model described in the next section.
- Dictionary TMs – this is only a dictionary of possible translations with relative frequencies (no contextual features are taken into account, called *static* in the source codes).
- Other – backoff components that focus on out-of-vocabulary t-lemmas using hand-crafted rules and various small dictionaries of morphological derivation.

Since the Pilot 2 version of TectoMT, multiple discriminative/dictionary TMs are used – a general-domain TM and an in-domain TM [Rosa et al., 2015]. TM interpolation is thus used for two purposes in TectoMT: first, to combine high-precision discriminative TMs with high-coverage dictionary models, and second, for domain adaptation (to the IT domain in case of QTLeap).

The n-best lists of most probable translations for t-lemmas and formemes are jointly re-ranked by Hidden Markov Tree Models (HMTMs), [Crouse et al., 1998, Žabokrtský and Popel, 2009]. HMTMs are similar to standard (chain) Hidden Markov Models but operate on trees. Transition probability is modeled by a tree language model, while emission probability is the probability of the particular source-language t-lemma/formeme being a translation of the hidden target-language t-lemma/formeme.

The translation of grammatemes is much simpler than the translation of t-lemmas and formemes since abstract linguistic categories such as tense and number are usually paralleled in both source and target languages. Therefore, a set of simple rules (with a list of exceptions) is sufficient for this task in all language pairs.

2.1.5 Transfer using VowpalWabbit

Although the MaxEnt translation models[3] are powerful, we have decided to substitute them with one model trained with VowpalWabbit [Langford et al., 2007] machine learning toolkit.[4] We did the first experiments on en→cs and t-lemmas, but the application to other language pairs and to formeme model is straightforward.[5] VowpalWabbit model has several advantages over the original MaxEnt model:

- Only one model for all t-lemmas is trained instead of a separate model for each source t-lemma. This is technically easier to work with. It also opens space for exploring novel features shared across multiple source t-lemmas (so-called *transfer*

[3] The MaxEnt models used in TectoMT are implemented in `Treex::Tool::ML::MaxEnt` (`https://github.com/ufal/treex/tree/master/lib/Treex/Tool/ML/MaxEnt`) which is in turn based on `AI::MaxEntropy` (`https://metacpan.org/pod/AI::MaxEntropy`).

[4] The VowpalWabbit translation models in TectoMT are implemented in `T2T::EN2CS::TrLAddVariantsVW2` block (`https://github.com/ufal/treex/blob/a65b6ce1/lib/Treex/Block/T2T/EN2CS/TrLAddVariantsVW2.pm`) which uses VowpalWabbit from `https://github.com/JohnLangford/vowpal_wabbit`.

[5] VowpalWabbit transfer model has been trained also for Spanish (Section 2.7.2), where it included WSD features, but this experiment did not bring any improvements, so it was not integrated into the final en→es Pilot 3.

learning using *label-dependent features*, but we have not experimented with them yet.

- The training is many times faster. Training MaxEnt t-lemma models on CzEng 1.0 takes more than one day when parallelized on 200 cores in SGE cluster (one needs to wait until the last t-lemma model is trained). Training Vowpal Wabbit t-lemma model on CzEng 1.0 takes less than two hours (with 2-pass training) on a single machine (2 cores). Both approaches require to extract the training data into a suitable format, which can be easily parallelized and takes several hours on the 200 cores cluster. It is obvious that Vowpal Wabbit allows researchers to try many more experimental setups than the MaxEnt in the same amount of time.

- No pruning of training data is needed. In order to be able to train the MaxEnt models in reasonable time, we had to limit the number of training instances per one source t-lemma to 10,000 and exclude source t-lemmas with less than 100 training instances. In VowpalWabbit no such training is needed because of the fast online learning and also because the model takes less space thanks to feature hashing.[6]

- VowpalWabbit's is trained with online learning, which allows domain adaptation using resumed learning. In our en→cs setting it means we first train two passes on CzEng and save the model. Then we take the model and continue training it with two more passes on Batch1a. Batch1a is much smaller than CzEng (1 K sentences versus 15 M sentences), but online training is more sensitive to the later training examples, so this approach is quite effective.

- The translation quality is significantly better than MaxEnt. We observed +1.33 BLEU improvement on en→cs Batch3a when using VowpalWabbit instead of the original MaxEnt. We have not done a proper ablation analysis yet to test which components are responsible for this improvement.[7]

Technically, we use cost-sensitive one-against-all reduction to logistic regression with label-dependent-features. The exact training commands are as follows:

```
$ vw -d czeng.dat.gz -f czeng.model -c --holdout_off -l 3 --passes=2 \
     --loss_function=logistic --csoaa_ldf=mc --probabilities -b 29 -qST

$ vw -d batch1a.vw -f final.model -c --holdout_off -l 3 --passes=2 \
    --loss_function=logistic -i czeng.model
```

Feature space `S` contains all the source-language context features. Feature space `T` contains the conjunction of source and target t-lemma.

We have improved VowpalWabbit by implementing the option `--probabilities`, which results in outputting the whole distribution of all possible translation options and

[6] Moreover, the size of the model trained with VowpalWabbit can be adapted. We use 29-bit hash function, so the models take about 3 GiB of disk and 8 GiB of memory. By using 27 bits, we could scale down the model to 2 GiB of memory with just a tiny degradation in translation quality.

[7] Possible components responsible for this improvement are: using modern machine learning in VowpalWabbit instead of MaxEnt, no pruning, online domain adaptation instead of translation models interpolation. We also use a slightly enriched feature set, which considers e.g. conjunction of neighboring t-lemmas and formemes as features, while our MaxEnt considered them only separately. In future, we plan to investigate this and hopefully find even more effective features and learning settings.

their probabilities (otherwise, VowpalWabbit reports only the most probable translation). We need this distribution because we combine the TM predictions with TreeLM scores using HMTM (cf. the previous section). The option `--probabilities` also instructs VowpalWabbit to report the multi-class logistic loss, which we consider a better intrinsic quality indicator for our purposes than the zero-one loss which is reported by default.

Experimental results of the VowapalWabbit integration are reported in D5.11.

2.1.6 Synthesis (generation)

The synthesis phase is a series of small, mostly rule-based modules that perform gradual changes on the t-trees, converting them to a-trees that contain inflected word forms and can be linearized to plain text. Generators in this scenario are designed to be domain-independent and known to reach high performance [Ptáček and Žabokrtský, 2006, Žabokrtský et al., 2008, Dušek et al., 2012].

The tasks carried out by the modules in the pipeline are language-specific but generally tackle the following problems:

- *Word ordering* – word order required by the target language is enforced.
- *Agreement* – morphological attributes are deduced based on grammatical agreement with surrounding nodes (as in subject-predicate agreement or noun-attribute agreement).
- *Inserting grammatical words* – a-nodes are created for prepositions, subordinate conjunctions, auxiliary verbs, particles, articles, punctuation, and other grammatical words which do not have separate nodes in t-trees.
- *Inflection and phonetics* – inflected word forms are produced based on known morphological and phonetic information from the context.
- *Capitalization* – the first word in a sentence is capitalized.

2.1.7 System training

Apart from VowpalWabbit (Section 2.1.5), there have been no major changes in the TectoMT training process since Pilot 1; therefore, we include here only a brief overview of the training process. A more detailed description can be found in D2.4.

While most analysis components (taggers, parsers, see Section 2.1.3) are trained in a standard fashion on annotated corpora and treebanks, training the translation models requires a more complex procedure using automatic annotation to obtain large deep parallel treebanks. This is due to two reasons – first, the process follows the real-life scenario where error-prone automatic features are extracted from the data, and second, the TMs require very large parallel treebanks, which are expensive to obtain by manual work.

We obtain parallel deep treebanks by using the analysis pipelines for both respective languages (see Section 2.1.3), starting from sentence-aligned bitexts and going through tokenization, morphology, dependency parsing to a-layer and t-layer conversion. The analysis pipeline is run independently on each of the two languages.

Word-alignment between t-nodes is obtained in three steps: automatic word alignment using the GIZA++ tool [Och and Ney, 2003], projection to the corresponding t-nodes, and additional heuristic rules used to align t-nodes that have no counterparts on the surface.

Note that parallel treebanks can be used for training translation models in both translation directions.

2.1.8 System testing

Since the Pilot 2 version of TectoMT, the TectoMT scenarios (sequences of TectoMT modules that make up the translation pipeline) are listed and versioned in special Perl modules in the main Treex/TectoMT Git repository.[8] This allows for parametric scenarios and easy synchronization of the scenarios with the source code of the modules ("blocks") their contain.

The testing framework[9] has a directory for each translation direction (e.g. `en-cs`) and a subdirectory for each test set (`batch1a`, `batch2a`, `batch3a`, `batch4a`, `news`). Replicating the Pilot 3 results (after installing Treex as described in D2.10) is easy:

```
git clone https://github.com/ufal/qtleap
cd qtleap/translate/en-cs/batch4a/
make translate eval D="optional description describing this experiment"
# Each experiment has a number, eg. 42 and is stored in runs/042_<date>.
make help # see a list of commands
# Now, copy the experiment #42 to the qtleap-corpus repository
make archive-042
cd ../../../qtleap-corpus/
git status
git commit -a
git push
# After few minutes, the results will be automatically evaluated
# and stored in the QTLeap Evaluation Workbench.
```

2.2 TectoMT: English Components

This section details English-specific features of the TectoMT pipeline, used for all language pairs within the TectoMT framework. The pipeline is a very slightly updated version of the Pilot 2 pipeline described in D2.8; therefore, we only give here a very brief overall explanation.

2.2.1 Analysis

The English analysis follows the annotation pipeline used for the CzEng 1.6 parallel corpus [Bojar et al., 2016b], using a (rule-based) tokenizer, a statistical part-of-speech tagger [Spoustová et al., 2007] and dependency parser to a-trees [McDonald et al., 2005a], followed by mostly rule-based post-processing.

The t-layer conversion starts from the a-tree and follows the process outlined in Section 2.1.3 very closely (see also D2.4 and D2.8 for details); there have been no significant changes in English analysis since Pilot 2.[10]

[8] https://github.com/ufal/treex

[9] https://github.com/ufal/qtleap/tree/master/translate

[10] We have implemented a new detector of functors using VowpalWabbit. It can be easily switched on by using `functors=VW` parameter of the scenario. We observed intrinsic improvements in the quality of functor assignment [Bojar et al., 2016b]. However, we have not observed significant improvements in the translation quality, so we have not integrated this into the final Pilot 3.

2.2.2 Synthesis

The English synthesis pipeline also adheres to the general setup presented in Section 2.1.6 (see also D2.4 and D2.8 for details). The block for adding definite and indefinite articles `T2A::EN::AddArticles` is now based on language-independent `T2A::AddArticles` and it relies on tectogrammatical grammateme definiteness. So it is a responsibility of transfer to fill this grammateme. The original rule-based code for guessing definiteness has been moved to en→cs transfer and it Pilot 3, it has been substituted by a machine-learning-based solution described in Section 2.4.2.

2.3 Basque: TectoMT

2.3.1 Analysis

The handling of Basque formemes has been modified, allowing the definition of *complex postpositions*, that is, postpositions made up by a postpositional suffix attached to the previous word and one or more words (*-en aurrean* 'in front of'). For that purpose, we marked the case identifier between square brackets and kept the words as they are. Therefore, the *-en aurrean* postposition is coded as 'n:[gen]+aurrean' according to the new formeme definition. The list of *complex postpositions* that are identified for Basque is based on Arriola [2012] and consists on 156 different combinations.

In Basque, subordination is usually expressed by a subordinating suffix, which can also be combined with one or more words (*-n bitartean* 'while'). The new formeme definition treats them in the same way as complex postpositions ('v:[erl]+bitartean')

Additionally, rule-based blocks have been defined to fix some recurrent errors in the analysis tools: Bad lemmatization of numbers, incorrect parsing of modal verb constructions.

2.3.2 Transfer

The changes made in the analysis (new definition of the formemes and the differences coming from the rule-based blocks) forced us to re-train all the translation models.

Additionally, we have included a rule-based block to translate relative pronouns from English to Basque. In Basque, subordination is expressed by a suffix added to the subordinated verb and usually there is no need of any pronoun. However, Basque lacks an equivalent suffix for *where* when used as a subordinating conjunction. Instead, the periphrasis *-en lekuan* 'in the place that' is used.

2.3.3 Synthesis

The rules used to place the postpositional suffixes have been refined. In Basque, only the last word of the linguistic phrase is inflected with definiteness, number and case information of the whole phrase. The hand-made rules used to determine the end of the phrase and the word that has to be inflected have been redefined.

Moreover, the synthesis has also been adapted to the changes made on the definition of the Basque formemes. This allows the system to create new words from the definition of the formeme just after the word that is inflected.

Finally, the generation of verb forms has been redefined, dividing the generation in two blocks. The first is devoted to the generation of the modal auxiliary particles, and the second focuses on the generation of the auxiliary verbs according to the verbal tense.

2.4 Czech: TectoMT

2.4.1 Analysis

The Czech analysis is almost unchanged since the last Pilot.[11] Therefore, we only give here a very brief description of the whole pipeline. Please refer to D2.4 and D2.8 for more details.

The analysis pipeline is based on the annotation pipeline of the CzEng 1.0 and CzEng 1.6 corpora [Bojar et al., 2012, 2016b], starting with a rule-based tokenizer and a statistical part-of-speech tagger [Straková et al., 2014] and dependency parser [McDonald et al., 2005b, Novák and Žabokrtský, 2007]. The a-trees produced by the parsers are then converted to t-trees using a rule-based process which follows very closely the description in Section 2.1.3.

2.4.2 Transfer

The en→cs transfer now uses VowpalWabbit and online-learning domain adaption (Section 2.1.5) instead of the MaxEnt models and TM interpolation (in-domain + general-domain) used in Pilot 2. Most rule-based transfer blocks are still in place from Pilot 2.

There are two main improvements to cs→en transfer in Pilot 3: statistical definiteness assignment and a fixed tense transfer rule for the IT-domain.

Statistical definiteness assignment. We created a new, statistical module for definiteness detection for the Czech-to-English transfer. Czech has no notion of definiteness and only uses implicit means to express it (and no definiteness grammateme is used in Czech t-layer), whereas in English, expressing definiteness is obligatory and determiners (articles or pronouns) must be used on the surface, which translates to the values of the definiteness grammateme on the English t-layer. Pilot 1 and Pilot 2 used an older rule-based module of Ptáček [2008] to assign definiteness to translated t-nodes; its performance was rather poor. We trained the VowpalWabbit linear classifier [Langford et al., 2007] using a feature set based on phenomena that influence definiteness and article usage. The set is partially based on Ptáček [2008]'s rules and consists of the following feature types:

- grammateme values of gender (in personal pronouns), number, negation of the current t-node
- diathesis grammateme of the current t-node's parent (if verb)
- t-lemma and formeme values (including parts of formemes, i.e., coarse part-of-speech, prepositions/conjunctions and syntactic position) in both source and target languages, as well as in the syntactic and topological neighborhood of the current node
- the current t-node's topological position relative to its parent as well as its distance from the parent; the same information for the source Czech t-node
- indicator values for the current t-node preceding the main verb in its clause

[11] Similarly as in English, we have implemented a VowpalWabbit-based assignment of functors, but have not integrated this into the final Pilot 3.

- indicator values of the current node being a pronoun, having a pronominal determiner, pre-modifier, relative clause, or further any specification (attribute)
- indicator values for the current node representing a meal, ocean, island, mountain, nation, and similar semantic groups with specific definiteness behavior
- indicator value for the current t-node representing a proper noun
- the current t-node's countability (also combined with grammatical number and other selected properties)
- preceding occurrence of the same t-lemma in the local context window (up to 30 topologically preceding t-nodes)
- all preceding t-lemmas in the local context window

We trained the VowpalWabbit classifier on the CzEng 1.0 corpus Bojar et al. [2012] in the cost-sensitive one-against-all setting with hinge loss with 4 passes and 24-bit feature space. Intrinsic classifier accuracy on the development section of CzEng was 93.72%. We did not perform any extensive feature tuning and the CzEng corpus does not match the QTLeap domain. Therefore, there is still room for improvement, but the new statistical method still yielded a gain of up to 1.1 BLEU point in comparison to the older rule-based module on the QTLeap corpus batches.

Tense transfer rules. Based on our observations on QTLeap Batch 1 and 2 corpora, we noticed that in the IT domain, the future tense in Czech most often translates to simple present tense in English. Therefore, we introduced a simple grammateme rule to transfer Czech future to English present tense. This results in a gain of around 0.5 BLEU points across the QTLeap batches and it has not been found to shift meaning or produce unintelligible results in the IT domain.

2.4.3 Synthesis

The Czech synthesis pipeline required no changes since Pilot 2 thanks to extensive testing and tuning during several years of TectoMT operation [Popel and Žabokrtský, 2009]. Please see Section 2.1.6 and D2.4 for more details.

2.5 Dutch: TectoMT with Alpino embedded

As before, the Pilot 3 system for the Dutch analysis and Dutch synthesis components consist of TectoMT pipelines, with Alpino embedded in both directions. In the Dutch analysis, no important improvements have been implemented for Pilot 3. Minor improvements in the lexical components of Alpino may have led to minor improvements in the analysis accuracy. Similarly, no major improvements have been made in transfer since Pilot 2. The translation models have been retrained on newly analyzed training corpora (Europarl v7, KDE corpus, Dutch Parallel Corpus), which resulted in minor BLEU score gains.

In contrast to analysis and transfer, quite some effort has been spent on the improvement of the Dutch synthesis component. As before, a number of rule-based blocks in TectoMT take the output of transfer, t-trees, and convert these to "abstract dependency structures", of the type expected by the Alpino synthesis module. The Alpino synthesis module provides a mapping of “abstract dependency structures” to surface strings, taking care of word order, agreement and inflection. For this purpose, Alpino incorporates a large attribute-value grammar (also used for syntactic analysis) and a large dictionary. Furthermore, a statistical fluency component based on Maximum Entropy is used to select the most natural and fluent realization for a given input structure.

In cases where the input dependency structure precisely corresponds to the dependency structure assigned by the grammar to a given utterance, the synthesis component works very well. In the translation set-up, however, the input for synthesis is often somewhat different. In such cases, synthesis must be extended to solve this mismatch. For earlier pilots, a number of manually defined transformation rules were implemented which take an “unexpected” input dependency structure and map it to a dependency structure for which synthesis is expected to provide a good realization. Furthermore, a number of heuristics is implemented which ensure that if a dependency structure cannot be realized by the synthesis component, the synthesis component is applied to each of the parts of the structure, concatenating each of the partial realizations afterwards.

For Pilot 3, some of the transformations have been improved, and the number of such transformations has been extended quite a lot, by careful manual inspection of the results of the synthesis component on the Batch1 set of answers. Before we started to implement improvements for Pilot 3, the set of answers for Batch1, when translated with the Pilot 2 system, led to 1461 dependency structures for which synthesis should provide an utterance. Thus, the Alpino synthesis component is called 1461 times. In the Pilot 2 version, because of various mismatches between the input structure and the expected structure, the synthesis component actually produced 5654 partial results (in some cases, a single input structure led to a single output result; in other cases a single input structure led to multiple results because of the robustness strategy described in the previous paragraph). Using the new, extended set of transformations, this number dropped from 5654 to 4637. This indicates that the synthesis component more often was able to generate a result for larger input structures – and in the large majority of cases this leads to an improvement in the final output as well.

Currently, there are 391 transformations, ranging from fairly generic ones (typically for particular syntactic constructions, e.g., to ensure that subject control is specified in the correct way for subject control and raising verbs), to transformations for particular English expressions (for example, “if so” should not be translated as “als het is” but is now translated as “zo ja”), transformations for context-sensitive translations (for example, “check for” must be translated as “controleren op” and not “controleren voor”), correcting neuter/non-neuter determiners, adding prepositions in particular cases (“press X” should not be translated as “druk X”, but as “druk op X” in Dutch), correcting typical translation mistakes in case of compounds (“programming language” is translated as “programmeertaal taal” by the transfer component), and domain-specific translations (“driver” should not be translated as “chauffeur” in the context of computer software).

2.6 Portuguese: TectoMT

2.6.1 Translation between Portuguese and English

Like for the previous MT Pilots, and for the other language pairs, given the real usage scenario against which the project was mostly developed (cf. Deliverables D3.6, D3.10 and D3.12), the direction pt→en was aimed at supporting information retrieval from the QA database whose question/answer pairs are recorded in the pivot language, i.e. English;

and the en→pt direction was aimed at supporting outbound translation thus supporting the delivery of the answer retrieved in the user's language, i.e. Portuguese.

Reasonable scores for the retrieval step were obtained already with the initial MT Pilots (cf. Deliverables D3.6 and D3.10). Concomitantly, it is in the outbound direction that quality translation has a paramount impact. Accordingly, for the development of Pilot 3, being reported here, and its evolution out of the previous Pilot 2, the direction pt→en received a few adjustments, while the bulk of the attention was devoted to improve the en→pt direction.

In what concerns the pt→en direction of Pilot 3, the English synthesis module is the same module as used for the Pilots 3 of other languages pairs in the project with TectoMT, viz. Basque, Czech, Dutch and Spanish (cf. Section 2.2 above).

The Portuguese analysis module, in turn, received some improvements targeting the filtering out of some specific and systematic translation errors emerging from some unwanted interaction of the blocks, which helped to improve with regard to the previous BLEU score of Pilot 2.

2.6.2 Analysis and Synthesis

In what concerns the direction en→pt, in turn, details are provided in this and the next subsections below.

The English analysis module is the same module as used for the Pilots 3 of other languages pairs in the project with TectoMT, viz. Basque, Czech, Dutch and Spanish (cf. Section 2.2 above).

The Portuguese synthesis module is basically the same module as used for the previous MT Pilot whose eventual improvements and expansion leading to Pilot 3 version resulted from the successful experiments undertaken in the workpackages WP4 and WP5. While these experiments are described at length, respectively, in Deliverables D4.13 and D5.11, and in the publications whose references are provided therein, an outlook of the more practical aspects of their development are provided below.

2.6.3 Terminology

The blocks with treelets concerning Microsoft Collection Terminology (MTC) were adapted to the Portuguese language pair by downloading the Microsoft terminology for Portuguese and pre-processing these data and converting them from their original XML format to the TSV format.

A blacklist was also created by manually analyzing the most frequent translation errors, as several cases were found where the terminology consistently leads to a wrong choice of the target equivalent for the relevant term. The top 5 most frequent such terms were added to the blacklist, which resulted in a list of 15,748 terms in total, leading to a further improvement of the BLEU score.

The use of these MTC blocks improved the translation in 0.33 BLEU points (using batch 2).

We have also experimented with enriching MTC terms with an additional terminological collection coming from the FREME project,[12] increasing the total list of terms to 16,637 entries. The BLEU score obtained with this extended terminology list, though, happened to be below the score achieved with MTC alone.

[12] http://www.freme-project.eu/

2.6.4 Word Sense Disambiguation

Building on the previous results of experiment 5.4.1, and further exploiting their potential, the work on Pilot 3, described in [Neale et al., 2016], showed that machine translation can be improved by incorporating word senses as contextual features in a maxent-based translation model.

Training these models over a large, open domain corpus, we have obtained statistically significant improvements in BLEU score when compared to a baseline version of our machine translation system. This demonstrates that including word sense information as features can increase the likelihood of pairings between words and phrases occurring in the translation model.

2.6.5 Multiword Expressions

Following the work described in Deliverable D5.7, the analysis of multiword expressions (MWE) in the TectoMT system was extended to cover the en→pt translation direction.

The automatic acquisition of MWEs resulted in a list of 111,351 Portuguese and 551,253 English expressions.

The source language analysis pipeline was prepared to address all MWEs with a compositionality threshold value of 0.2. In a subsequent step we trained a translation model with these MWEs.

Resorting to batch 2, and with the best Pilot 2 system plus the MWEs, the resulting translation pair was evaluated. A positive delta was obtained in terms of BLEU score, which represented however a non-significant improvement.

Additionally, when this block for MWEs was active together with the block for MTC, the BLEU score was found to have a decrease. Accordingly, in the final version of this pipeline, the block for MWEs was eventually not used.

2.7 Spanish: TectoMT

2.7.1 Analysis

The Spanish analysis is almost unchanged since the last Pilot. Therefore, here we only give a very brief description of the whole pipeline. Please refer to D2.4 and D2.8 for further details.

The analysis pipeline is based on the IXA pipes tools.[13] So far, we have used Treex tokenization, IXA pipes modules for POS tagging and lemmatization, and Mate tools for dependency parsing. Next, in the rule-based pipeline, the dependency trees are converted to a-layer-compatible trees and finally to t-trees.

2.7.2 Transfer

On es→en transfer we have experimented with a number of modifications that we finally decided not to include in Pilot 3. Those modifications are directly related to WP5 and will be presented in detail in D5.11.

[13] http://ixa2.si.ehu.es/ixa-pipes/

Sense aware VowpallWabbit transfer. We used the new VowpalWabbit transfer module (Section 2.1.5) to integrate the semantic information obtained by a WSD system. We inspected a number of features including the WordNet synset and the Super Sense tag of the word that had to be translated as well as its parent and siblings. However, we did not observe any significant improvement (the best configuration was 0.06 BLEU points below the MaxEnt transfer used in Pilot2). Therefore, we decided to discard the new transfer module and use the transfer developed for Pilot2. Please refer to D5.11 form further details.

Automatic gathering of specialized terminology. The use of specialized terminology can drastically improve the domain translations as we observed in the development of Pilot2. The use of gazetteers to translate domain specific terms brought an improvement of 3.19 BLEU points. Here, we have tried to automatically extend those gazetteers, which were originally extracted from collaborative hand-made resources (localization dictionaries and Wikipedia articles). We gathered parallel terminology from comparable texts, and included them in the TectoMT systems. We incorporated the new terminology dictionaries as gazetteers (which do not allow morphological inflection). The use of the automatically gathered terminology brings an improvement over the systems without domain terminology, but the improvement is smaller than the one obtained by the dictionaries already used in Pilot2. The use of the new terminology in combination with the dictionaries already used in Pilot2 does not yield any gain in comparison to Pilot2. Therefore, we decided to keep the original configuration for Pilot3 as well. Please refer to D5.11 for further details.

2.7.3 Synthesis

The Spanish synthesis is almost unchanged since the last Pilot. The whole generation process is divided into three steeps: (1) conversion of t-trees into a-trees, (2) morphological generation and (3) linearization into plain text. All the synthesis process is rule-based, except for the morphological generation, which is based on Flect [Dušek and Jurčíček, 2013]. Please refer to D2.4 and D2.8 for further details.

2.8 Bulgarian: Deep factored MT

Bulgarian ↔ English Pilot 3 systems build on Pilot 1 (described in D2.4) and Pilot 2 (described in D2.8). Pilot 3 presents a hybrid machine translation system consisting of three main steps (depicted in Figure 1 and further described in Sections 2.8.1–2.8.3). The source-language text is linguistically annotated, then translated with the Moses system to the target language and post-processed using the linguistic annotation projected from the source side.

During the translation with the Moses system the word alignment is stored in order to be used for the projection of the linguistic analyses from the source text to the target text.

It is important to mention that the number of the tokens in the source and the target language might differ. Also, the alignments can include many-to-many correspondences, not just one-to-one. Nevertheless, in practice about 80 % of the alignments are one-to-one or two-to-two tokens.

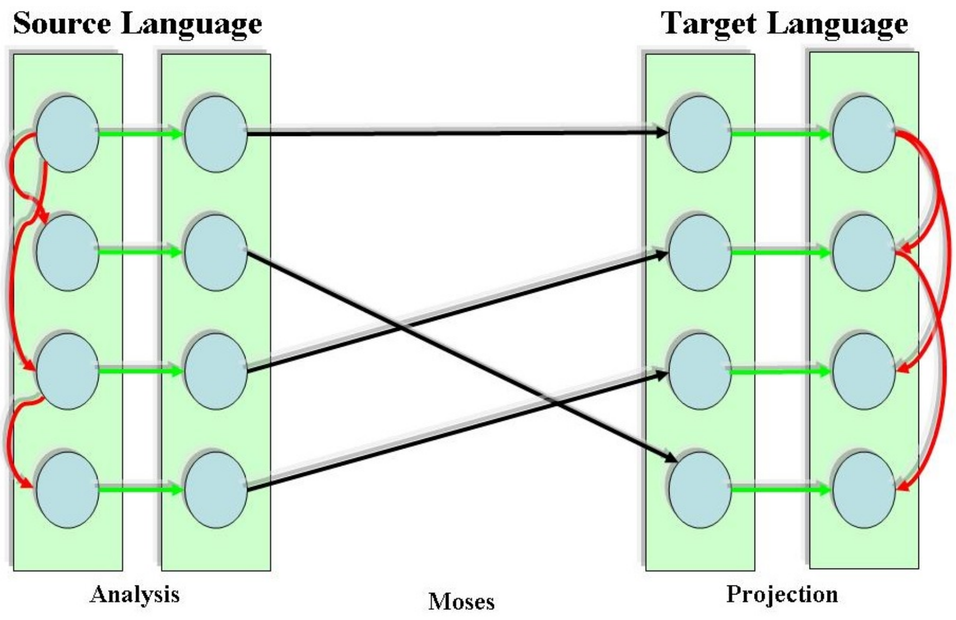

Figure 1: A hybrid architecture of bg↔en Pilot 3 for transferring linguistic information from the source to the target language. The linguistic analyses for the source language (Analysis - column 1) are projected to a tokenized source text (Analysis - column 2); then the Moses models (Moses) are applied for producing a target language translation. The translation alignment (Projection - column 1) is used for transferring the information to the corresponding tokens in the target language (Projection - column 2). The projected linguistic information interacts with the linguistic features of the tokens in the target text (for example the morphosyntactic features). Finally, the resulting annotation of the target text is used for post-processing.

Here is an example of aligned texts annotated with morphosyntactic information of the English[14] sentence "Place them in the midst of a pile of dirty, soccer kit." and its translation into Bulgarian[15]:

```
(place/VB them/PRP in/IN) = (postavyaneto/Ncnsd im/Ppetdp3;Ppetsp3;Pszt--3 v/R)
(the/DT midst/NN of/IN)   = (razgar/Ncmsi na/R)
(a/DT pile/NN of/IN)      = (kup/Ncmsi)
(dirty/JJ)                = (izmyrsyavam/Vpitf-r1s)
(,/,)                     = (,/Punct)
(soccer/NN)               = (futbolni/A-pi)
(kit/NN)                  = (komplekt/Ncmsi)
(./.)                     = (./Punct)
```

From the alignment and rules for mappings between the two tagsets we could establish the following alignments on token level:

```
(them/PRP)    = (im/Ppetdp3;Ppetsp3;Pszt--3)
(in/IN)       = (v/R)
```

[14] For English, the tagset of Pen treebank is used: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html.

[15] For Bulgarian, the tagset of BulTreeBank is used: http://www.bultreebank.org/TechRep/BTB-TR03.pdf.

```
(midst/NN)    = (razgar/Ncmsi)
(of/IN)       = (na/R)
(pile/NN)     = (kup/Ncmsi)
(,/,)         = (,/Punct)
(soccer/NN)   = (futbolni/A-pi)
(kit/NN)      = (komplekt/Ncmsi)
(./.)         = (./Punct)
```

Additionally, the alignment `(place/VB) = (postavyaneto/Ncnsd)` would be possible because the noun `(postavyaneto/Ncnsd)` is a deverbal noun, derived from a verb `(postavyam/Vpitf-r1s)`. To establish such an alignment we would need a derivational lexicon which however is not available to us. Thus, we do not consider this type of alignment. Likewise, the alignment between the English adjective `(dirty/JJ)` and the Bulgarian verb `(izmyrsyavam/Vpitf-r1s)` would be also possible as much as "something to be dirty" could be a result from the action denoted by the verb. We consider such rules also quite unreliable and thus we do not have such alignment rules.

On the basis of these alignments, we were able to transfer additional information like dependency links, word senses and elementary predicates. It is clear from the example that the transfer is only partial. The alignment `(soccer/NN) = (futbolni/A-pi)` is allowed because of the fact that they form a compound (see below).

After the transfer of additional information, a set of rules for post processing are applied. For example, here a rule for agreement between the adjective `futbolni` and the noun `komplekt` has been applied. More details are given in D4.13.

For the en→bg translation, we extended the above architecture by adding another Moses model. Our goal was to reuse the Pilot 2 setup (see Deliverable 5.7).

The motivation for using the representative lemma in the target language is our expectation for unification of the various synset IDs with the similar translations in the target language. For example, in the en→bg direction, the two concepts referred by *donor*: `wn30-10025730-n` ("person who makes a gift of property") and `wn30-10026058-n` ("a medical term denoting someone who gives blood or tissue or an organ to be used in another person") are very close to each other. They have the same translation in Bulgarian in both corresponding synsets: *донор*. The representative word is selected on the basis of a frequency list of Bulgarian lemmas constructed over large corpora (70 million words).

As an example, the procedure we performed with respect to the training, testing and tuning of the Moses system is as follows:

English sentence:
This is real progress .

English sentence with factors:
this|this|dt is|be|vbz реален|real|jj напредък|progress|nn .|.|.

Bulgarian sentence with factors:
това|това|pd е|съм|vx реален|реален|a напредък|напредък|nc .|.|pu

Bulgarian sentence:
Това е реален напредък.

We selected this Moses model for en→bg because in the earlier versions of Pilot 3 it performed slightly better than the phrase-based model.

The architecture for en→bg is depicted on Figure 2. Here the source language is analyzed linguistically, then the tokenized text is processed in two ways in order to produce the text for the Source/Target text (in the example above it corresponds to **English sentence with factors**). First, the replacements with the Bulgarian lemmas were done on the basis of Word Sense annotation of the source text. Additionally we translated the source text with phrase-based Moses model (Moses1).

From this translation we selected some words to be used as factors. The idea was to enrich S/T text with more target-language factors. Here it is important to keep in mind that the number of tokens in the S/T text is the same as in the source text. Thus, the analyses produced for the source text are easy to transfer to the S/T text.

Then the actual translation was done with factor-based Moses model (Moses2) where the alignment is used for the projection of the linguistic analyses over the source text.

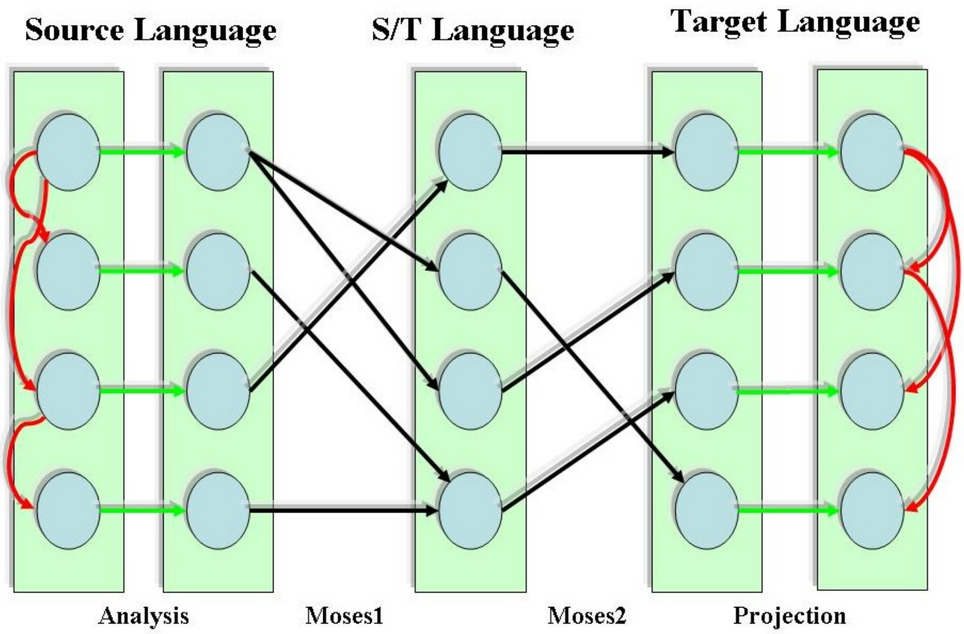

Figure 2: A hybrid architecture of en→bg Pilot 3 for transferring linguistic information from the source to the target language. The linguistic analyses for the source language are projected to a tokenized source text; then Moses models (Moses1 and Moses2) are applied for producing a target language translation. The translation alignment is used for transferring the information to the corresponding tokens in the target language. Finally, the target linguistic annotation is used for post-processing.

Our work on the projection of linguistic analyses from the source to the target text is similar to Ramasamy et al. [2014] and Mareček et al. [2011].

2.8.1 Analysis

For the en→bg direction the source-language linguistic annotation consists of tokenization, POS tagging, lemmatization, dependency parsing, Minimal Recursion Semantics annotation and word sense disambiguation.

The analysis of English (tokenization, lemmatization, POS tagging and dependency parsing) as a source language was done with the CoreNLP tools[16] of Stanford University. The word sense disambiguation was done by the UKB tool.[17] The MRS structures and

[16] http://stanfordnlp.github.io/CoreNLP/

[17] http://ixa2.si.ehu.es/ukb/

the post-processing rules were implemented in the CLaRK System.[18]

For the analysis of Bulgarian as a source language, we trained Mate tools[19] on the Bulgarian treebank.

In order to adapt the processing to the domain, we have annotated Batch1 and Batch2 with morphosyntactic information.

2.8.2 Transfer

In the en→bg system, we use a two-step translation strategy.

The first step (Moses1) is done using a phrase-based Moses model. We have used the following parallel data: SETimes parallel corpus, LibreOffice parallel corpus, Bulgarian English Dictionary aligned on wordform level, Microsoft product descriptions and Microsoft Terms.

The text were tokenized with the tokenizers for the corresponding languages.

We trained a phrase-based Moses model using the following options `-alignment grow-diag-final-and` and `-reordering msd-bidirectional-fe`.

The language model used is a 5-gram language model trained with SRILM on the data from SEtimes corpus, LibreOffice corpus, domain articles from Wikipedia, Microsoft data, and the Bulgarian National Reference Corpus (mainly news data and fiction).

The tuning was done on Batch1a of the QTLeap corpus.

The second step (Moses2) includes a factor-based Moses model (similar to the en→bg Pilot 2 system) which starts from a partially translated source language (S/T language).

In S/T language, some of the source tokens were replaced with target-language lemmas using the results from the WSD of the source language. The result from step one was used to extend the S/T-language text with lemmas for words that are not translated via WSD.

The training was done on the same parallel data but processed as in the example given above on page 24. Both the source (**English sentence with factors**) and the target (**Bulgarian sentence with factors**) texts were processed with the corresponding language pipelines. The options used in training the factor-based model are `--translation-factors 0,2-0,2+1,2-0,2`, `--decoding-steps t0:t1`, `-alignment grow-diag-final`, and `-reordering distance`.

The factors are `SWF-TL|SL|STag` for the input text and `TWF|TL|TTag` for the output. `SWF-TL` denotes source language word form or target language lemma, if the pipeline established correspondence for the input source word form. `SL` denotes the lemma for source language word form, `STag` denotes the POS tag for source language word form. `TWF`, `TL`, and `TTag` denote target language word form, lemma, and POS tag.

The language model used is a 5-gram language model trained with SRILM on the data from SEtimes corpus, LibreOffice corpus, and the Bulgarian National Reference Corpus (mainly news data and fiction).

The tuning was done on Batch1a of QTLeap corpus.

For some functional words (where we are sure about the alignment), the source language word form was replaced with the target language lemma from the translation pro-

[18] http://www.bultreebank.org/clark/index.html

[19] http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/matetools.en.html

type of change English→Bulgarian	frequency	example
noun1 noun2→noun1 noun2	rare	business meeting→бизнес среща
noun1 noun2→noun2 noun1	frequent	Rila mountain→планина Рила
noun1 noun2→adj1 noun2	frequent	antivirus software→антивирусни програми
noun1 noun2→noun2 prep noun1	frequent	email settings→настройки за поща

Table 1: Examples of structural changes in translation of English noun-noun phrases into Bulgarian.

duced by Moses1 model. The idea is to have as much as possible substituted source language word form in S/T language text.

For the bg→en system, we are using the same parallel corpora and options as for the phrase-based model for en→bg, but the whole transfer is done in one step.

For the language model we are using the SEtimes corpus, LibreOffice corpus, domain articles from Wikipedia, Microsoft data, and Europarl corpus.

The tuning was done on Batch1q of the QTLeap corpus.

2.8.3 Post-processing

The post-processing is a rule-based system that includes linguistically-enhanced information. This information is projected from the source side with the help of the word alignments produced by Moses1 and Moses2. The projected information is the linguistic knowledge in the form of MRS-based elementary predicates, labeled dependencies, word senses (synset ids from WordNet) and POS tags in the source language.

It should be noted that the alignment between the source language to the S/T language and from the S/T language to the target language is not one-to-one. It generally maps sets of tokens from the source language to sets of tokens in the target language. Thus, as it was presented above in the example, the transfer of the linguistic information from the analysis of the source language to the target one is not straightforward. Here we apply heuristic rules. Thus, the transferred linguistic information is only partial. For the rules definition we also exploited the language resources and tools for the target language — a morphological lexicon, a lemmatizer, and a morphological generator.

Once the linguistic annotation is projected via the alignment, the post-processing can be applied. It includes various types of rules: morphological, syntactic and semantic.

An example of a syntactic rule is the transformation of the English noun compounds into the appropriate syntactic structures in Bulgarian. The different templates are represented in Table 1. The direct transfer is rare, since the NN compounds are not so frequent in Bulgarian. The combination in which the first noun is a Named Entity is the most frequent one in the domain data. In the case of a phrase with an adjective and a noun in Bulgarian, a morphological rule for agreement is applied.

More details about the post-processing are presented in D4.13.

2.8.4 Summary of improvements since Pilot 2

The extensions with respect to Pilot 1 and Pilot 2 include:

- improved knowledge graphs[20] for the UKB system,
- extension of the parallel data with aligned terminology and multi-word expressions,
- rules for generation of Minimal Recursion Semantics structures,
- rules for transfer of linguistic information from source to target text and
- post-processing rules that were implemented manually.

2.9 German: Quality systems combination

Our overall en→de Pilot 3 hybrid architecture "Qualitative" includes:

- a phrase-based SMT baseline system (Moses, as described in Pilot 0)
- an improved version of transfer-based system of Pilot 1 (Lucy)
- a neural MT system, and
- an informed selection mechanism ("ranker").

The architecture is illustrated in Figure 3 and the different components are described below. The new component added for Pilot 3 is the neural MT system. Towards the end of QTLeap, neural MT had become a hot topic driven on the one hand by announcements of large companies and at the same time by good performance of academic NMT systems, e.g., in WMT 2016 [Bojar et al., 2016a]. The QTLeap project therefore decided to dedicate a few person months for the experiment of setting up an NMT system for German, trained on the same data as the German Pilot 0.

For the "inbound direction" de→en of the QTLeap usage task (see, e.g., D3.12), we have used only the neural MT system as we were interested to what extent today's neural technology can outperform Moses on this cross-lingual information retrieval task. For en→de, we have integrated the neural MT system into the hybrid Pilot 3 architecture described below.

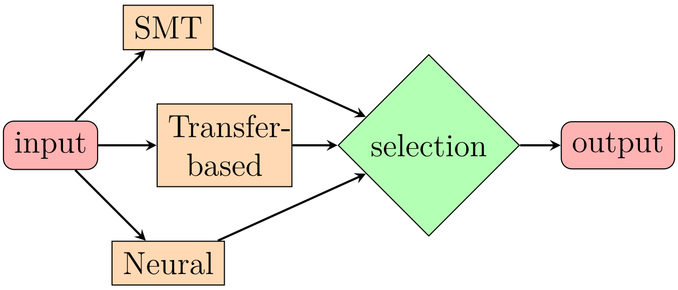

Figure 3: Architecture of the Pilot 3 selection mechanism

[20] http://www.bultreebank.org/QTLeap/

	BLEU	METEOR	manual
baseline	24.90	44.38	
quotes	24.00	44.29	
sepMenus	25.39	45.01	
sepMenus + normPunct	25.41	45.06	15.8%
sepMenus + normPunct - WhereItSays	25.36	45.00	84.2%
SMTmenus	24.06	42.83	
unk	24.50	44.05	
unk + sepMenus	23.68	43.30	
unk + SMTmenus	25.41	44.95	
unk + SMTmenus - WhereItSays*	25.36	44.88	

Table 2: Improvements on the transfer-based system. (*) Indicates the variant used in Pilot 3.

2.9.1 Improved transfer-based component

The transfer-based system Lucy [Alonso and Thurmair, 2003] is also part of our experiment, due to its state-of-the-art performance in the previous years. Additionally, manual inspection on the development set has shown that it provides better handling of complex grammatical phenomena particularly when translating into German, due to the fact that it operates on transfer rules from the source to the target syntax tree.

Pilot 3 work on the transfer-based system focused on issues revealed through manual inspection of its performance on the development set:

- **Separate menu items**: The rule-based system was observed to be incapable of handling menu items properly, mostly when they were separated by the ">" symbol, as they often ended up as compounds. We identified the menu items by searching for consequent title-cased chunks before and after each separator. These items were translated separately from the rest of the sentence, to avoid them being bundled as compounds. The rule-based system was then forced to treat the pre-translated menu items as chunks that should not be translated.
- **Menu items by SMT**: Additionally, we used the method above to check whether menu items could be translated with the baseline SMT system instead of Lucy.
- **Unknown words by SMT**: Since Lucy is flagging unknown words, we translated these individually with the baseline SMT system.

Finally, we experimented with normalization of the punctuation (which was previously included in the pre-processing steps of SMT but not in the transfer-based), addition of quotes on the menu items and some additional automatic source pre-processing in order to remove redundant phrases such as "where it says".

We ran exhaustive search with all possible combinations of the modification above and the most indicative automatic scores are shown in Table 2. Although automatic scores have in the past shown low performance when evaluating transfer-based systems, our proposed modifications have a lexical impact that can be adequately measured with n-gram based metrics. Our investigation and discussion is performed on Batch 2. The best combination of the suggested modifications achieves an overall improvement of 0.51 points BLEU and 0.68 points METEOR over the baseline. In particular:

- Adding quotes around menu items resulted in a significant drop of the automatic scores, so it was not used; this needs to be further evaluated, as references do not use quotes for menu items either. Nevertheless, quotes were not always useful due to an occasional erroneous identification of menu item boundaries.

- Separate translation of the menu items (sepMenus) gives a positive result of about 0.49 BLEU and 0.63 METEOR.

- Normalizing punctuation (normPunct) has a slightly positive effect when the menu items are translated separately by Lucy.

- Passing only unknown words (unk) to SMT results in a loss of 0.4 BLEU.

- Translating the menus with SMT (SMTmenus) also deteriorates the scores.

- Translating both menu items and unknown words with SMT (unk+SMTmenus) has a positive effect against the baseline and it seems to be comparable with the best system without SMT (sepMenus+normPunct).

The phrase “where it says” appears in 7% of the sentences in Batch 2 and 2% of the sentences in Batch 1. Although the removal of “where it says” in the source sentence seems to slightly lower the automatic scores, the difference does not seem significant, and manual inspection raised the concern that this may be because of the way this phrase has been translated in the references. We therefore conducted manual sentence selection on 38 (out of the 69) sentences where this phrase appeared and in 84.2% of the cases its removal made the translation preferable. We therefore concluded in selecting this variation, despite the slightly lower scores.

2.9.2 Neural MT system

Our Neural MT algorithms follow the description of [Bahdanau et al., 2014]. The input sequence is processed using a bidirectional RNN encoder with gated recurrent units (GRU) [Cho et al., 2014] into a sequence of hidden states. The final backward state of the encoder is then projected and used as the initial state of the decoder. Again, our decoder is composed of an RNN with GRU units. In each step, the decoder takes its hidden state and the attention vector (a weighted sum of the hidden states of the encoder, computed separately in each decoding step), and produces the next output word.

In addition to the attention model, we use byte pair encoding (BPE) [Sennrich et al., 2015] in the preprocessing step. This ensures that there are no out-of-vocabulary words in the corpus and, at the same time, enables for open-vocabulary decoding.

We trained our model on the same data as the phrase-based SMT baseline system. We used Batch 1 for validation during the training. In the experiments, the sentence length was limited to 50 tokens. The size of the hidden state of the encoder was 300 units, and the size of the hidden state of the decoder was 256 units. Both source and target word embedding vectors had 300 dimensions. For training, batch size of 64 sentences was used. We used dropout and L2 for regularization.

Our model was implemented using Neural Monkey,[21] a sequence-to-sequence learning toolkit built on top of the TensorFlow framework [Abadi et al., 2016]. This toolkit was used before by Libovický et al. [2016] in the submission of WMT-2016’s multimodal translation and automatic post-editing tasks.

[21] http://github.com/ufal/neuralmonkey

2.9.3 Selection mechanism

The three systems above are combined with a selection on the sentence level. For every source sentence, the output of every available system is analyzed with several automatic NLP techniques to produce numerical values which indicate some aspects of quality. Out of the numerical values, we form one feature vector which represents the qualitative characteristics of every produced translation output. Consequently, we employ an empirical mechanism which aims to *rank* and *select* given these feature vectors.

Machine Learning The core of the selection mechanism is a ranker which reproduces ranking by aggregating pairwise decisions by a binary classifier [Avramidis, 2013]. Such a classifier is trained on binary comparisons in order to select the best one out of two different MT outputs given one source sentence at a time. The binary comparisons are aggregated per system and the winner is the system which wins the most pairwise comparisons. The approach of pairwise comparisons is chosen because it poses the machine learning question in a much simpler manner. Instead of treating a whole list of ranks, the classifier has to learn and provide a binary (positive or negative) answer to the simple question "*which of these two sentences is better?*". This also provides the flexibility of experimenting with many machine learning algorithms for the classification, including those which only operate on binary decisions.

There are often cases where the classifiers where two systems win an equal number of pairwise comparisons. As a result, the selection mechanism fails to express any quality preference between the two systems. If such is the case for the first ranked translation, the system combination reaches a point of not being able to select one single translation, i.e. two systems win the same amount of the pairwise comparisons so it is not clear which one is the best. In order to eliminate these cases, we weigh each pairwise comparison with its confidence score (soft pairwise recomposition) [Avramidis, 2013]. In Pilot 3, as compared to Pilots 1 and 2, the employed algorithms waived out ties entirely.

As training material, we use the test-sets of WMT evaluation task (2008–2014) for German-English, which consist of 25385 human judgments of various MT outputs which can be decomposed to 338627 pairwise training instances. Contrary to Pilots 1 and 2, where we used automatic reference-based metrics as rank labels, the rank labels for the training of Pilot 3 are given by human annotators, as part of the WMT evaluation campaign. One tenth of the data, 2542 judgements including 9883 pairwise comparisons is separated and used for testing the algorithms. The results for the top algorithms were later confirmed with a 10-fold cross-validation over the entire dataset.

We exhaustively tested the feature vectors of Pilot 2 on a shorter development set with many machine learning methods including Gaussian Naïve Bayes, k-nearest Neighbors (kNN), Logistic Regression, Linear Discriminant Analysis (LDA), Extremely Randomized (ExtRa) Trees, Random Tree Forests, Bagging Classifiers, AdaBoost over 50 single decision trees and Gradient Boosting over 100 single decision trees. The models produced were scored in terms of correlation with the original human ranks with Kendall's tau. The scoring was confirmed using a cross-validation with 10 folds over the entire amount of WMT data. The results are summarized in Table 3, which reports Kendall's tau (tau, our primary evaluation metric), its empirical confidence intervals, p-value, Expected Reciprocal Rank (ERR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG).

The indicated p-value indicates the significance test of Kendall tau correlation, based on investigating the null hypothesis that *there is no correlation between the two sets*

learner	tau	+/-	p-value	ERR	MRR	NDCG
Decision Tree	0.012	0.014	0.068	0.425	0.381	0.627
kNN	0.187	0.013	$< 10^{-4}$	0.541	0.406	0.696
Random Tree Forests	0.187	0.014	$< 10^{-4}$	0.555	0.441	0.704
ExtRa Trees	0.188	0.014	$< 10^{-4}$	0.580	0.478	0.717
Bagging	0.221	0.014	$< 10^{-4}$	0.576	0.464	0.718
Gaussian Naïve Bayes	0.241	0.014	$< 10^{-4}$	0.557	0.432	0.709
LDA	0.250	0.014	$< 10^{-4}$	0.581	0.454	0.723
Logistic Regression	0.258	0.013	0	0.560	0.456	0.710
AdaBoost	0.265	0.013	0	0.597	0.467	0.732
Gradient Boosting	0.276	0.014	0	0.595	0.468	0.731

Table 3: Comparison among learning algorithms

of ranks, i.e. the rank produced by the system and the corresponding rank produced by the human annotator. Because in our development set there may be ranking sets with different length k, *inverse-variance weighing* [Hartung et al., 2008] was applied on the mean, in order to calculate the significance over all the distributions. According to Kendall's theory, $\bar{\tau}$ is approximately following the normal distribution under the null hypothesis and therefore the z-test was used to test the significance with a two-tailed test. According to Kendall's theory, continuity correction would also be required, but given a high n, as is the case in our experiments, this is not needed.

As a conclusion, based on Kendall's tau, the best correlation was given using Gradient Boosting. This has better performance than the Linear Discriminant Analysis, which was used for Pilots 1 and 2.

Feature selection The feature selection is applied on the test-sets of WMT evaluation task (2008–2014) for English-German, which consist of 19980 human judgments of various MT outputs. Contrary to Pilots 1 and 2, where we used automatic reference-based metrics as rank labels, the rank labels for the training of Pilot 3 are also given by human annotators, as part of the WMT evaluation campaign.

The feature vector was selected from a broader range of 139 features using Recursive Feature Elimination with cross validation (RFECV), applied on a proportion of the original training data explained above. In particular, since RFECV is computationally intensive, we performed stratified sub-sampling to keep three sets, using 1%, 2.5% and 5% of the original amount of sentences. These resulted in feature sets of 23, 26 and 56 features respectively.

These 3 feature sets were then used to train the ranking model on the entire set of training data and to test it with 10-folded cross validation. The performance of the three sets with the two best learning algorithms is compared in Table 4. The feature set with 56 features is performing the best, according to all ranking metrics (Kendall's tau, ERR, MRR, NDCG). Based on the empirical confidence intervals of Kendall's tau rank correlation metric, using Gradient Boosting with the set with 56 features is significantly better than using the set with 23 features.

The iterative cross-validation process of selecting features can be seen at Figure 4. RFECV concluded to a set of 56 distinct features including:

- **Parse probabilities**: the number of feasible k-best parse trees, the highest and the

learner	#features	tau	+/-	p-value	ERR	MRR	NDCG
AdaBoost	23	0.132	0.015	< 0.005	0.611	0.506	0.720
	26	0.153	0.014	< 0.005	0.620	0.518	0.728
	56	0.158	0.014	< 0.005	0.618	0.514	0.728
Gradient Boosting	23	0.142	0.015	< 0.005	0.614	0.510	0.724
	26	0.165	0.014	< 0.005	0.620	0.517	0.731
	56	0.175	0.014	< 0.005	0.625	0.521	0.734

Table 4: Comparison among the three features sets that resulted from the Recursive Feature Elimination

lowest probability in the k-best parse tree list, the mean and the standard deviation of the parse probabilities in the k-best parse tree list

- **Parse nodes**: the distance of main and subordinate VPs from the end of the target sentence, the count and the average position of nouns in the sentence, the count and the standard deviation of the positions of NPs, PPs and VPs in the sentence, the average and maximum height of VPs in the parse tree, count of target NPs aligned with source NPs via IBM model 1

- **Punctuation and case**: count, average position and standard deviation of commas, count of dots, uppercase sentence start

- **Contrastive scores**: BLEU and METEOR using the rest two systems as references

- **Language modeling**: 5-gram language model probability,

- **IBM model 1**: the IBM model 1 scores on both directions, and their ratios, thresholded by either 0.01 and 0.2

- **Baseline features**: the baseline features of WMT12.

The features can be reproduced using the tool Qualitative as described in Avramidis [2016], which has been presented as an open source tool, as part of Pilot 3.

3 Chimera Systems

Chimera [Bojar et al., 2013b, Bojar and Tamchyna, 2015] is one of several approaches how to combine a deep-linguistic MT system (TectoMT) with a phrase-based system (Moses). See Rosa et al. [2016a] for an overview of related approaches.

The main idea is simple: TectoMT is used to provide additional training data for Moses. The development set and the test set (and optionally other in-domain data) are translated by TectoMT. This way we obtain a parallel corpus with genuine source-language side and synthetic target-language side. A secondary phrase table is extracted from this corpus. This is then used together with the primary phrase table, extracted from the large training data, to train Moses. Finally, the input is translated by the resulting Moses system.

This setup enables Moses to use parts of the TectoMT translation that it considers good, while still having the base large phrase table at its disposal. This has been shown to

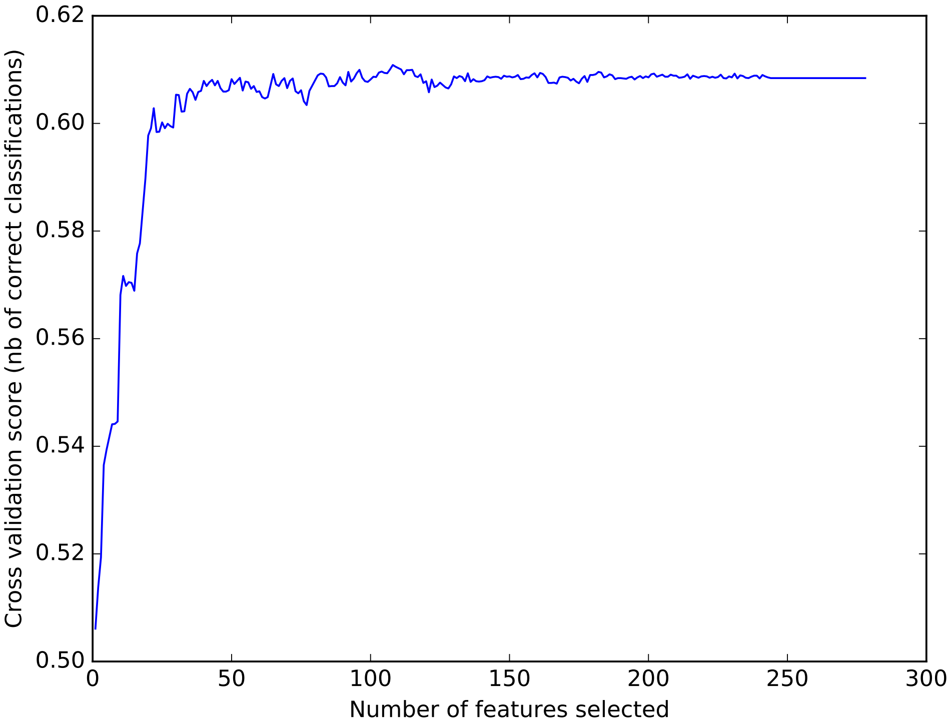

Figure 4: Plotting the Recursive Feature Elimination via Cross Validation that was performed in order to find the optimal amount of features.

have a positive effect, e.g., in choosing the correct inflection of a word when the language model encounters an unknown context, or in generating a translation for a word that constitutes an out-of-vocabulary item for Moses (as TectoMT can abstract from word forms to lemmas and beyond, which Moses cannot).

Optionally, the output can be improved with Depfix automatic post-editing [Mareček et al., 2011, Rosa, 2014]. We have skipped this step because we do not have Depfix adapted for all the QTLeap languages that participated into the TectoMT framework.

Chimera was the winning system of the WMT English-to-Czech translation task in the years 2013, 2014 and 2015 [Bojar et al., 2013a, 2014, 2015] both in automatic (BLEU) and manual evaluation. We have successfully used Chimera for the WMT 2016 IT-domain translation task [Bojar et al., 2016a], where we also applied a dictionary-based (gazetteer) domain adaptation [Rosa et al., 2016b]. This domain-adaptation improved the BLEU scores for all tested languages (CS, ES, NL and PT)[22] when applied on top of Moses or TectoMT (which confirms our previous experiments with domain adaptation reported in D2.8[Popel et al., 2015]). If applied on top of Chimera it helped only for ES, NL and PT, but not for CS, which was confirmed by the WMT manual evaluation.

Another conclusion of Rosa et al. [2016b] is that using the in-domain data as additional phrase table is more effective than using it as gazetteer for forced translation, but it requires retraining the system (which is not always possible or convenient). In our case, we had to retrain the Chimera system anyway (so it uses the newest version of Pilot 3), so we decided to add all batches (with TectoMT translations) to the in-domain phrase table. To summarize it, P3-Chimera is a Moses trained with two phrase tables (and two

[22] en→eu Chimera was prepared only after the WMT IT-task, so it is not listed here, but it is included in the evaluation in Section 4.

sets of parameters for MERT training):

- first phrase table is extracted from the (big, general-domain) parallel corpus, used for training Pilot 0 (e.g. Europarl, see D2.2 for details)
- second phrase table consists is extracted from much smaller, but in-domain training data, namely:
  - Batches 1a, 2a, 3a and 4a with TectoMT translations[23]
  - gazetteers (LibreOffice, KDE, VLC, Microsoft Terminology, see D5.7)

We have also tried an alternative setup with three phrase tables, where Batch1a has its own phrase table and a set of parameters and uses the gold reference translations instead of TectoMT translations. However, this alternative setup resulted in worse results (−0.4 BLEU) in preliminary en→cs experiments, probably because there were too many parameters for MERT. So we used the setup with two phrase tables for the final Pilot3 Chimera system.

The above is the general Pilot 3 Chimera setup, which is language independent. Except for the additional in-domain phrase table and the exceptions listed below, the Chimera setup (tokenization, true-casing, reordering models, language models etc.) is the same as for Pilot 0.

Spanish and Basque Chimera differs from the setup described above just slightly: it does not directly include the gazetteers for training of the second phrase table. The same gazetteers are indirectly included as they are used in the Pilot3 systems.

Czech Chimera uses Operation Sequence Model [Durrani et al., 2015] and new version of CzEng (1.6).

4 Intrinsic Evaluation

This section describes the intrinsic evaluation of the Pilot 3 results, starting with automatic measures (Section 4.1) and then describing a manual evaluation study (Section 4.2). The presentation of intrinsic evaluation is completed in deliverable D3.12, where further human evaluation results are presented in the more convenient context of that report on the online evaluation forms and procedures.

4.1 Automatic evaluation results

The main test corpus for the evaluation of Pilot 3 is QTLeap Batch 4 (Section 4.1.1). We have also measured out-of-domain performance on the QTLeap News corpus (Section 4.1.2). Similarly to the automatic evaluation of previous Pilots in D2.4 and D2.8, scores have been computed using the official BLEU and NIST script `mteval-v13a.pl --international-tokenization` and F-MEASURE using `rgbF.py` as implemented in the QTLeap Evaluation Workbench.

[23] Batch4a is the final test set. Note that we are not using Batch4a reference translations for training (that would be cheating), we need just the source side (English).

For BLEU and F-MEASURE, the best system in each column is marked in bold if it is significantly ($p < 0.05$, using bootstrap resampling) better than the remaining systems; otherwise, we mark in bold the smallest set of best-scoring systems such, that no other system is significantly better. For NIST, we did no significance test and we just mark the best system in bold if it is at least 0.1 better that the second-best system; otherwise, we mark the smallest set of systems which are at least 0.1 better than the rest of the systems.

It is interesting that all the three automatic measures almost always agree on the ordering of the systems. There are no cases of “significant disagreement”, where acording to one metric systemA would be significantly better than systemB, but according to another metric systemA would be significantly worse than systemB. Thus in the following discussion, we cite the BLEU scores only, but as can be seen in Tables 5–8, F-MEASURE and NIST follow the same pattern.

4.1.1 QTLeap Batch 4 results

Like for the previous MT Pilots, given the real usage scenario against which the project was mostly developed (cf. Deliverables D3.6, D3.10 and D3.12), the direction X→EN was aimed at supporting information retrieval from the QA database whose question/answer pairs are recorded in the pivot language, i.e. English; and the EN→X direction was aimed at supporting outbound translation thus supporting the delivery of the answer retrieved in the user’s language.

Hence the quality of the automatic translation was a much more pressing desideratum for the project in the EN→X direction, than in X→EN. Accordingly, this is reflected in the different efforts devoted to the two directions, and thus in the overall difference in the scores between Table 5 and Table 6.

Translation into English

system	metric	bg→en	cs→en	de→en	es→en	eu→en	nl→en	pt→en
Pilot0	BLEU	18.54	**20.53**	**34.74**	**26.88**	**13.70**	27.89	**13.75**
Pilot1	BLEU	17.04	17.19	33.01	12.19	3.72	23.08	8.56
Pilot2	BLEU	19.08	**19.86**	–	17.68	6.27	21.99	9.54
Pilot3	BLEU	**24.93**	**21.31**	24.51	18.07	7.30	**30.34**	9.72
Pilot0	F-measure	25.85	27.73	**39.74**	**33.20**	**21.89**	34.37	**21.59**
Pilot1	F-measure	24.72	26.07	38.37	20.97	12.04	29.89	17.81
Pilot2	F-measure	25.47	28.16	–	26.01	15.79	28.61	19.29
Pilot3	F-measure	**30.64**	**29.03**	30.73	26.31	16.69	**36.03**	19.46
Pilot0	NIST	5.7602	6.0332	**7.8231**	**6.6810**	**5.1590**	6.8487	**4.7746**
Pilot1	NIST	5.5562	5.8754	7.4645	4.9157	3.6770	6.3154	4.1777
Pilot2	NIST	5.4762	6.3105	–	5.7072	4.3653	6.0702	4.5581
Pilot3	NIST	**6.4078**	**6.5461**	6.0316	5.7944	4.5610	**7.1891**	4.6280

Table 5: BLEU, F-MEASURE and NIST scores of Pilot0 (baseline), Pilot1, Pilot2 and Pilot3 on translations into English of Batch4q (questions) part of the QTLeap Corpus.

For the translation into English (Table 5), the baseline Pilot 0 has been outperformed by Pilot 3 for three language pairs (bg→en, cs→en, nl→en) according to the automatic measures. The evaluation reported in D3.12 also confirms these improvements.

We can see a progress from Pilot 1 to Pilot 2 and to Pilot 3 for almost all language pairs except for de→en, where an experimental Neural MT system was used as Pilot 3 as discussed in Section 2.9 and in D3.12. There are notable improvements from Pilot 2 to Pilot 3 for some language pairs, e.g. +8.35 BLEU for nl→en or +5.85 BLEU for bg→en.

Translation from English

system	metric	en→bg	en→cs	en→de	en→es	en→eu	en→nl	en→pt
Pilot0	BLEU	20.30	23.17	**34.90**	24.11	**17.94**	25.42	12.01
Pilot1	BLEU	18.86	19.81	31.54	15.02	9.46	19.54	12.88
Pilot2	BLEU	16.42	21.49	29.47	24.89	10.93	20.60	13.87
Pilot3	BLEU	**23.91**	24.24	31.12	24.94	11.24	22.35	15.33
Chimera	BLEU	–	**26.16**	–	**35.36**	17.16	**26.65**	**19.64**
Pilot0	F-measure	25.92	29.34	**39.46**	29.60	**24.10**	31.00	18.19
Pilot1	F-measure	24.57	26.46	36.19	22.52	16.00	26.52	20.24
Pilot2	F-measure	22.54	28.09	34.20	31.41	17.13	27.39	21.55
Pilot3	F-measure	**29.05**	30.49	35.51	31.46	17.60	29.02	22.73
Chimera	F-measure	–	**31.66**	–	**40.16**	23.46	**32.12**	**26.43**
Pilot0	NIST	5.4974	6.4614	**7.8698**	6.3439	**5.5301**	6.7286	4.4412
Pilot1	NIST	5.3066	6.1446	7.2668	5.2517	4.0337	6.2335	5.0005
Pilot2	NIST	4.9562	6.4085	6.9162	6.9075	4.2000	6.3686	5.2510
Pilot3	NIST	**5.9615**	**6.7644**	7.0254	6.9157	4.3130	6.6271	5.4685
Chimera	NIST	–	**6.8319**	–	**7.9426**	5.3964	**6.9139**	**6.0770**

Table 6: BLEU, F-MEASURE and NIST scores of Pilot0 (baseline), Pilot1, Pilot2, Pilot3 (DeepFactoredMoses for BG, Qualitative for DE, TectoMT for CS, ES, EU, NL and PT) and Pilot3-Chimera on translations from English of Batch4a (answers) part of the QTLeap Corpus.

For the translation from English (Table 6), the baseline Pilot 0 has been outperformed for 5 language pairs (en→bg, en→cs, en→es, en→nl, en→pt) according to the automatic measures.

Basque is a less-resourced language and though the QTLeap project has contributed greatly to improve on this situation with the datasets and processing tools curated for this language, the results obtained for the Basque MT Pilots cannot be taken as a surprise and should rather be understood as a further stimulus to keep looking to improve the language technology for this language. Basque Chimera obtained significantly better BLEU than Pilot 3 (17.16 vs. 11.24), but it still did not outperform Pilot 0 (17.94). This confirms our hypothesis that Chimera cannot help in cases when TectoMT is substantially worse than Moses (more than 6 BLEU points).

German Pilot 3 has significantly lower BLEU score than Pilot 0 (34.90 vs. 31.12), but the extrinsic evaluation in D3.12 shows mixed results with positive cost reduction as measured by the low probability of calling an operator.

For Dutch, Pilot 3 (TectoMT) is worse than Pilot 0 in BLEU (25.42 vs. 22.35), but again not in the extrinsic evaluation in D3.12. Moreover, the Dutch Chimera (26.65) is the best system for Dutch. This shows that even if TectoMT is more than 3 BLEU points worse than Moses, this difference is not substantial from the combination point of view and thus Chimera (combination of the two systems) can help.

For Bulgarian, we can see an improvement of +3.61 BLEU points of Pilot 3 over Pilot 0 (and +7.49 BLEU points over Pilot 2).

For the rest of the languages (Czech, Spanish and Portuguese), Pilot 3 outperformed Pilot 0, and Chimera outperformed Pilot 3.

In general, for all the language pairs where Chimera was developed (en→cs, en→es, en→pt, en→nl and en→eu), it significantly outperformed all the QTLeap Pilots 1, 2 and 3 with margins (BLEU improvements of Chimera over Pilot 3): +1.92 for en→cs, +10.42 for en→es, +4.31 for en→pt, +4.30 for en→nl and +5.92 for en→eu.

4.1.2 QTLeap News corpus results

The results of QTLeap Pilots in the News domain are generally lower than in the IT domain because there was no specific tuning for the News domain. We have just switched off the adaptations specific for the IT domain. Still, we can see that Chimera significantly outperformed Pilot 0 for en→cs and en→es.[24]

system	metric	bg→en	cs→en	de→en	es→en	eu→en	nl→en	pt→en
Pilot0	BLEU	**18.05**	**24.03**	**26.26**	**27.53**	**8.84**	**23.50**	**21.85**
Pilot1	BLEU	17.72	12.07	17.77	8.78	2.40	11.33	6.69
Pilot2	BLEU	17.30	13.04	–	13.55	3.07	19.40	7.55
Pilot3	BLEU	15.44	14.39	10.04	14.34	3.19	19.68	7.84
Pilot0	F-measure	**24.50**	**30.06**	**31.79**	**33.37**	**16.11**	**29.49**	**28.32**
Pilot1	F-measure	24.12	20.36	24.65	18.15	8.61	19.07	15.89
Pilot2	F-measure	23.88	21.45	–	22.55	9.90	26.17	16.75
Pilot3	F-measure	22.33	22.29	17.44	23.07	10.15	26.39	16.96
Pilot0	NIST	**5.9261**	**6.4104**	**7.1936**	**7.1269**	**4.2403**	**6.6055**	**6.2846**
Pilot1	NIST	**5.8483**	5.3041	5.7891	4.7929	2.6788	5.3089	4.3250
Pilot2	NIST	**5.8320**	5.4994	–	5.5867	3.0465	6.3691	4.4653
Pilot3	NIST	5.7496	5.6301	3.4526	5.7073	3.1334	6.4121	4.5242

Table 7: BLEU, F-MEASURE and NIST scores of Pilot0 (baseline), Pilot1, Pilot2 and Pilot3 (NeuralMonkey for DE Pilot3) on translations into English on the QTLeap News Corpus.

4.1.3 Discrepancies of automatic scoring

During the development of Pilot 3, we became aware of severe discrepancies between the automatic scoring and the human evaluation, whereas the latter, as per standard practice is considered the gold standard. Such discrepancies have been noted also in the past [Callison-Burch et al., 2006], with the automatic scores of transfer-based systems being

[24] For en→eu, Chimera achieved almost the same results as Pilot 0, but this may be attributed to the fact that en→eu TectoMT (Pilot 3) was much worse that Moses (Pilot 0), so Chimera learned to ignore the secondary phrase table extracted from TectoMT translations. However, we observed many differences in the output of Chimera and Pilot 0 (only 130 sentences out of 1000 are identical), so we cannot confirm this hypothesis based on the output.

system	metric	en→bg	en→cs	en→de	en→es	en→eu	en→nl	en→pt
Pilot0	BLEU	**15.45**	17.57	**17.41**	29.96	**5.36**	**19.66**	**21.13**
Pilot1	BLEU	14.48	12.40	16.73	9.24	2.09	12.72	8.64
Pilot2	BLEU	14.65	14.36	12.89	13.60	2.10	13.72	7.60
Pilot3	BLEU	13.15	14.74	15.56	13.60	1.82	13.56	8.03
Chimera	BLEU	–	**19.71**	–	**32.21**	**5.43**	19.08	11.52
Pilot0	F-measure	**21.93**	23.63	**23.99**	34.73	**11.95**	**26.10**	**27.68**
Pilot1	F-measure	21.00	19.08	23.49	17.62	7.41	20.15	17.31
Pilot2	F-measure	21.15	21.23	19.21	22.49	7.84	21.35	16.47
Pilot3	F-measure	19.84	21.47	21.59	22.48	8.02	21.50	16.10
Chimera	F-measure		**25.87**		**37.06**	**11.95**	25.52	17.99
Pilot0	NIST	**4.9192**	5.5785	**5.9356**	7.4626	**3.3667**	**6.2240**	**6.3710**
Pilot1	NIST	4.7159	4.8418	5.8066	4.6941	2.1151	5.4337	4.6801
Pilot2	NIST	4.7254	5.2770	4.8378	5.6849	2.2584	5.7047	4.2903
Pilot3	NIST	4.4757	5.3038	5.2765	5.6847	2.4010	5.7513	4.3015
Chimera	NIST	–	**5.9834**	–	**7.8329**	**3.3309**	6.0501	4.4448

Table 8: BLEU, F-MEASURE and NIST scores of Pilot0 (baseline), Pilot1, Pilot2, Pilot3 (DeepFactoredMoses for BG, Qualitative for DE, TectoMT for CS, ES, EU, NL and PT) and Pilot3-Chimera on translations from English on the QTLeap News Corpus.

heavily underestimated when compared against statistical MT, particularly in languages with complex syntax and reordering (see for example the performance of German RBMT systems in the shared translation task of WMT2009 Callison-Burch et al. [2009]).

We hereby include an example of an extension of our en→de Pilot 2, which was used as a preliminary step for Pilot 3. During the development phase, we submitted to the WMT Shared Task for the IT-domain, our original transfer-based system of Pilot 1 and its two best variations for Pilot 3 (see section 2.9.1), along with Pilot 0 and the selection mechanism of Pilot 2 [Avramidis et al., 2016b]. As part of the shared task, the translated sentences were scored by BLEU, but were also evaluated by several human annotators, following the standard ranking evaluation of WMT. An excerpt of the official results is given in Table 9.

	rank	TrueSkill	BLEU
(another system)	1		
Transfer-based SMTmenus	2–6	−0.062	25.4
Transfer-based baseline	3–6	−0.093	25.2
Transfer-based menus	3–6	−0.098	25.2
(other systems)	7–8		
Pilot 2 selection mechanism	9	−0.382	29.0
SMT baseline	10	−0.485	34.0

Table 9: Human ranks and automatic scores of our submitted systems on the tests, as a result of the official evaluation. Ranks are given in a range in order to account for confidence intervals.

A comparison of the evaluation scores suggests that whereas the system "Transfer-based SMT menus" is the best performing among our systems, reaching rank positions 2–6 in the human evaluation (TrueSkill rank), it only gets a BLEU score of 25.4 points. This is 8.6 points lower than the SMT baseline, which only gets the 10th position according to the humans, although it had the highest BLEU score. There is a similar discrepancy for the Pilot 2 selection mechanism, which humans seem to find significantly better than the SMT baseline, although it scores 5 BLEU points less. These observations justify the use of further manual evaluation, which is being discussed in the following chapters.

4.2 Manual evaluation methodology

In QTLeap, evaluation is taken very seriously – a whole work package is devoted to the integration and evaluation of MT technology in a real use-case scenario. Yet, we have to keep in mind that manual evaluation in WP 2 reported below is performed voluntarily by partners, so we had to make sure that the effort is kept moderate. In past evaluations, we have experimented with several different tools and methods for more detailed automatic and manual evaluation including Hjerson and MQM (cf. Aranberri et al. [2016]). Supported by the reviewers at the second review meeting, we decided to try another manual evaluation technique for the final evaluation that provides qualitative (and on a small scale quantitative) insights into the performance of the system. In line with a general strategy to include language experts in the MT development cycle described in Burchardt et al. [2016], we have performed a detailed *source-driven* error analysis using a dedicated "test suite".

Test suites are a best-practice instrument in areas such as grammar checking, to ensure that a parser is able to analyze certain sentences correctly or test the parser after changes to see if it still behaves in the expected way. In the context of MT, we use the term "test suite" to refer to a selected set of input-output pairs that reflects interesting or difficult, error-prone cases. Test suites have not generally been used in MT research. Reasons for this might include the theoretical issue that there is no eternal notion of "good translation" and the more practical issue that there are usually many different good translations for a given input. Even if one could assume the existence of some gold-standard translation, there would be no simple notion of deviation that could be used. In the QT21 project[25], DFKI is constructing an expansive test suite (English ↔ German) containing a wide range of various linguistic phenomena that provides a basis for manual analyses in different contexts.

Inspired by the performance of the German QTLeap system components on the test suite, we have constructed a small domain-specific test suite based on examples from the QTLeap corpus that represent interesting linguistic phenomena. The "linguistic phenomena" are understood in a pragmatic sense and cover various aspects that influence the translation quality. Therefore, our phenomena include morpho-syntactic and semantic categories as well as formatting issues, issues of style, etc.

It is important to note that we are only counting selected errors in this scenario, namely the ones related to the respective test item.

Starting with the evaluation of our contribution to the WMT2016 IT task (Avramidis et al. [2016a]), we have by now developed an efficient manual evaluation process, performed by a professional linguist. This procedure consists of the following steps:

1. The linguist has a close look at the output of the different MT systems and identifies systematically occurring translation errors that are related to linguistic phenomena.

[25] www.qt21.eu

2. For each of these linguistic phenomena that seem to be prone to translation errors, up to 100 segments containing the phenomenon in the source language are extracted.

3. For each phenomenon, the total occurrences in the source language are counted.

4. Consequently, the total occurrences in the outputs of the different MT systems are counted.

5. The accuracy of the MT outputs for the phenomena is measured by dividing the overall number of correctly translated instances by the overall number of instances in the source segments.

The phenomena that we found to be prone to translation errors in QTLeap context were **imperatives**, **compounds**, **menu item separators** (">"), **quotation marks**, **verbs**, and **terminology**.

For the selected six linguistic phenomena, 600 English source segments were extracted from Batch 2 of the QTLeap corpus. In those source segments, 2015 instances of the different phenomena were found overall, as it was often the case that more than one instance occurred per segment.

As there may always be several correct translations, an occurrence of a phenomenon is not only counted as correctly translated when it matches the reference translation but also when it is for example realized in a different structure that correctly translates the meaning. The following example demonstrates the manual evaluation technique for German on several different MT systems:

(A)	source:	Yes, type, for example: 50 miles in km.	*1 inst.*
	Pilot 0:	Ja, Typ, zum Beispiel, 50 Meilen in km.	*0 inst.*
	NMT:	Ja, Typ, beispielsweise: 50 Meilen in km.	*0 inst.*
	Lucy:	Tippen Sie zum Beispiel, ja: 50 Meilen in km.	*1 inst.*
	reference:	Ja, geben Sie, zum Beispiel: 50 Meilen in km ein.	

In example A, the source segment contains one imperative: "type". A correct German translation needs to have the right verb from + the personal pronoun "Sie" in this context. In most of the cases, the imperative "type" is mistranslated as the German noun "Typ" instead of the verb "tippen" or "eingeben", e.g., in the Pilot 0 and Neural MT output. The Lucy system on the other hand correctly translates the imperative. Note that the reference translation contains the phrasal verb "eingeben" and due to the imperative construction the suffix "ein" moves to the end of the sentence.

4.3 Manual evaluation results

As described above, the evaluation methodology has been first tested on German. Consequently, the selected error types were those relevant to the German engines, where 100 segments per phenomenon have been inspected. After presentation of the results, all partners have volunteered to repeat this manual inspection for their languages on at least 20 segments. It has been decided that partners would use the same error classes (as far as possible) to see if they also show differences for the systems working in different languages. Below, we will report on the individual findings.

4.3.1 Basque

	#	Pilot 0	Pilot 3
imperatives	43	100%	98%
compounds	34	59%	44%
">" separators	20	100%	100%
quotation marks	80	85%	95%
verbs	94	74%	66%
terminology	90	39%	53%
sum	361		
average		71%	73%

Table 10: Translation accuracy on manually evaluated sentences in Basque focusing on particular phenomena. Test-sets consist of hand-picked source sentences of Batch 2 that include the respective phenomenon.

The linguistic features studied for Basque show diverging results for Pilot 0 and Pilot 3. Imperatives and ">"-separators behave very similarly in both systems. As can be seen in Example A, both Pilot 0 and Pilot 3 address imperatives correctly. Note that the lemmas used by the systems and the reference are different but nonetheless valid and the verb is correctly formed using the lemma and the suffix *-tu*.

(A) source: Try going to Encoding and choose one that has UTF8.
Pilot 0: Saiatu joan kodeketa eta aukeratu duena UTF8.
Pilot 3: Saiatu Kodeketa joan da eta aukeratu UTF8a duen bakarrik.
reference: Zoaz Kodifikazioa atalera eta hautatu UTF8 duena.

Example B shows the performance of Pilot 0 and Pilot 3 while dealing with ">"-separators. As we can see, both systems correctly place the separators between the relevant UI strings.

(B) source: Yes, go to Tools ">" Word Count.
Pilot 0: Bai, joan Tresnak ">" hitz kopurua.
Pilot 3: Bai joan Tresnak ">" Hitz zenbaketa.
reference: Bai, zoaz Tresnak ">" Hitzak Zenbatu atalera.

The translation quality of compounds and verbs decreases slightly when using Pilot 3 compared with the performance of Pilot 0. We detected that a number of compounds were incorrectly translated by Pilot 3 mainly because external elements were introduced between the elements of the compound (see Example C). This is probably because the compound was not identified as such during analysis and therefore it was split to form the final t-level tree.

(C) source: Click the right mouse button on the chart and then go to Chart Data Table[...]
Pilot 0: Egin klik saguaren eskuineko botoiarekin diagraman eta ondoren joan diagrama-datuen taula[...]
Pilot 3: Egin klik saguaren diagrama botoi eskuinerantz eta ondoren joan Diagrama-datuen taula[...]
reference: Sakatu saguaren eskuin botoiarekin grafikoaren gainean eta ondoren joan Grafikoaren Datuen Taula[...]

Although the difference is small, Pilot 0 translates verbs better than Pilot 3. Example D below shows how one of the imperatives *select* - *hautatu* is incorrectly translated with progressive aspect and with a subordinate suffix. It is possible that these features were incorrectly spread in the tree from the verb *says* - *dioen* in the previous relative clause.

(D) source: Perform a search, and then click the button that says Filters and then select the Channel option.
Pilot 0: Bilaketa bat egiteko, eta gero egin klik botoian dioen iragazkiak eta hautatu kanala aukera.
Pilot 3: Egin bilaketa bat eta ondoren egin klik botoia u Iragazkiak dioen, ondoren Kanala aukera hautatzen den eta.
reference: Egin bilaketa bat eta sakatu Iragazkiak dioen botoian eta hautatu Kate aukerak:

However, Pilot 3 addresses quotation marks and terminology better than Pilot 0. Pilot 3 is able to correctly identify and transfer the position of quotation marks, whereas Pilot 0, relying on local information alone, often misses the position of the marks (see Example E).

(E) source: Click on the button beneath the Google+ logo, select the option Settings, and change where it says, “Who can send you notifications?”
Pilot 0: Egin klik botoian Google + logotipoaren azpian, hautatu aukera ezarpenak, eta aldatu eta han dioenez, “nor” bidal diezazuke jakinarazpenak?
Pilot 3: Egin klik Googler+eko logotipo botoiaz, hautatu aukera Ezarpenak eta aldatzen da “nork zure jakinarazpenak bidaliz” dioen tokian.
reference: Egin klik Google+ logoaren azpian dagoen botoian eta hautatu Ezarpenak aukera eta aldatu “Nork bidal diezazuke jakinarazpenak?” aukera.

Terminology is also better handled by Pilot 3 as compared to Pilot 0. Example F below shows an case where three out of the four terms in the source sentence where correctly translated by Pilot 3 and none by Pilot 0. Pilot 3 includes modules that specifically try to deal with terminology and therefore this might be the reason why it is performing better than the statistical system.

(E) source: Click the gear at the top of the page and select the option Settings. On the new page that appears, click on the tab that says “Email notifications.”

Pilot 0: Egin klik engranaje, orriaren goialdean, eta hautatu aukera ezarpenak. Orri berria agertzen den, egin klik fitxa “dioen jakinarazpen elektronikoa.”

Pilot 3: Egin klik orri goialdean gear eta hautatu Ezarpenak aukera. Agertzen den orri berrian egin klik “Helb. el. jakinarazpenak” dioen fitxa.

reference: Egin klik orriaren goiko aldean dagoen gurpilean eta hautatu Ezarpenak aukera. Agertzen den orri berrian, egin klik “Posta elektronikoen jakinarazpenak” dioen tokian.

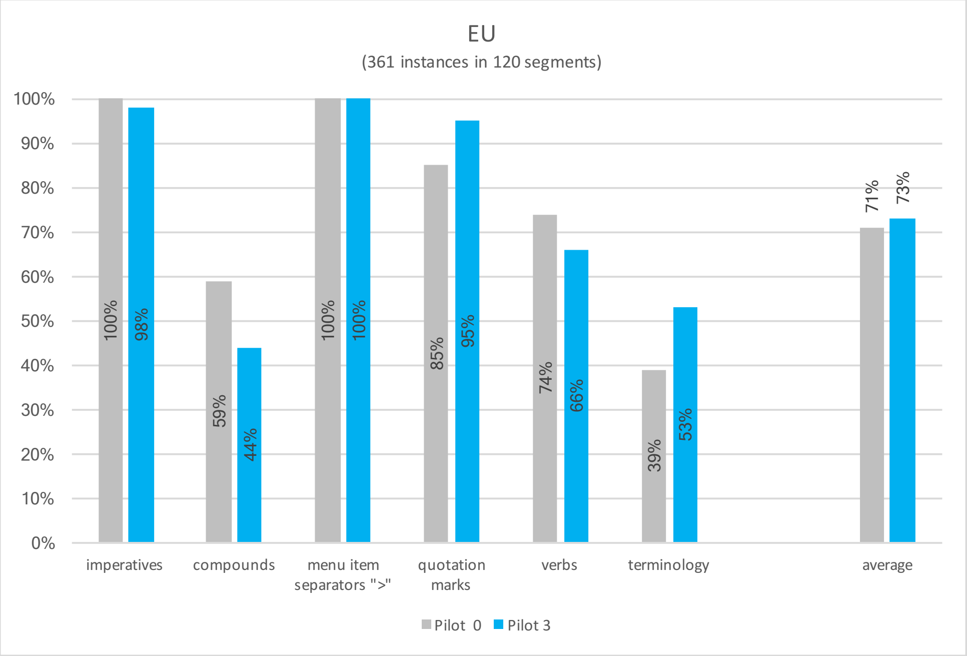

Figure 5: Manual evaluation results of Basque

4.3.2 Bulgarian

	#	Pilot 0	Pilot 3
imperatives	97	74%	68%
compounds	100	44%	35%
“>” separators	60	100%	98%
quotation marks	200	90%	69%
verbs	221	78%	73%
terminology	153	67%	60%
sum	831		
average		76%	56%

Table 11: Translation accuracy on manually evaluated sentences in Bulgarian focusing on particular phenomena. Test-sets consist of hand-picked source sentences of Batch 2 that include the respective phenomenon.

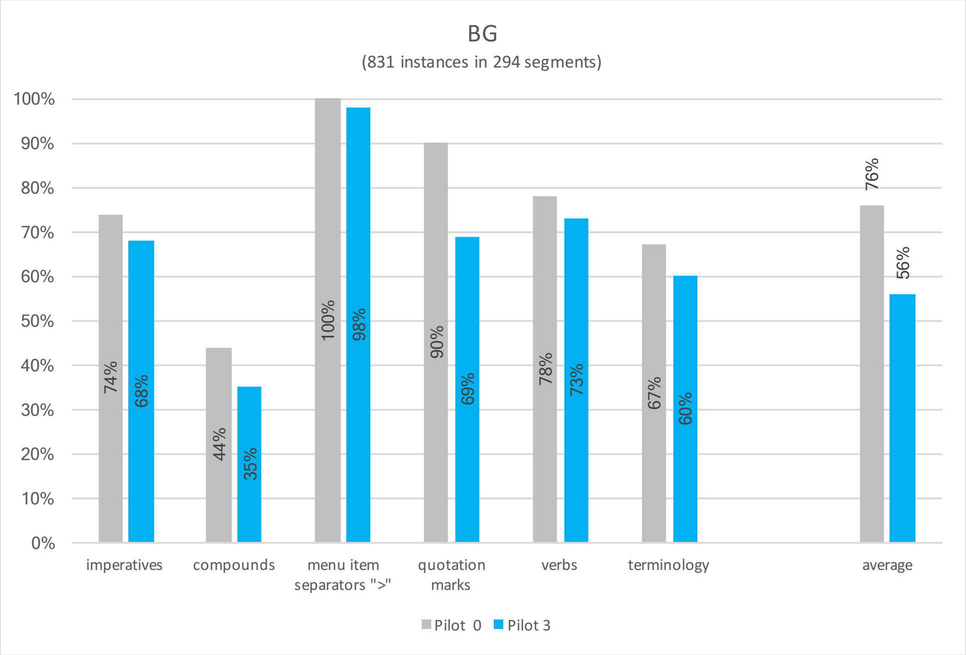

Figure 6: Manual evaluation results of Bulgarian

(A)	source:	Yes. Go to the Contacts menu $\geq$ Advanced $\geq$ Back up contacts to file ...	*2 inst.*
	Pilot 0:	Да. Преход към контактите меню $\geq$ разширени $\geq$ подкрепи контакти до файл ...	*2 inst.*
	Pilot 3:	Да. Отидете на contacts меню > > разширени назад с контакти за file ...	*0 inst.*
	reference:	Да. Отидете в менюто Contacts $\geq$ Advanced $\geq$ Back up contacts to file ...	

Example A depicts the analysis of the menu item. The source contains two instances of the separator. The Pilot 0 system treats all separators correctly. Pilot 3 system places the separators next to each other, so there is no correct instance.

(B)	source:	If you have WinRAR installed, press the right mouse button on the file and then select Extract here ...	*2 inst.*
	Pilot 0:	Ако сте winrar инсталирани натискате десния бутон на мишката върху файла и отметнете добива тук ...	*1 inst.*
	Pilot 3:	Ако имате winrar installed, натиснете десния бутон на мишката върху файла и след това изберете extract тук ...	*2 inst.*
	reference:	Ако имате инсталиран WinRAR, натиснете десния бутон на мишката върху файла и след това изберете Extract here ...	

(C) source: The left side has a tab that says, my subscriptions... click and go to the subscription you want to cancel. Click the arrow next to the subscription and select Unsubscribe ... *4 inst.*

Pilot 0: Лявата страна има раздел, в която се казва, че моята абонаменти... щракнете и да отиде в абонамента искате да отмените. Щракнете върху стрелката до иконата абонамент и изберете отписвам ... *3 inst.*

Pilot 3: Лявата страна има раздел, в която се казва, че, моят subscriptions... Щракнете върху и да отиде в subscription искате да CANCEL. Щракнете върху стрелката непосредствено до subscription и да посочите unsubscribe ... *2 inst.*

reference: Отляво се намира табулатора My subscriptions... кликнете върху него и отидете на абонамента, който искате да отмените. Кликнете върху стрелката до абонамента и изберете Unsubscribe ...

Examples B and C illustrate the translation of imperative forms. There are two correct instances in the source sentence in B. Pilot 0 translates the second verb form from example B (select) correctly, but the first one (press) is translated in the form of present tense. At the same time, this system translates correctly three of all four instances from example C. Pilot 3 translates both verb forms from example B correctly, but only two of the four imperative forms from example C are translated correctly.

(D) source: Press and hold the Alt key and then click the color you want to duplicate. *5 inst.*

Pilot 0: Натиснете и задръжте върху Alt и щракнете върху цвета, който желаете да дублира. *4 inst.*

Pilot 3: Натиснете и задръжте клавиша Алт ключ и после щракнете върху цвят искате да duplicate. *4 inst.*

reference: Натиснете и задръжте клавиша Alt и след това кликнете върху цвета, който искате да използвате.

(E) source: Click the second mouse button on the desktop, select ”Personalize” and, finally, choose the theme that suits you. *4 inst.*

Pilot 0: Щракнете върху втората бутон на мишката върху работния плот, изберете "Персонализиране" и, накрая, изберете темата, която ви. *3 inst.*

Pilot 3: Щракнете върху втората бутон на мишката върху десктопът, изберете "personalize "и, накрая, изберете темата, че дела ви. *3 inst.*

reference: Кликнете с втория бутон на мишката върху десктопа, изберете „Personalize" и след това изберете темата, която ви харесва.

(F)	source:	Yes, after doing the search, at the top left of Google Maps there is an icon with a bus, click on it.	*3 inst.*
	Pilot 0:	Да, след прави издирване в горния ляв на Google карти има икона с автобус, щракнете върху нея.	*2 inst.*
	Pilot 3:	Да, след doing търсенето, в горния ляв ъгъл на Google maps има икона с bus, Click по него.	*1 inst.*
	reference:	Да, след като изпълните търсенето, в горния ляв ъгъл на Google Maps има икона с автобус, кликнете върху нея.	

Examples D, E and F illustrate the translation of verbs. The results in this area are satisfactory. Pilot 3 obtains the lower average value. The ratio ‘translated correctly – not translated correctly’ is the same in examples D and E. The verb ‘duplicate’ from example D is translated wrongly by system 0. Pilot 3 does not translate it and also there is an English verb form in the Bulgarian sentence. The situation in example E is similar to the one described in example D. The verb form ‘suits’ is not translated by both systems.

(G)	source:	In the terminal, type ”netstat-a”.	*1 inst.*
	Pilot 0:	В терминал, тип "netstat-a".	*1 inst.*
	Pilot 3:	В terminal, въведете "netstat-a ".	*0 inst.*
	reference:	В терминала напишете „netstat-a".	

Example G illustrates the translation of terms. There is only one term in the example. It is translated in Pilot 0 (although not grammatically correctly), but the verb in the sentence is translated like noun. Pilot 3 does not translate the term at all, but at the same time translates the verb correctly.

4.3.3 Czech

	#	Pilot 0	Pilot 3	Chimera
imperatives	73	51%	88%	86%
“>” separators	40	95%	95%	98%
quotation marks	90	98%	100%	100%
verbs	109	96%	97%	95%
terminology	99	92%	92%	93%
sum	411			
average		87%	95%	94%

Table 12: Translation accuracy on manually evaluated sentences in Czech focusing on particular phenomena. Test-sets consist of hand-picked source sentences of Batch 2 that include the respective phenomenon.

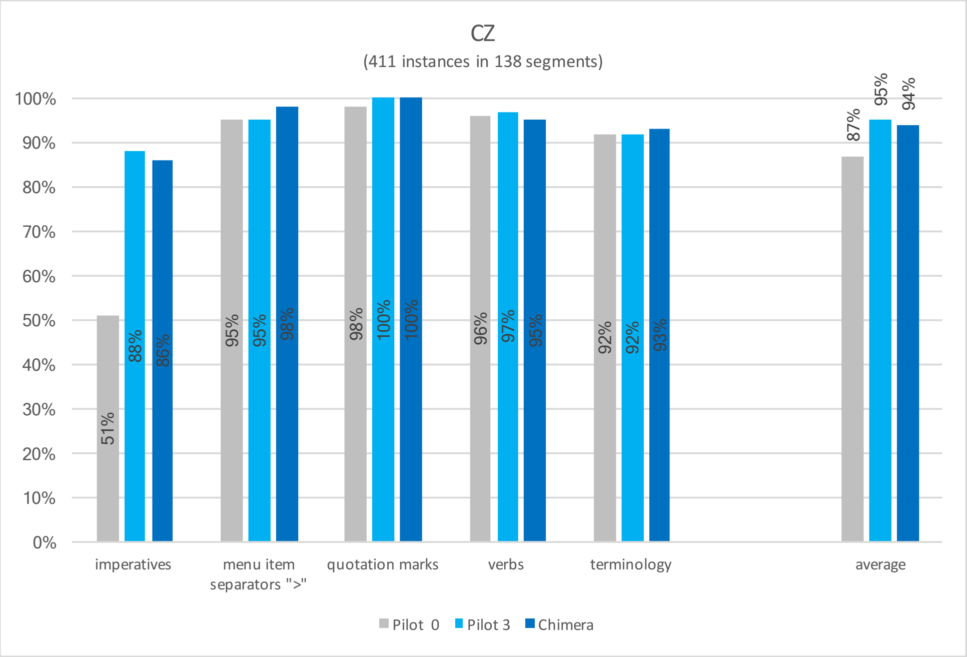

Figure 7: Manual evaluation results of Czech

For Czech, we have compared three systems: Pilot 0 (Moses), Pilot 3 (TectoMT, Section 2.4) and Chimera (TectoMT+Moses, Section 3). In general, Pilot 3 and Chimera achieved almost the same scores in all categories,[26] while being better than Pilot 0, especially in translation of imperatives. Pilot 0 translations use infinitive instead of imperative quite often, which leads to ungrammatical but mostly understandable Czech sentences. However, some problems with imperatives in Pilot 0 are more severe:

(A) source: Select the image and then adjust with the arrow keys.
Pilot 0: Vyberte obraz a pak na pomocí kurzorových kláves.
Pilot 3: Vyberte obrázek a potom upravte klávesy šipek.
Chimera: Vyberte obraz a pak upravte pomocí kurzorových kláves.
Reference: Vyberte obrázek a potom jej upravte pomocí kurzorových kláves.

In Example A, Pilot 0 completely omitted the second imperative verb, so the meaning is damaged. Pilot 3 and Chimera translated both imperatives correctly, but Pilot 3 has the *arrow keys* as direct object of the second imperative, while Chimera translated it correctly with *pomocí kurzorových šipek* (literally meaning *with the help of arrow keys*).

4.3.4 Dutch

The evaluation for Dutch consists of a comparison of three systems: Pilot 0 (Moses), Pilot 3 (TectoMT, Section 2.5) and Chimera (TectoMT+Moses, Section 3). While Chimera is better than Pilot 0 in all categories, Pilot 3 is only worse in the placement of quotation marks. Overall, both Chimera as Pilot 3 perform better as compared to Pilot 0.

[26] There are only very little compound nouns in Czech (unlike in German), so we have excluded this category from the evaluation.

	#	Pilot 0	Pilot 3	Chimera
imperatives	80	74%	90%	94%
compounds	37	65%	73%	76%
“>” separators	40	90%	100%	100%
quotation marks	86	88%	80%	97%
verbs	110	70%	87%	95%
terminology	164	80%	91%	95%
sum	517			
average		78%	88%	94%

Table 13: Translation accuracy on manually evaluated sentences in Dutch focusing on particular phenomena. Test-sets consist of hand-picked source sentences of Batch 2 that include the respective phenomenon.

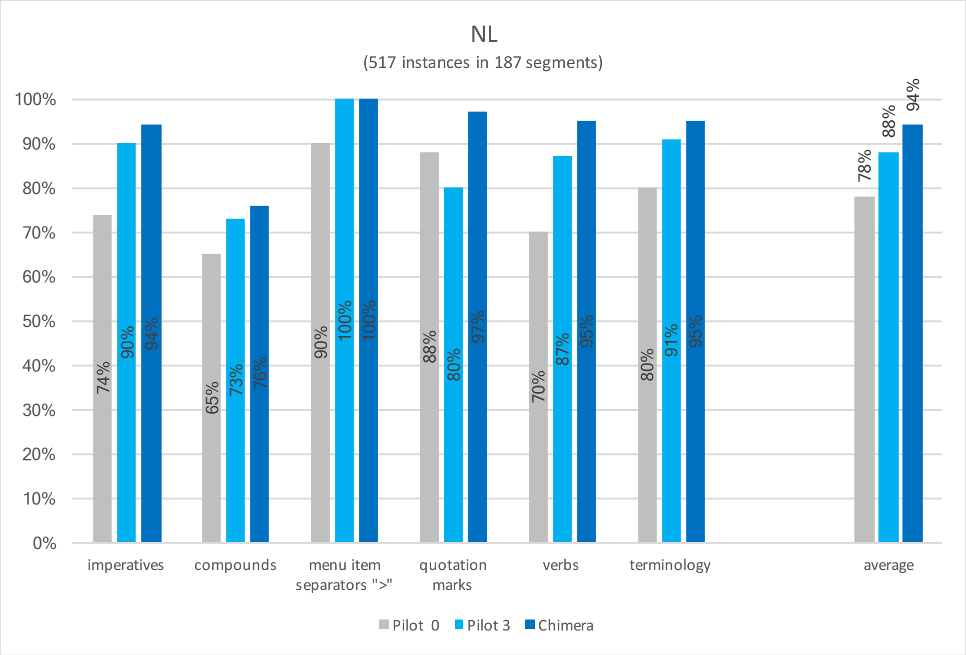

Figure 8: Manual evaluation results of Dutch

For example, Pilot 3 and Chimera perform better in translating imperatives, as can be seen in example A. Here Pilot 0 chooses to translate the imperative with an infinitive while the other systems translate it correctly. Example B, then, provides an illustration of a case where Chimera and Pilot 3 outperform Pilot 0 in the translation of terminology. With Pilot 0 the term “key” is translated with the Dutch word “sleutel” which refers to a tool that can be used to open the door. Chimera and Pilot 3, however, correctly translate this word with “toets” referring to a key on a keyboard.

(A) source: In the Insert menu, select Table.
Pilot 0: In het menu Invoegen Tabel selecteren.
Pilot 3: Selecteer Tabel in het Invoegen menu.
Chimera: In de Invoegen menu, selecteer Tabel.
Reference:In het menu Invoegen kiest u de optie “Tabel”

(B) source: Try pressing the F11 key.
Pilot 0: Probeer de toets F11 sleutel.
Pilot 3: Probeer de F11 toets in te drukken.
Chimera: Probeer pressing de F11 toets.
Reference:Probeer op F11 te drukken.

4.3.5 German

For the evaluation in German, we added another linguistic phenomenon that seemed prone to translation errors, namely the phrasal verbs. We included the evaluation on this category only in German because phrasal verbs exhibit a special behavior in German in certain contexts as their prefixes move to the end of a sentence in certain constructions. Thus, only translations in which the verb and its prefix are present are counted as correctly translated. Yet, the evaluation results of the phrasal verbs need to be treated with care because there are verbs that are acceptable with and without their prefix (e.g., auswählen vs. wählen) which makes the correct translations difficult to count. Additionally, only 57 segments with phrasal verbs could be extracted for German instead of 100 like for the other categories.

The manual evaluation for German includes five of the systems described in section 2.9: Pilot 0, Lucy, Lucy-improved, the neural system (NMT), and Pilot 3 (the selection mechanism choosing from the former systems except for Lucy). The results are presented in Table 14 and Figure 9.

	#	Pilot 0	Lucy	Lucy-imp.	NMT	Pilot 3
imperatives	247	68%	79%	79%	74%	73%
compounds	219	55%	87%	85%	51%	70%
“>” separators	148	99%	39%	83%	93%	80%
quotation marks	431	97%	94%	75%	95%	80%
verbs	505	85%	93%	93%	90%	90%
phrasal verbs	90	22%	68%	77%	38%	53%
terminology	465	64%	50%	53%	55%	54%
sum	2105					
average		76%	77%	77%	75%	74%

Table 14: Translation accuracy on manually evaluated sentences in German focusing on particular phenomena. Test-sets consist of hand-picked source sentences of Batch 2 that include the respective phenomenon. Lucy is separated as it does not participate in Pilot 3.

The overall average performance of the systems is very similar. The Lucy systems have the highest overall average scores but even though the other systems display similar overall average scores, their performances on the different linguistic phenomena are quite complimentary:

While the baseline **Pilot 0** system operates best of all systems on the menu item separators (“>”), the quotation marks and terminology, the baseline **Lucy** system performs best on the remaining linguistic categories, namely the imperatives, compounds and verbs; furthermore it is doing very well on phrasal verbs and quotation marks but has the

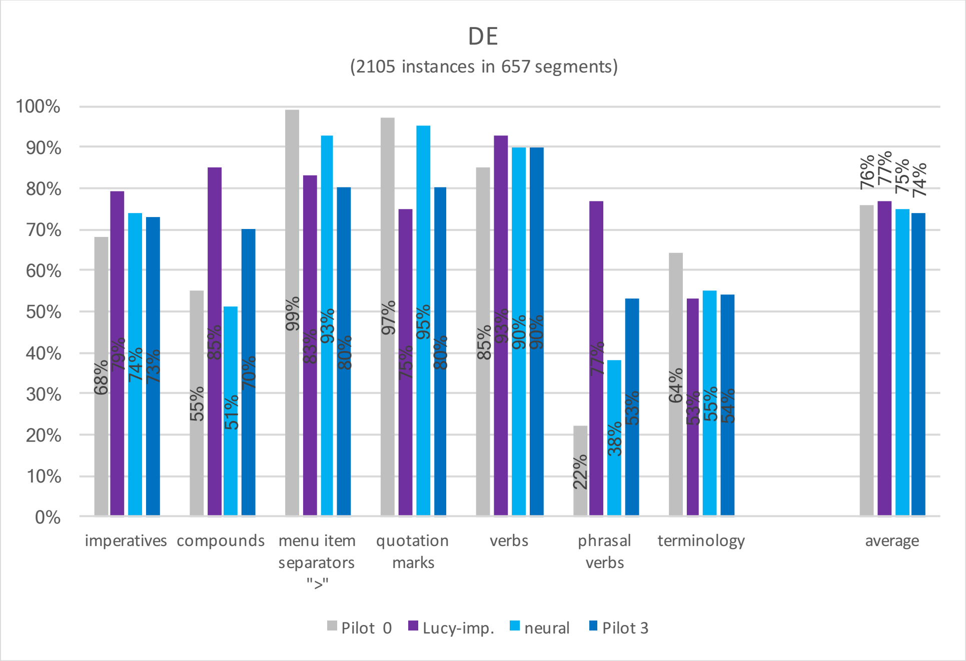

Figure 9: Manual evaluation results of German

lowest scores of all systems for the ">"-separators. The Pilot 0 system is also doing well on imperatives and verbs but performs worst of all systems on the phrasal verbs.

The improved version of the Lucy system, namely the **Lucy-improved**, reaches the same overall average score as its base system. Likewise, it ranks among the best-performing systems in terms of imperatives, compounds and verbs. Furthermore, it considerably improved on the category it was developed for, i.e., the menu item separator ">", and also on phrasal verbs as well as slightly on terminology. As a side effect of the improved treatment of the menu item separators, it unfortunately has visibly lower (but still good) scores for the quotation marks.

(A)	source:	[...] Adjustments ≥ Notification Center ≥ Mail.	*2 inst.*
	Pilot 0:	[...] Adjustments>-Benachrichtigungszentrale ≥ E-Mail.	*1 inst.*
	Lucy:	[...] Anpassungs->-Benachrichtigungs-Zentrums->-Post [...].	*0 inst.*
	Lucy-imp.:	[...] Anpassungen ≥ Benachrichtiungs-Zentrum ≥ Post [...].	*2 inst.*
	reference:	[...] Anpassungen ≥ Benachrichtigungszentrum ≥ Post [...].	

Example A depicts the analysis of the menu item separators and includes the two baselines as well as the improved Lucy system. The source contains two instances of the separator. The Pilot 0 output treats the words before and after the first separator as a compound, adding a hyphen after the separator. Therefore, only the second separator counts as correct. Lucy treats the separators similarly, adding hyphens before and after the separators, resulting in no correct instances. The improved Lucy version treats all separators correctly. Again, we are only evaluating the performance of the systems on menu item separators. There are other translation issues that are ignored at this point.

The **NMT** system ranks among the best systems regarding the imperatives, ">"-separators, quotation marks and verbs. Its score for the compounds on the other hand is the lowest of all systems. Although NMT is known for its ability to generate compounds (in contrast to phrase-based SMT), the domain-specific nature of the experiment might be the reason for this failure.

Pilot 3 obtains the lowest average value of all systems but this score is still only three percentage points less than the highest average value. The Pilot 3 selection mechanism is one of the best performing systems on verbs and terminology. Additionally, it manages to get a score that is higher than the average of its three component systems on the phrasal verbs. For the other phenomena this is not the case as it mostly reaches a score that is lower than the scores of two of its component systems.

(B)	source:	This feature prevents viruses from running and exploring the Windows Automatic Executions functionality [...].	*3 inst.*
	Pilot 0:	Diese Funktion verhindert, dass Viren aus und die Windows Automatic Execution Funktionalität, [...].	*1 inst.*
	Lucy/L-imp.:	Diese Funktion hält Viren davon ab, die Fenster-Automatisch-Ablauf-Funktionalität laufen zu lassen und zu erforschen, [...].	*3 inst.*
	Neural:	Diese Funktion verhindert Viren, die die Windows-Automationsfunktion mit Windows Automationsfuntionen zu starten, [...].	*2 inst.*
	Pilot 3:	Diese Funktion hält Viren davon ab, die Fenster-Automatisch-Ablauf-Funktionalität laufen zu lassen und zu erforschen, [...].	*3 inst.*
	reference:	Diese Funktion verhindert, dass Viren die automatische Windows-Ausführungs Funktion nutzen und erkunden, [...].	.

The source sentence in example B contains the three verbs “prevents”, “running” and “exploring” that are translated in the reference as “verhindert”, “nutzen” and “erkunden”. While Pilot 0 only translates “prevents” - “verhindert” and completely loses the other two verbs, the neural system translates “prevents” - “verhindern” and “running” - “starten”, losing the third verb. Only Lucy and the Lucy-improved correctly translate all three verbs and in this case the selection mechanism in Pilot 3 correctly selects Lucy/Lucy-improved as the best system.

(C)	source:	Right-click with the mouse on the element that is selected, then choose the Deselect option.	*2 inst.*
	Pilot 0:	Mit der rechten Maustaste klicken Sie mit der rechten Maustaste auf das Element, das ausgewählt ist, dann wählen Sie die Option Deselect.	*2 inst.*
	Lucy/L-imp.:	Recht-Klick mit der Maus auf dem Element, das ausgewählt wird, dann wählen der Option Deselektieren.	*0 inst.*
	Neural:	Klicken Sie auf die Maustaste auf das Element, das ausgewählt wird, wählen Sie dann die Option Deswählungsoption.	*2 inst.*
	Pilot 3:	Recht-Klick mit der Maus auf dem Element, das ausgewählt wird, dann wählen der Option Deselektieren.	*0 inst.*
	reference:	Klicken Sie mit der rechten Maustaste auf das ausgewählte Element und dann wählen Sie ’Option abwählen’ aus.	

In example C on the other hand, the selection mechanism falsely selects the Lucy/Lucy-improved output as the best output even though it mistranslates both of the two imperatives “Right-click” and “choose”, translating the former in a noun compound (“Recht-

Klick") and the latter in a verb ("wählen") without the needed pronoun "Sie". All the other systems correctly translate the imperatives including the obligatory pronoun as "Klicken Sie" and "wählen Sie".

It is not surprising that the selection mechanism does not always choose those systems performing best on the given error categories. The error categories are just one selected view on the output and the sentences typically contain other errors. In other words, e.g., getting all imperatives right is not a guarantee for producing the best possible translation.

As described above, the selection mechanism is using a huge variety of features to find the best translations. However, if the goal was to optimize performance on these error types, it one could device special features that trigger certain selections given that the input sentence contains the respective phenomenon. In future work, we plan to continue this line of cascaded development and human evaluation.

4.3.6 Portuguese

For the evaluation in Portuguese, one of the linguistic phenomenon in the list for testing was replaced. We evaluated word order instead of compounds.

The evaluation of word order in Portuguese was included because the nominal phrases with two or more nouns, or adjective plus noun, have a different default order in Portuguese than in English, and it is an important issue in the translation between these two languages. Because there are exceptions, we only analysed those cases where in Portuguese an order different than the order in English is required, so that there is an explicit word order reversal to be performed by the translation system. For example, cases in which the default English word order ADJ + N can and should be kept also in Portuguese were not taken into account for the purposes of the current evaluation.

Compounds for the translation between Portuguese and English, in turn, do not represent the major issue they may represent for other languages, like for instance, German.

Thus, the linguistic dimensions that were studied in order to analyse the performance of the English to Portuguese translation were six, namely: imperatives, word order, ">" separators, verbs, terminology and quotation marks (but for the special situation concerning quotation marks, see explanation below).

The manual evaluation of those dimensions shows the improvement of Pilot 3 with respect to Pilot 0. The Chimera system is better than Pilot 3 in four such dimensions, namely ">" separators, quotations marks, verbs and terminology. And both systems exhibit similar performance with imperatives. Finally, in the case of word order, Pilot 3 is one percent better than Chimera.

The results are presented in Table 11 and Figure 9.

	#	Pilot 0	Pilot 3 Tecto	Chimera
imperatives	245	16%	55%	55%
word order	174	30%	51%	49%
“>” separators	148	0%	74%	99%
quotation marks	429	81%	0%	93%
verbs	504	82%	84%	85%
terminology	285	38%	64%	67%
sum	1785			
average		54%	53%	77%

Table 15: Translation accuracy on manually evaluated sentences in Portuguese focusing on particular phenomena. Test-sets consist of hand-picked source sentences of Batch 2 that include the respective phenomenon.

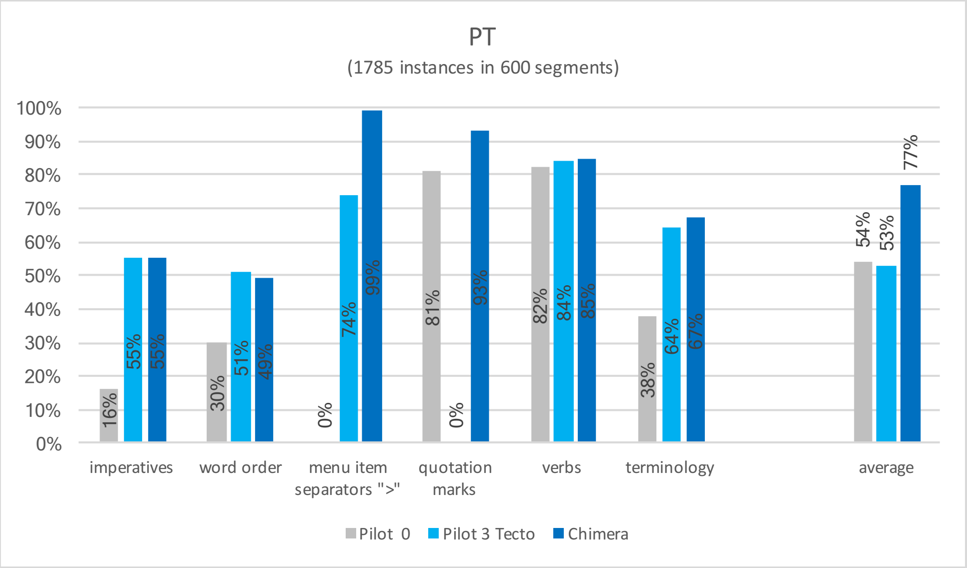

Figure 10: Manual evaluation results of Portuguese

Examples A-D below illustrate cases where Chimera performed better than Pilot 0 and Pilot 3. Example E shows a case where Chimera and Pilot 3 had the same performance and both were better than Pilot 0. Examples F-G illustrate the word order cases.

Example A illustrate the performance of the three systems when dealing with ”>”-separators. Pilot 0 was the only system having consistent problems with this dimension, while Chimera and Pilot 3 correctly place the separators. The surface elements of interest appear underline, in this and the subsequent examples.

(A) source: Go to Settings ≥ General≥ Accessibility≥ Invert Colors.
Pilot 0: Ir a contextos Geral ”” ”Accessibility inverte Colors.
Pilot 3: Vá a Configuração ≥ Geral ≥ Acessibilidade ≥ Inverter as Cores.
Chimera: Vá a Configuração ≥ Geral ≥ Acessibilidade ≥ Inverter Cores.
reference: Vá a Definições≥ Geral ≥ Acessibilidade ≥ Inverter Cores.

Example B shows the case of a sentence in which Chimera includes quotation marks while Pilot 3 does not, in accordance to the reference translation.

This situation deserves the following clarification. The QTLeap corpus is the result of the translation by human translators of the interactions, in Portuguese, which occured with users in the QA system, into English. This English version was then used by further human translators to obtain the other parallel versions in the other languages of the project.

It happens that, in the vast majority of the (base) Portuguese corpus, there are no quotations marks (that is the reason why the reference in Example B has no quotation marks) — this is so because the human operators replying to users through the chat of the QA helpdesk did not use quotation marks. However, the human translator superimposed his understanding of the correct spelling (in English) of these expressions and surrounded them with quotation marks. All the other human translators repeated these marks when translating into the other languages of the project.

Since the reference sentences in Portuguese do not have quotation marks, the translation system of Pilot 3 has a rule that does not allow the placing of quotation marks in a translation from English to Portuguese — hence providing the correct translation according to the reference sentence.

The SMT system Moses learns from massive parallel corpora where quotation marks appearing in an expression of one of the languages, also appear in the aligned expression in the other language. Hence, when translating into Portuguese, this system makes these marks appear around the Portuguese translations of the respective source English expressions.

As Chimera is based on translations from Moses, it also keeps the quotation marks across the translation step.

Summing up: though in Figure 9 the bar for quotation marks is null for Pilot 3, it represents full accuracy in translation with respect to the gold standard version in Portuguese.

Accordingly, **the average scores adjusted to accomodate this issue and reflect a correct ranking of the Portuguese systems are different than those displayed in Figure 9, and are respectively, 34% for Pilot 0, 76% for Pilot 3, and 55% for Chimera**.

(B)	source:	Click on the button beneath the Google+ logo, select the option Settings, and change where it says, “Who can send you notifications?”
	Pilot 0:	Clique no botão abaixo do Google + logotipo, escolher a opção contextos e mudança, onde se diz, “Quem pode enviar-lhe as notificações?”
	Pilot 3:	Clique no botão o logótipo de Google+ selecione a opção definição e altere onde diz Quem poderá enviar notificações dele?
	Chimera:	Clique no botão o logótipo de Google +, select a opção Configuração, e alterar onde diz, “Quem pode enviar-lhe as notificações?”
	reference:	Clique no botão por baixo do logotipo do google+, escolha a opção Definições, e altere onde diz Quem lhe pode enviar notificações?.

Example C illustrates how Pilot 3 and Chimera performed better in the translation of verbs. It includes also a common error of Pilot 0: the translation of a verb by a synonym, which is a noun with a very different meaning.

This example contains the verb “tap”, which was translated by Pilot 0 as the noun *torneira* (valve/faucet), instead of the verb *toque* (press/touch) as in translation output by Pilot 3 and Chimera.

(C) source: Open the file, application or message where you want to paste the text. Tap where you want to paste the text. In the text suggestions bar above the keyboard, tap Paste
Pilot 0: Abrir o dossiê, aplicação ou mensagem onde querem pasta o texto. Torneira onde querem pasta o texto. No texto sugestões fasquia acima do teclado, torneira pasta
Pilot 3: Abra o ficheiro, a aplicação ou a mensagem onde quer colar o texto. Toque onde quer colar o texto. Colar de no barra de sugestões de texto o teclado toque.
Chimera: Abrir o ficheiro, aplicação ou a mensagem onde pretende colar o texto. Toque onde quer colar o texto. Na barra de sugestões de texto o teclado, toque Colar
reference: Abra o ficheiro, aplicação ou mensagem onde pretende colar o texto. Toque no local em que pretende colar o texto. Na barra de sugestões de texto acima do teclado, toque em Colar

Example D illustrates the dimmension of terminology. Even though the terms “Layer” and “New” were not translated in the reference translation (a recurrent issue), we can see that Pilot 3 and Chimera have correct translations for both terms. With Pilot 0 there are consistent problems in the translation of terminology as the portuguese words chosen usually convey different meanings unrelated to the domain. That is what happens in example D with “menu” being translated by Pilot 0 into *ementa* (restaurant menu) and “layer” into *ozono* (ozone layer).

(D) source: Go to menu Layer > New > Layer...
Pilot 0: Ir ao ozono ”ementa” Novo ozono...
Pilot 3: Vá menu Camada > Novos > Camada...
Chimera: Vá menu Camada > Novo > Camada...
reference: Vá ao Menu Layer > New > Layer...

As for Imperatives, Chimera and Pilot 3 exhibit similar results, which are better than Pilot 0 results in this respect. Example E shows an example of the output of infinitives by Pilot 0 to translate English imperative forms. It shows also the output by Pilot 3 of an informal Imperative (the use of the second person of the Imperative). None is semantically incorrect, even though the pragmatically correct option is, as shown in the reference and in Chimera, the use of the Imperative form in the third person - *selecione* / *seleccione*, depending on the orthographic rule used.

(E) source: Select the folder with the right mouse button and choose add to archive
Pilot 0: Seleccionar os folder com direito rato botão e escolher acrescentar ao arquivo
Pilot 3: Seleciona a pasta com o direito botão de rato e escolha adicione arquivo.
Chimera: Selecione a pasta com o direito botão do rato e escolha adicionar para arquivo
reference: Seleccione com o segundo botão do rato a pasta e escolha add to archive...

Examples F and G illustrate the translation of noun plus noun and adjective plus noun. In the word order dimension, Chimera performed slightly worse than Pilot 3. In both examples the correct Portuguese translation results from the change of word order, which happened when "main window" was translated to *janela principal* (window main) and "mouse wheel" was translated to *roda do rato* (wheel of_the mouse).

(F) source: In the main window you should see a button saying Code. Click on it.
Pilot 0: No principal janela deveria ver um botão dizendo Código. Clique sobre ela.
Pilot 3: Na janela principal veja um botão dito Código. Clique nele.
Chimera: Na janela principal veja um botão dito Código. Clique nele.
reference: Na janela principal têm um botão que diz Code. Clique no mesmo.

(G) source: Press the Alt key and then rotate the mouse wheel.
Pilot 0: A imprensa Alt chave e então rotação o rato roda.
Pilot 3: A tecla de Alt de carregar e depois inverte a roda do rato.
Chimera: Prima a tecla de Alt e depois rodar a roda do rato.
reference: Carregue na tecla Alt e depois rode a roda do rato.

4.3.7 Spanish

	#	Pilot 0	Pilot 3	Chimera
imperatives	43	0%	81%	84%
compounds	34	15%	62%	53%
">" separators	20	0%	50%	100%
quotation marks	80	79%	84%	100%
verbs	94	32%	68%	71%
terminology	90	14%	46%	48%
sum	361			
average		31%	66%	73%

Table 16: Translation accuracy on manually evaluated sentences in Spanish focusing on particular phenomena. Test-sets consist of hand-picked source sentences of Batch 2 that include the respective phenomenon.

Six linguistic features were studied to analyse English to Spanish translation performance, namely, imperatives, compounds, ">"-separators, quotation marks, verb formation and terminology. The linguistic features studied show sigfinicantly improved results for Pilot 3 with respect to Pilot 0 and even better results for Chimera (with the exception of compounds). The increase in correctness is particularly evident for imperatives, compounds, ">"-separators, verbs and terminology.

Examples A-D below show cases where Chimera and Pilot 3 performed better than Pilot 0. In particular, Example A shows how Pilot 0 outputs infinitives for English imperative forms. Whereas this is not incorrect, it is common practice - and this is what the reference shows - to use the imperative form in Spanish, as output by Pilot 3 and Chimera.

(A) source: In the Insert menu, select Table.
Pilot 0: En el menú Insert, seleccionar el Cuadro.
Pilot 3: En el menú de Insertar seleccione Tabla.
Chimera: En el Insertar menú, seleccione Tabla.
reference: En el menú Insertar, seleccione Tabla.

Example B shows the incorrect translation of ">"-separators by Pilot 0, which tends to output quotations marks in the relative positions where the separators should be present. Cases where Pilot 3 is worse than Chimera usually include the ">"-separator in the correct relative position but the noun phrases joined by it are split.

(B) source: Yes, go to Tools> Word Count.
Pilot 0: Sí, ir a Tools Palabra “el Conde.
Pilot 3: Sí vaya a Herramientas > Número de palabras.
Chimera: Sí, vaya a Herramientas > Número de palabras.
reference: Sí, vaya a Herramientas > Contar palabras.

Example C shows the case of a source sentence which includes five instances of conjugated verbs. Pilot 0 does not manage to correctly translate any of the verbs, whereas Pilot 3 and Chimera correctly output three out of five. The copulative verb within the relative clause was incorrectly translated and the imperative *choose* was translated following the pattern for regular verbs despite being irregular in Spanish (*elegir* -> *elija*).

(C) source: Select the cells you want to format, then click the right mouse button on one that is selected and choose Format Cells.
Pilot 0: Seleccionar las células quieren formato, entonces clic el derecho al ratón botón en uno que sea seleccionado y elegir Format de Combustible.
Pilot 3: Seleccione las celdas que quiere formato, después haga clic el botón derecho del el ratón en uno, que selecciona y eleja Formato de celdas.
Chimera: Seleccione las celdas que quiere formato, después haga clic el botón derecho del el ratón en uno que es seleccionares y eleja Formato de celdas.
reference: Seleccione las celdas que desea dar formato y, a continuación, haga clic derecho en una que está seleccionada y elija Formatear celdas.

Example D focuses on terminology. The source sentence contains four terminological elements, *menu Layer, New* and *Layer*. Pilot 0 does not translate any of them correctly, whereas Pilot 3 and Chimera correctly use the terms *menú Capa, Nuevo* and *Capa* in their translations.

(D) source: Go to menu Layer > New > Layer...
Pilot 0: Ir a Comunidades menú ”Nueva Comunidades”...
Pilot 3: Vaya menú Capa > Nuevo > Capa...
Chimera: Vaya a menú Capa > Nuevo > Capa...
reference: Vaya al menú Capas > Nuevo > capas...

Quotation marks were already well handled by Pilot 0, leaving little room for immprovement. Example E below shows a cases where Pilot 0 and Pilot 3 translate one out of two correctly and where Chimera translates both correctly.

(E) source: At the bottom right of the screen has a yellow “snowman”; drag it to the street that you want to see.
Pilot 0: En el fondo derecho de la pantalla tiene un “amarilla snowman”; arrastrar a la calle que quieren ver.
Pilot 3: Inferior derecho de la pantalla tiene un “snowman yellow”; arrastre su a la calle que quiere ver.
Chimera: El inferior derecho de la pantalla tiene un yellow “snowman”; arrastre, a la calle que quiere ver.
reference: En la parte inferior derecha de la pantalla hay un hombre de nieve Amarillo; Arrástrelo a la calle que quiere ver.

The only cases where Chimera performed slightly worse than Pilot 3 were when dealing with compounds. Example F shows a case where the UI string *Language menu* was correctly translated as *menú de idioma(s)* by Pilot 0 and Pilot 3 and not by Chimera, which output the two elements in the incorrect order.

(F) source: Yes you can. You have to go to the Language menu and there choose the language in which you are programming.
Pilot 0: Sí puede usted: Usted ha de ir a el menú de Idiomas y no elegir la lengua en la que usted es la programación.
Pilot 3: Sí poder. Debe ir al menú de Idioma y allí elega el lenguaje, en que estar programa.
Chimera: Sí puede. No deben ir a la Idioma menú y allí elega el lenguaje, en que que son programa.
reference: Sí, es posible. Tiene que ir al menú de idiomas y allí elegir el idioma en el que está programando.

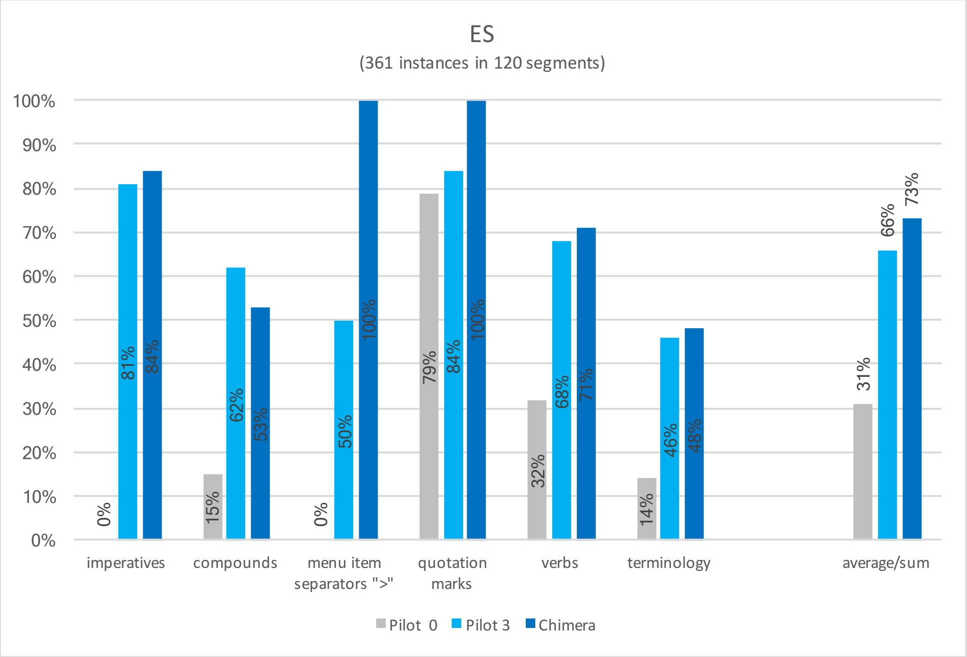

Figure 11: Manual evaluation results of Spanish

5 Conclusion

This deliverable has described the final state of MT systems created in the QTLeap project whose aim was to progressively include deeper linguistic processing methods into the MT pipeline. This deliverable has described also the results of their automatic and manual intrinsic evaluation.

Depending on the availability of training material, it may be hard to beat a pure SMT baseline with a more complex “deep” architecture given the limitation in the available resources for parallel deep language processing. However, as the BLEU scores suggest, this happens for the vast majority of the language pairs in the project, with the deeper systems (here named Pilot 3) showing very good results that outperform the SMT baselines (Pilot 0).

We worked also on further hybrid-MT research, where we developed a system called Chimera, which is a combination of TectoMT (Pilot 3) and Moses (Pilot 0, state-of-the-art phrase-based SMT system). Chimera significantly improves upon both Pilot 1 and Pilot 3, achieving thus the best results for en→cs, en→es, en→nl and en→pt.

The evaluations described in deliverable D3.12 have confirmed the main outcome of the intrinsic evaluations presented in this deliverable, that also the human users tend to favour the deeper systems. That said, there were few cases of disagreement between the automatic evaluations and human evaluations, so for the readers interested in performance of a particular pilot system for a a given language pair in the scope of the HF helpdesk usage scenario, we recommend reading D3.12.

In the last two years, we have seen a trend of Neural MT systems that sometimes may produce more fluent output than standard SMT systems. We have implemented such experimental neural system for German within the last phase of the project, following the recent trends in Neural MT research (bidirectional GRU-based RNN encoder-decoder with attention and BPE preprocessing, cf. Section 2.9.2). However, we have not confirmed the conjecture of more fluent translations within the QTLeap settings.

Although the test-suite-based manual evaluation presented in the end of this deliverable has only been performed towards the end of the project (it was developed in cooperation with the ongoing QT21 project), we believe that the insights gained will drive future research on these systems and languages. In contrast to purely statistical systems, where the only option would be the creation or acquisition of new data, the systems described here offer the possibility to perform targeted improvements – a feature that is one of the top desiderata from language industry.

References

Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Juan A Alonso and Gregor Thurmair. The Comprendium Translator system. In *Proceedings of the Ninth Machine Translation Summit*. International Association for Machine Translation (IAMT), 2003.

Nora Aranberri, Eleftherios Avramidis, Aljoscha Burchardt, Ondrej Klejch, Martin Popel, and Maja Popovic. Tools and guidelines for principled machine translation development. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 1877–1882. European Language Resources Association, 5 2016.

Jose Maria Arriola. Processing euskara complex postpositions in a rule based approach in order to improve shallow syntactic disambiguation. In *31 e Colloque International sur le Lexique et la Grammaire*, 2012.

Eleftherios Avramidis. Sentence-level ranking with quality estimation. *Machine Translation (MT)*, 28(Special issue on Quality Estimation):1–20, 2013.

Eleftherios Avramidis. Qualitative: Python Tool for MT Quality Estimation Supporting Server Mode and Hybrid MT. *The Prague Bulletin of Mathematical Linguistics (PBML)*, 106:147–158, 2016. URL https://ufal.mff.cuni.cz/pbml/106/art-avramidis.pdfhttps://www.dfki.de/web/forschung/publikationen/renameFileForDownload?filename=art-avramidis.pdf{&}file{_}id=uploads{_}2965.

Eleftherios Avramidis, Aljoscha Burchardt, Vivien Macketanz, and Ankit Srivastava. Dfki's system for wmt16 it-domain task, including analysis of systematic errors. In *Proceedings of the First Conference on Machine Translation*, pages 415–422. Association for Computational Linguistics, 8 2016a.

Eleftherios Avramidis, Burchardt, Aljoscha, Vivien Macketanz, and Ankit Srivastava. DFKI's system for WMT16 IT-domain task, including analysis of systematic errors. In *Proceedings of the First Conference on Machine Translation*, pages 415–422, Berlin, Germany, aug 2016b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W16/W16-2329.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, September 2014. URL http://arxiv.org/abs/1409.0473. arXiv: 1409.0473.

Ondřej Bojar and Aleš Tamchyna. CUNI in WMT15: Chimera strikes again. In *Proceedings of the 10th WMT*, pages 79–83, Stroudsburg, PA, USA, 2015. ACL. ISBN 978-1-941643-32-7.

Ondřej Bojar, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. The joy of parallelism with CzEng 1.0. In *Proceedings of the 8th International Conference on*

Language Resources and Evaluation (LREC 2012), pages 3921–3928, İstanbul, Turkey, 2012. European Language Resources Association. ISBN 978-2-9517408-7-7.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2013 WMT. In *Proceedings of the Eight WMT*, pages 1–44, Sofija, Bulgaria, 2013a. Bălgarska akademija na naukite, ACL. ISBN 978-1-937284-57-2.

Ondřej Bojar, Rudolf Rosa, and Aleš Tamchyna. Chimera – three heads for English-to-Czech translation. In *Proceedings of the Eight WMT*, pages 92–98, Sofija, Bulgaria, 2013b. Bălgarska akademija na naukite, ACL. ISBN 978-1-937284-57-2.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 WMT. In *Proceedings of the Ninth WMT*, pages 12–58, Baltimore, MD, USA, 2014. ACL. ISBN 978-1-941643-17-4.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. Findings of the 2015 WMT. In *Proceedings of the 10th WMT*, pages 1–46, Stroudsburg, PA, USA, 2015. ACL. ISBN 978-1-941643-32-7.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation (WMT16). In Ondřej Bojar and et al ., editors, *Proceedings of the First Conference on Machine Translation (WMT). Volume 2: Shared Task Papers*, volume 2, pages 131–198, Stroudsburg, PA, USA, 2016a. ACL. ISBN 978-1-945626-10-4.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Proceedings of the 19th International Conference on Text, Speech and Dialogue*, Lecture Notes in Artificial Intelligence, Brno, 2016b. Springer.

Aljoscha Burchardt, Kim Harris, Georg Rehm, and Hans Uszkoreit. Towards a systematic and human-informed paradigm for high-quality machine translation. In Georg Rehm, Aljoscha Burchardt, Ondrej Bojar, Christian Dugast, Marcello Federico, Josef van Genabith, Barry Haddow, Jan Hajic, Kim Harris, Philipp Koehn, Matteo Negri, Martin Popel, Lucia Specia, Marco Turchi, and Hans Uszkoreit, editors, *Proceedings of the LREC 2016 Workshop "Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem"*. o.A., 5 2016.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy, apr 2006. ISBN 1932432590. doi: 10.1145/1083784.1083789. URL http://www.iccs.inf.ed.ac.uk/{~}osborne/papers/eacl06.pdf.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece, mar 2009. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv:1409.1259 [cs, stat]*, September 2014. URL http://arxiv.org/abs/1409.1259. arXiv: 1409.1259.

Matthew S Crouse, Robert D Nowak, and Richard G Baraniuk. Wavelet-based statistical signal processing using Hidden Markov Models. *Signal Processing, IEEE Transactions on*, 46(4):886–902, 1998.

Nadir Durrani, Helmut Schmid, Alexander Fraser, Philipp Koehn, and Hinrich Schütze. The operation sequence model—combining n-gram-based and phrase-based statistical machine translation. *Computational Linguistics*, 2015.

Ondřej Dušek and Filip Jurčíček. Robust multilingual statistical morphological generation models. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 158–164, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-3023.

Ondřej Dušek, Zdeněk Žabokrtský, Martin Popel, Martin Majliš, Michal Novák, and David Mareček. Formemes in English-Czech Deep Syntactic MT. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 267–274, Montréal, Canada, 2012. Association for Computational Linguistics.

Rosa Gaudio, Gorka Labaka, Eneko Agirre, Petya Osenova, Kiril Simov, Martin Popel, Dieke Oele, Gertjan van Noord, Luís Gomes, João António Rodrigues, Steven Neale, João Silva, Andreia Querido, Nuno Rendeiro, and António Branco. Smt and hybrid systems of the qtleap project in the wmt16 it-task. In *Proceedings of the First Conference on Machine Translation*, pages 435–441, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W16/W16-2332.

Joachim Hartung, Guido Knapp, and Bimal K. Sinha. *Statistical Meta-Analysis with Applications*. Wiley, 2008. ISBN 9780470386347. doi: 10.1002/9780470386347. URL http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470290897.html.

John Langford, Lihong Li, and Alex Strehl. Vowpal wabbit online learning project, 2007.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Pavel Pecina, and Ondřej Bojar. CUNI system for WMT16 automatic post-editing and multimodal translation tasks. *CoRR*, abs/1606.07481, 2016. URL http://arxiv.org/abs/1606.07481.

David Mareček, Rudolf Rosa, Petra Galuščáková, and Ondřej Bojar. Two-step translation with grammatical post-processing. In Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan, editors, *Proceedings of the Sixth WMT*, pages 426–432, Edinburgh, UK, 2011. University of Edinburgh, ACL. ISBN 978-1-937284-12-1.

David Mareček, Martin Popel, and Zdeněk Žabokrtský. Maximum entropy translation model in dependency-based mt framework. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 201–206, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W10-1730.

Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd annual meeting of the Association for Computational Linguistics*, pages 91–98. Association for Computational Linguistics, 2005a.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530. Association for Computational Linguistics, 2005b.

Steven Neale, Luís Gomes, Eneko Agirre, Oier Lopez de Lacalle, and António Branco. “word sense-aware machine translation: Including senses as contextual features for improved translation models”. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 2777–2783, Portorož, Slovenia, 6 2016. European Language Resources Association (ELRA).

Václav Novák and Zdeněk Žabokrtský. Feature Engineering in Maximum Spanning Tree Dependency Parser. In Václav Matoušek and Pavel Mautner, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 10th International Conference on Text, Speech and Dialogue*, volume 4629 of *Lecture Notes in Computer Science*, pages 92–98, Berlin / Heidelberg, 2007. Springer. ISBN 978-3-540-74627-0.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

M. Popel and Z. Žabokrtský. Improving English-Czech Tectogrammatical MT. *The Prague Bulletin of Mathematical Linguistics*, 92:115–134, 2009.

Martin Popel, David Mareček, Nathan Green, and Zdeněk Žabokrtskỳ. Influence of parser choice on dependency-based MT. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 433–439. Association for Computational Linguistics, 2011. URL http://dl.acm.org/citation.cfm?id=2133019.

Martin Popel, Ondřej Dušek, António Branco, Luís Gomes, João Rodrigues, João Silva, Eleftherios Avramidis, Aljoscha Burchardt, Arle Lommel, Nora Aranberri, Gorka Labaka, Gertjan van Noord, Rosa Del Gaudio, Michal Novák, Rudolf Rosa, Jaroslava Hlaváčová, Jan Hajič, Velislava Todorova, and Aleksander Popov. Report on the second mt pilot and its evaluation. deliverable d2.8. Technical report, 2015.

Jan Ptáček and Zdeněk Žabokrtský. Synthesis of Czech sentences from tectogrammatical trees. In Petr Sojka, Ivan Kopeček, and Karel Pala, editors, *Lecture Notes in Artificial Intelligence, Text, Speech and Dialogue. 9th International Conference, TSD 2006, Brno, Czech Republic, September 11–15, 2006, Proceedings*, volume 4188, pages 221–228, Berlin / Heidelberg, 2006. Masarykova univerzita, Springer. ISBN 978-3-540-39090-9. URL http://ufal.mff.cuni.cz/~zabokrtsky/papers/tsd06-synthesis.pdf.

J. Ptáček. Two Tectogrammatical Realizers Side by Side: Case of English and Czech. In *Fourth International Workshop on Human-Computer Conversation*, Bellagio, Italy, 2008.

Loganathan Ramasamy, David Mareček, and Zdeněk Žabokrtský. Multilingual dependency parsing: Using machine translated texts instead of parallel corpora. *The Prague Bulletin of Mathematical Linguistics*, 102:93–104, 2014. ISSN 0032-6585.

Rudolf Rosa. Depfix, a tool for automatic rule-based post-editing of SMT. *PBML*, 102: 47–56, 2014. ISSN 0032-6585.

Rudolf Rosa, Ondřej Dušek, Michal Novák, and Martin Popel. Translation model interpolation for domain adaptation in TectoMT. In Jan Hajič and António Branco, editors, *Proceedings of the 1st Deep Machine Translation Workshop*, pages 89–96, Praha, Czechia, 2015. ÚFAL MFF UK. ISBN 978-80-904571-7-1.

Rudolf Rosa, Martin Popel, Ondřej Bojar, David Mareček, and Ondřej Dušek. Moses & Treex hybrid MT systems bestiary. In *Proceedings of the Second Deep Machine Translation Workshop*, Lisbon, Portugal, 2016a.

Rudolf Rosa, Roman Sudarikov, Michal Novák, Martin Popel, and Ondřej Bojar. Dictionary-based domain adaptation of mt systems without retraining. In *Proceedings of the First Conference on Machine Translation*, pages 449–455, Berlin, Germany, August 2016b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W16/W16-2334.

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

P. Sgall, E. Hajičová, and J. Panevová. *The meaning of the sentence in its semantic and pragmatic aspects*. D. Reidel, Dordrecht, 1986. ISBN 9027718385.

Drahomíra Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. The best of two worlds: Cooperation of statistical and rule-based taggers for czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing 2007*, pages 67–74, Praha, Czechia, 2007. Univerzita Karlova v Praze, Association for Computational Linguistics. ISBN 978-1-932432-88-6.

Jana Straková, Milan Straka, and Jan Hajič. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P14/P14-5003.

Zdeněk Žabokrtský. *From Treebanking to Machine Translation*. Habilitation thesis, Charles University in Prague, 2010. URL http://ufal.mff.cuni.cz/~zabokrtsky/publications/theses/hab-zz.pdf.

Daniel Zeman. Reusable tagset conversion using tagset drivers. In *Proceedings of LREC*, pages 213–218, 2008.

Z. Žabokrtský, J. Ptáček, and P. Pajas. TectoMT: highly modular MT system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 167–170. Association for Computational Linguistics, 2008.

Zdeněk Žabokrtský and Martin Popel. Hidden Markov Tree Model in Dependency-based Machine Translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 145–148, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1667583.1667628.