qtleap

quality
translation
by deep
language
engineering
approaches

Final report of the curation of LRTs for deep MT

DELIVERABLE D2.12

VERSION 1.0 | 2016-12-15

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	Oct 16, 2016	Petya Osenova	IICT-BAS	First draft
0.2	Oct 18, 2016	Martin Popel	CUNI	contribution
0.3	Oct 24, 2016	Eleftherios Avramidis	DFKI	contribution
0.4	Oct 24, 2016	Gorka Labaka	UPV-EHU	contribution
0.5	Oct 25, 2016	João Silva	FCUL	contribution
0.6	Oct 26, 2016	Gertjan van Noord, Dieke Oele	UG	contribution
0.7	Oct 27, 2016	Rosa Del Gaudio	HF	contribution
1.0	Oct 28, 2016	Markus Egg	UBER	Final version review completed

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.

Final report of the curation of LRTs for deep MT

DOCUMENT QTLEAP-2016-D2.12
EC FP7 PROJECT #610516

DELIVERABLE D2.12

completion
FINAL
status
SUBMITTED
dissemination level
PUBLIC

responsible
Kiril Simov (Task 2.4 Coordinator)
reviewer
Markus Egg
contributing partners
FCUL, DFKI, CUNI, IICT-BAS, UBER, UPV-EHU, UG, with HF

authors
Petya Osenova, Rosa Del Gaudio, João Silva, Eleftherios Avramidis,
Martin Popel, Gertjan van Noord, Dieke Oele, Gorka Labaka

Contents

List of Abbreviations

BDT	Basque Dependency Treebank
CoNLL	Conference on Natural Language Learning
LRTs	Language Resources and Tools
MT	Machine Translation
NAF	NLP Annotation Format
NER	Named Entity Recognition
NLP	Natural Language Processing
POS	Part-Of-Speech
QA	Question Answering
RBMT	Rule-based Machine Translation
UD	Universal Dependencies
WSJ	Wall Street Journal

1 Introduction

This deliverable reports on the curation of language resources and tools (LRTs) for Machine Translation (MT) pertaining to WP2. This reporting refers to the whole cycle of the MT Pilots within the Project. An interim status of curated LRTs was reported in D2.5 (M17). It should be noted that some of the resources and tools have already been mentioned in other deliverables.

The summary of the reporting is as follows:

LRTs that support Pilot 2 with the enhancement of lexical semantics received a first description in deliverable D5.4. LRTs that support Pilot 3 with the enhancement with deep processing received a first description in deliverable D4.7. These deliverables had already been delivered in M12.

Then, the LRTs further developed to support Pilot 2 were documented at M18 and M30, respectively in deliverables D5.6 and D5.9.

And the LRTs further developped for Pilot 3 were documented at M24 and M35 in deliverables D4.10 and D4.12.

The deliverable pertaining to WP2 that reported on the curation of LRTs was D2.5. Its focus was on the curation of LRTs for Pilot 1 in accordance to the plans in the DoW and their further specification in Section 5.2 of deliverable D1.3. It also took into account the initial status of the QTLeap corpus, which was relevant for Pilot 1 as well as for the other MT Pilots.

In WP2, that Deliverable D2.5 is the predecesor of the current deliverable, which describes the developments of the QTLeap corpus and the WP2 specific LRTs after Pilot 1 (i.e., in relation to Pilot 2 and Pilot 3).

The curation of LRTs reported in D2.5 and in the current deliverable refers to the dimensions of: *adaptation*, *further developments*, and *improvement*.

The present deliverable is structured into two major chapters. In the next chapter, the QTLeap corpus is presented with a short synopsis from the previous stage as well as with its new developments. In the subsequent chapter, the status and the improvement of the rest of the LRTs are addressed in seven sections, each concerning one of the relevant project language pairs.

2 QTleap Multilingual Corpus

The QTLeap corpus is a language resource that was created within the project. This corpus was gathered and organized for serving multiple purposes, such as the monitoring of the translation pipelines and the webservices (Tasks 3.1 and 3.2) and the evaluation of the MT pilots (Tasks 2.5 and 3.5).

The corpus contains two parts: in-domain (IT-related) and out-of-domain (newsmedia related). The in-domain part was reported in D2.5. Afterwards, it was further cleaned, curated, and annotated per language pair. The out-of-domain part was added after D2.5.

Here we first repeat the information that relates to the in-domain part of the corpus and that was already reported in D2.5. Then we report on the out-of-domain part.

The QTLeap corpus is composed of 4 000 pairs of questions and respective answers in the domain of ICT troubleshooting for both hardware and software. This material was collected using a real-life, commercial, online support service via chat. The corpus is thus composed of naturally occurring utterances produced by users while interacting with that service. The support system, called PcWizard, was created as the first point of

Question	Without Wi-Fi
Answer	Check if your PC can detect a wireless network. Otherwise, the wireless card may be disabled. Sometimes, computers have a physical button or switch that can turn the wireless network card on and off.
Question	Help on installing a printer
Answer	Try installing the drivers from the CD that came with the printer. If you do not have the CD, you can go to the manufacturer's website to obtain the drivers needed.
Question	How do I change the homepage in Internet Explorer?
Answer	In Internet Explorer, go to Properties, and change the homepage

Table 1: Examples from the QTLeap Corpus.

contact for troubleshooting, trying to offer a rapid reply and solution to not too complex questions from the users.

2.1 Gathering the corpus

The PcWizard incorporates an application to automatically answer simple requests from users. This process of providing support to end-users is implemented as a Question Answering (QA) application that supports it in preparing the replies to clients. Using techniques based on natural language processing, each query for help is matched against a memory of previous QAs, and, drawing from that repository, a list of possible replies is displayed, ranked by relevance according to internal heuristics of this support system. If the top reply scores over a threshold, it gets returned to the client. If the reply does not score over that threshold, a human operator is presented with the list of possible answers delivered by the system and he can either pick the most relevant or write a complete new answer (which will then be stored, and will thus contribute to extend the QA database).

The corpus was collected by selecting data contained in the database of the PcWizard application, where all the interactions with the clients are saved. The interactions that better support the automatic QA module were selected. Only interactions composed by one question and the respective answer were included in the corpus.

2.2 Characterization: examples and statistics

As became evident in the compilation of the corpus, the QTLeap corpus is characterized by short sentences, usually a request of help followed by an answer, and each conversation thread involves only two persons, the user and the operator. The request for help is often a well-formed question or a declarative sentence reporting a problem, but in a relevant number of cases, the question is not grammatically correct, presenting problems with coordination, missing verbs, etc. In some cases, the request is composed by a list of key words. This kind of utterance is quite frequent in informal communication via chat. On the other hand, a more formal register is to be found in the answers, as they are produced by well-trained operators and need to be very precise and concise in order to provide clarification to the user and to not lead to even more confusion. Table 1 shows some examples of interactions taken from the corpus.

Table 2 presents some statistics on the corpus focussing on the questions, answers, and the general composition. On average the questions are composed of just one sentence with a length of 12.6 tokens, while the answers comprise 1.5 sentences, with a length of 15 tokens. This means that an interaction is usually composed by two or three sentences.

	Tokens	Sentences	Tokens/Sentences	Sentences/Interactions
Questions	50905	4031	12.6	1
Answers	88536	5919	15	1.5
Total	139411	9959	14	2.5

Table 2: The QTLeap Corpus in numbers.

This kind of corpora is not very common, as most of the mainstream research is based on corpora using data sets composed by published texts, such as newspapers or books, or transcription of oral conversations. Furthermore, corpora with interrogatives are extremely rare, and most of them contain interrogatives that are artificially produced by manipulating sentences that were originally declarative ones.

In the last years, a few corpora were collected that consist of chat conversations via the internet. These corpora contain informal conversations about personal topics (Forsyth and Martell [2007]). Other corpora are more focused on technical topics such as the LINUX corpus (Elsner and Charniak [2010]), the IPHONE/PHYSICS/PYTHON corpus (Adams [2008]) and the Ubuntu chat corpus (Uthus and Aha [2013]). These corpora differ from the one presented here as they include large amounts of social conversations even though the chats used as sources for these corpora were initially intended only for tech support. In all the aforementioned corpora, the conversation threads involve several participants using an informal register. In almost all the cases (except for the Ubuntu corpus) the language addressed in these corpora is limited to English.

2.3 A parallel multilingual resource

The QTLeap corpus is a unique resource given that it is a multilingual data set with parallel utterances in different languages (Basque, Bulgarian, Czech, Dutch, English, German, Portuguese and Spanish), from four different language families (Basque, Germanic, Romance and Slavic). This multilingual resource was obtained by translating the original Portuguese corpus to the other languages. In particular, the Portuguese corpus was translated to the pivot language, which is English. The resulting English corpus was then translated into all the other languages covered in the project.

In order to translate the original corpus from Portuguese to English, several translation agencies were contacted. Each agency provided the translation of a small sample of the corpus, and in this way it was possible to select the best agency for the translation task. After this, all the corpus were revised internally by the HF project partner in order to ensure that the translation of technical terms was correct.

The translators were instructed to keep the informal register when translating from Portuguese to English, but to be precise regarding the terminology. We aimed to obtain a translation that is as close as possible to the original language, but still sounds natural to a native speaker of the target language.

In order to translate the English corpus to the other languages, a similar process was carried out. First several translation agencies were contacted and asked to provide the translation of a sample.

The translated sample was checked by the partner in charge for the respective language in the project. The best translation service was selected on the basis of that assessment.

The translation work was then carried out by at least two translators for each language, one performed the translation and the other revised it.

In order to ensure the quality of the translation, the corpus was divided in several batches. In this way it was possible to monitor the translation while it was being produced, and make any necessary adjustments.

The aligned corpus that was obtained was cleaned by removing extra spaces and non printable characters. The process of improvement is still going on, at a residual pace, by detecting and fixing possible remaining issues, such as small typos. Each partner keeps a register of the improvements made on its language specific part of the corpus and the corpus is eventually updated under new version numbers.

2.4 News corpus

In addition to the IT domain, which is considered the in-domain corpus for the project, the QTLeap MT services have been evaluated also on a News corpus (considered out-of-domain). In this deliverable a short description of the News corpus is added for the sake of completeness. The News corpus is presented also in Deliverable D3.13.

The annual workshops/conferences on Statistical Machine Translation (WMT, see http://www.statmt.org/) include translation tasks and several other tasks that are an important European benchmark of MT performance. The QTLeap consortium thus decided to evaluate of the Pilots also on WMT data, which is from the News domain. Since not all project languages were represented at WMT, the missing translations have been produced by professional translators.

To this end, 1104 English sentences and their corresponding human translations into Czech, German, and Spanish from the WMT 2012 and WMT 2013 translation tasks were taken as basis (the later years did not include Spanish translations). The sentences were chosen in such a way that their original source language was English, i.e., "reversed translations" originating from languages other than English that exist in the WMT datasets have been ignored. These 1104 English sentences were then professionally translated to Bulgarian, Dutch, Portuguese and Basque via a subcontract from QTLeap.

This data is released on META-SHARE[1] as well.

2.5 Updates, license and distribution

The QTLeap corpus in IT domain is available through the META-SHARE repository under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence.

The QTleap corpus in newsmedia domain is freely available through the META-SHARE repository.

For each language, the corpus is composed by two plain text files, one listing all the questions and the other listing all the answers. The two files are aligned, this means that the question in the first line of the file containing the questions corresponds to the answer in the first line of the file containing the answers. This correspondence holds across the different languages.

[1] QTLeap News Corpus: http://metashare.metanet4u.eu/go2/qtleap-news-corpus

3 Further curated LRTs for QTLeap Project

In this chapter, we present the LRTs that were curated to support the development of MT Pilots 1 towards Pilots 2 and 3 along a number of sections, each devoted to a specific language pair.

3.1 Basque to/from English

The translation pipeline for the Basque/English language pair uses TectoMT, which handles English analysis, English synthesis and tectogrammatical transfer. As a result, most of our effort in terms of LRTs has focused on Basque analysis and Basque synthesis. For Pilot 1, LRT curation involved (i) integrating existing analysis tools into Treex and (ii) implementing Basque synthesis in Treex.[2] This Section describes this curation.

UPV/EHU already provided tools that perform linguistic analysis covering all stages from tokenization to parsing, as well as tools for co-reference resolution, Semantic Role Labelling, and Word Sense Disambiguation, as described in Deliverable D2.4. For Pilot 1, we have integrated PoS tagging and dependency parsing and the remaining linguistic processes are directly performed by Treex blocks.

Treex contains tokenization and sentence splitting modules based on non-breaking prefixes. These blocks have been extended to work with Basque. To this end, a list of Basque non-breaking prefixes have been added.

After tokenization, UPV/EHU modules for PoS tagging, lemmatization and dependency parsing have been integrated into Treex. *ixa-pipe-pos-eu* (Alegria et al. [2002]), the PoS and lemmatization tool, which also performs tokenization and sentence splitting, has been modified to reuse previous tokenization. The tool now tokenizes on white-spaces only and sentence splitting considers each line in the input as a new sentence. Additionally, both ixa-pipe-pos-eu and the *ixa-pipe-dep-eu* dependency parser (based on MATE-tools (Bohnet and Kuhn [2012])) have been modified to accept NAF format in input/output to allow for pipeline executions similar to ixa-pipes tools (Agerri et al. [2014]) available for Spanish and English. Those modifications, undertaken in the project, have been integrated in the official distribution of the tools.

After applying these modifications, we integrated the tools as a wrapper block that, given a set of tokenized sentences, creates the appropriate input and calls the relevant tools. Once the tools finish their work, the analyses are read and loaded into Treex documents.

The analyses generated by UPV/EHU tools follow the guidelines of the Basque Dependency Treebank (BDT) corpus (Aduriz et al. [2003]) for both morphological tags and dependency tree structures. Therefore, to fully integrate the analyses into Treex, they must be modified to use the Interset tagset and follow Treex guidelines. To implement this change, we have used existing modules that have been improved for QTLeap purposes: (i) Interset driver for BDT tagset by Dan Zeman. It is published under open-source license (Perl Artistic + GPL) at CPAN[3]; (ii) Harmonization Treex block for BDT-style dependencies by Dan Zeman. It is available under open-source license (Perl Artistic + GPL) at QTLeap git repository[4]

[2] TectoMT and Treex described in other project deliverables, including Deliverable D2.4

[3] https://metacpan.org/source/ZEMAN/Lingua-Interset-2.041/lib/Lingua/Interset/Tagset/EU/Conll.pm

[4] https://github.com/ufal/treex/tree/master/libTreex/Block/HamleDT/EU/Harmonize.pm

For Basque synthesis, we have trained a model for Flect (Dušek and Jurcícek [2013]). Flect is a fully trainable morphological generation system aimed at robustness to previously unseen inputs, based on logistic regression and Levenshtein distance edit scripts between the lemma and the target word form. Given an already analyzed corpus, Flect is able to automatically learn the edits needed to generate the wordform based on the lemma and a set of morphological tags. Additionally, a number of Treex modules have been created to deal with diverse issues such as word order and capitalization.

The tools curated for Pilot 1 are published under open-source licenses: (1) ixa-pipe-pos-eu, the PoS tagger for Basque, is distributed under the GPLv3 license[5]. Note that ixa-pipe-pos-eu requires the reinstallation of several free tools, which have different licenses. Please refer to the installation instructions. (2) *ixa-pipe-dep-eu*, the Basque dependency parser, is distributed under the GPLv3 license[6]. (3) Treex and all the extensions developed for EN-EU are distributed under Perl Artistic + GPL licenses.

3.2 Bulgarian to/from English

For Bulgarian-English and English-Bulgarian Pilot 2 and 3 we have exploited Moses factor models over Word Sense Annotation and transfer of linguistic information on the basis of word alignments constructed during the translation process — see Deliverables D4.13 and D2.11. For Pilot 2 we proceeded using the Bulgarian pipeline described in D2.4 for performing POS tagging, lemmatization and dependency parsing of Bulgarian and IXA pipeline for the similar processing in English.

Ixa-pipes wrapper

For processing the English part of the data we have created a wrapper of several modules of the *ixa-pipes* system (Agerri et al. [2014]), which performs corresponding levels of analysis to the Bulgarian pipeline: sentence splitting, tokenization, lemmatization, part of speech (POS) tagging, and dependency parsing. These modules include *ixa-pipe-tok* (version 1.7.0), *ixa-pipe-pos* (version 1.3.3), and *IXA-EHU-srl* (version 1.0). The wrapper includes an additional module which generates factored output, suitable for use with a Moses factored system (Koehn and Hoang [2007]).

The first module, *ixa-pipe-tok*, takes as input a plain text document, and carries out rule-based tokenization and sentence segmentation.

The next step in the pipeline, *ixa-pipe-pos*, includes POS tagging and lemmatization. For tagging we have selected one of the provided POS models for English – Perceptron (Collins [2002]), which was trained using the WSJ treebank.

The last *ixa-pipes* module carries out dependency parsing. It is a wrapper of the English dependency parser and semantic role labeler of the *mate-tools* system (Bohnet [2010]). The module which was used for parsing is one of the provided models for English, which was trained on a concatenation of all of the CoNLL 2009 Shared Task (Hajič et al. [2009]) data sets for English.

The intermediate and final results of each of the *ixa-pipes* processing steps is stored in a NAF format. The wrapper provides the option to preserve the number of lines in the input and output English file. This option should be used when processing parallel corpora to ensure that the resulting factored output can be aligned to its corresponding Bulgarian file in case when the English file contains more than one sentence in certain lines.

[5] http://ixa2.si.ehu.es/ixa-pipes/eu/ixa-pipe-pos-eu.tar.gz
[6] http://ixa2.si.ehu.es/ixa-pipes/eu/ixa-pipe-dep-eu.tar.gz

For Pilot 3 we have modified the English pipeline exploiting the CoreNLP tools[7] for English. Thus, the analysis of English (tokenization, lemmatization, POS tagging and dependency parsing) as a source language was done with the CoreNLP tools[8] of Stanford University. The word sense disambiguation was done by the UKB tool.[9] The MRS structures and the post-processing rules were implemented in the CLaRK System.[10] There is no separate NLP pipeline for English, but the different components were used during different steps of Pilot 3 machine translation system. For details see Deliverable D2.11.

For the analysis of Bulgarian as a source language, we trained Mate tools[11] on the Bulgarian treebank. In order to adapt the processing to the domain, we have annotated Batch1 and Batch2 with morphosyntactic information.

3.3 Czech to/from English

Most of the tools used in Czech to/from English Pilot 1 experiments had existed at ÚFAL, CUNI. Some of the better known tools in this set are MorphoDiTa and NameTag, which have been integrated into Treex for the purpose of Pilot 1. The Treex wrappers are available under open-source license (Perl Artistic + GPL).[12]

The tools themselves are also open source (LGPL) and available from GitHub or http://www.lindat.cz/.

MorphoDiTa, Morphological Dictionary and Tagger, is an open-source tool for morphological analysis of natural language texts. It performs morphological analysis, morphological generation, tagging, and tokenization and is distributed as a standalone tool or a library, along with trained linguistic models. In the Czech language, MorphoDiTa achieves state-of-the-art results (Straková et al. [2014]) with a throughput around 10-200K words per second. MorphoDiTa is free software under the LGPL license and the linguistic models are free for non-commercial use and distributed under the CC BY-NC-SA license, although for some models the original data used to create the model may impose additional licensing conditions.

NameTag is an open-source tool for named entity recognition (NER). It identifies proper names in text and classifies them into predefined categories, such as names of persons, locations, organizations, etc. NameTag is distributed as a standalone tool or a library, along with trained linguistic models. In the Czech language, NameTag achieves state-of-the-art performance (Straková et al. [2014]). NameTag is a free software under the LGPL license and the linguistic models are free for non-commercial use and distributed under the CC BY-NC-SA license, although for some models the original data used to create the model may impose additional licensing conditions.

During the development of Pilot 3, a new tool was created – UDPipe ([?]),[13] which builds upon MorphoDiTa, but handles the whole annotation pipeline in the Universal

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

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

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

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

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

[12] https://github.com/ufal/treex/tree/master/lib/Treex/Tool/Tagger/MorphoDiTa.pm
https://github.com/ufal/treex/tree/master/lib/Treex/Block/W2A/TagMorphoDiTa.pm
https://github.com/ufal/treex/tree/master/lib/Treex/Tool/NER/NameTag.pm
https://github.com/ufal/treex/tree/master/lib/Treex/Block/A2N/NameTag.pm

[13] http://ufal.mff.cuni.cz/udpipe and http://hdl.handle.net/11234/1-1702

Dependencies (UD) style of dependency trees including: segmentation, tokenization, morphological analysis, tagging, lemmatization, and labelled parsing. Pretrained models are available for over thirty languages. We have integrated UDPipe into Treex.[14]

We have created and published a new parallel Czech-English corpus CzEng 1.6 ([**?**]).[15] It consists of about 0.5 billion words ("gigaword") in each language and 62 million sentence pairs, so it is four times bigger than the previous version. The corpus is equipped with automatic annotation at a deep syntactic level of representation (tectogrammatical layer) and alternatively in Universal Dependencies (and CoNLL-U format).

3.4 Dutch to/from English

For the Dutch-English and English-Dutch pipeline, a combination of Treex and Alpino is used.

From English to Dutch, English analysis, transfer, and conversion to Dutch a-trees is performed in Treex. A special block in Treex then converses those a-trees to Alpino-style dependency trees. These are then input to Alpino, and Alpino produces full sentences.

From Dutch to English, Alpino is used for Dutch analysis. A special Treex block takes care to map the Alpino-style dependency structures to a-trees. From there on, the Treex pipeline is used to map these to English sentences.

For the purpose of Pilot 1, a few new tools have been developed. These tools include the integration of Alpino in Treex, some new blocks in Treex, and some adaptations in Alpino.

The new Treex components include:

- Interset driver for Dutch CGN/Lassy/Alpino-style tagset by Dan Zeman and Ondrej Dusek. This driver is used in EN-NL and NL-EN Pilot1. It did not exist before QTLeap and was created only for QTLeap purposes. It is published under open-source license (Perl Artistic + GPL) at CPAN[16]

- Harmonization Treex block for Dutch Alpino-style trees by Ondrej Dusek and David Marecek. This block is used in EN-NL and NL-EN Pilot1. It did not exist before QTLeap and was created only for QTLeap purposes. It is available under open-source license (Perl Artistic + GPL) at the QTLeap git repository.[17]

  The following changes were made in Alpino for the sole purpose of the QtLeap Pilot 1. All resulting changes are integrated in the latest Alpino release, and freely available (including all sources) from the Alpino homepage.[18]

- Based on error-mining applied to the available development data, we dealt with a number of small issues in the lexicon, and in the Alpino grammar. One rule was added to the Alpino grammar for imperatives which are preceded by a modifier, for the Dutch equivalent of sentences such as "In the main menu, open the file-editor".

[14] https://github.com/ufal/treex/blob/master/lib/Treex/Tool/UDPipe.pm https://github.com/ufal/treex/blob/master/lib/Treex/Block/W2A/UDPipe.pm

[15] http://ufal.mff.cuni.cz/czeng

[16] https://metacpan.org/source/ZEMAN/Lingua-Interset-2.041/lib/Lingua/Interset/Tagset/NL/Cgn.pm

[17] https://github.com/ufal/treex/tree/master/libTreex/Block/P2A/NL/Alpino.pm

[18] http://www.let.rug.nl/vannoord/alp/Alpino

- The generation algorithm has been tuned and improved both for efficiency considerations and robustness considerations. If an input structure cannot be generated fully, the algorithm will generate each of the sub-structures. As a consequence, the current version of the Alpino generator is both more effective and (much!) faster than previous versions - not only on QtLeap data but also on other data-sets.

- For pilot 1, a large number of heuristic rules have been written in a new pre-processor for the Alpino generator, to map input dependency structures which are not fully consistent with expected input structures to more suitable input structures. The component now consists of over 3000 lines of Prolog code. For pilot 3, more rules have been added further improving the input dependency structures even more.

3.5 German to/from English

Apart from the processing pipeline of the transfer-based RBMT system (Lucy), additional deep processing tools for German are being used in several parts of the project, through the usage of specific parsing tools. A description of our efforts to adapt, evaluate, and enhance LRTs and processing tools for each use case is following:

3.5.1 The Berkeley Parser

The Berkeley Parser is a state-of-the-art Probabilistic Context-Free Grammar (PCFG) parser that supports unlexicalized parsing with hierarchically state-split PCFGs, supporting optimal pruning via a coarse-to-fine method (Petrov and Klein [2007]). It has the advantage of being is accurate and fast, by using multi-threading technology. Apart from the best tree for each parse, it also provides the parsing log-likelihood and a number of k-best trees along with their parse probabilities. The English grammar has been trained on the Wall Street Journal. The German grammar, using Latent Variable Grammars (Petrov and Klein [2008]), has been trained on the TIGER (Brants et al. [2004]) and TueBaD/Z (Telljohann et al. [2004]) treebanks, as released by the ACL 2008 workshop on Parsing German (Kübler [2008]).

The use of the Berkeley parser has shown good results as a quality indicator for Quality Estimation. In this context, our engineering efforts have focused on connecting the parser in the broader pipeline of sentence selection in Pilot 1, by providing a socket interface that export the Java library of the parser as a python object (see Py4J[19]).

3.5.2 Bilingual node alignments and CFG rules via IBM Model 1

Additionally, in our effort to acquire features for qualitative translation, we use word alignment methods based on the IBM Model 1, in order to map Berkeley tree node labels between source and produced translations. This resulted in 22,970 aligned tree node labels between English and German, extracted over a random sample of 55,544 sentences originating from the evaluation sets of WMT2008-WMT2015. Out of these, only 26 tree node labels are aligned in more than 10,000 sentences, 46 of them in more than 5,000 sentences, whereas 197 tree node labels are aligned in more than 1,000 sentences.

Additionally, we extracted the positions of the aligned tree node labels in the sentence, in order to assess possible re-ordering patterns. Finally, the single-side trees were parsed sequentially and the CFG rules from every parent node to its children were derived.

[19] http://py4j.sourceforge.net/

All the above resources were used as features for comparing translation performance on output by different systems, as part of the selection mechanism in Pilot 3.

3.5.3 BitPar

BitPar is a parser for highly ambiguous probabilistic context-free grammars. It makes use of bit-vector operations that allow parallelising and speeding up the the basic parsing operations (Schmid [2006]). The English grammar is based on the PENN treebank (Marcus et al. [1993]), whereas the German grammar is also based on the TIGER treebank. BitPar was also included on our annotation pipeline in order to provide additional evidence and allow comparisons to the observations on the Berkeley parses. It also provides sentence-level tree likelihood and k-best lists. The tree likelihood by BitPar on English translations was found (through Recursive Feature Elimination) more useful than the likelihood by Berkeley Parser, in terms of judging the quality of the machine translation outputs. Nevertheless, contrary to the Berkeley Parser, the k-best lists of BitPar were of limited usability due to the small differences in their relative likelihood.

3.5.4 ParZu

The Zurich Dependency Parser for German (ParZu) follows a hybrid architecture including both a hand-written grammar and a statistics module that chooses the most-likely parse of each sentence (Sennrich et al. [2009]). As compared to many other German parsers, it integrates morphological information (Sennrich et al. [2013]) and it does not use a chunker.

This parser has been employed for the parsing needs of the German version of TectoMT, which nevertheless remained on the level of a prototype and has not been included in any released QTLeap pilot. The parser has been chosen after an analysis of the capabilities of several parsers and their compiled grammars, including MDparser, Stanford Dependency Parser and MaltParser. ParZu was found optimal, as it provides the necessary morphological disambiguation upon parsing and connects well with relevant morphological analyzers and generators. Additionally, ParZu has shown to perform well in comparison to the other parsers in previous work (Williams et al. [2014]).

In order to acquire morphological analysis for ParZu, we have been using the Zurich Morphological Analyzer for German (ZMORGE), based on finite-state-transducers automatically extracted from Wiktionary (Sennrich and Kunz [2014]). The tool can also function as a morphological generator and outputs the analysis in a modified SMOR format.

As part of our QTleap efforts on TectoMT, a “driver” between the SMOR format and the universal Lingua Interset (Zeman [2008]) was built and commited to the open repository. This was required as the Lingua Interset is needed by TectoMT. In a later stage, this conversion can allow interaction with the Universal Dependencies standard (de Marneffe et al. [2014]).

3.6 Portuguese to/from English

The Portuguese translation pipeline uses TectoMT, which already handles English analysis, English synthesis and tectogrammatical transfer. Accordingly, as stated in Deliverable D1.3, most of our effort in terms of LRTs was directed at Portuguese analysis and Portuguese synthesis.

Regarding Pilot 1, LRT curation consisted mainly of (i) performing fixes and improvements to the tools that form the Portuguese pipeline, (ii) integrating those tools into Treex and (iii) implementing and fine-tuning Portuguese blocks in Treex.

A great deal of the Portuguese analysis and synthesis is handled by a set of processing tools, each with different licensing terms, that already existed when the project started. These tools are grouped under the LX-Suite of tools (Branco and Silva [2006]), which includes a sentence segmenter, a tokenizer, a POS tagger, morphological analysers for nominal and verbal categories, and a dependency parser. These tools have traditionally been run from the command line, and communicate among themselves using Unix pipes (i.e. the standard output of a process feeds into the standard input of the process that follows it). Using these tools from within Treex raised technical issues tied with how inter-process communication is handled by the operating system. Fixing these issues required changing some of the code that handles input and output, at the level of the tools, and disabling data-transfer buffering in pipes, at the level of inter-process communication in the shell. To ease integration of our analysis tools into Treex, LX-Suite was wrapped in a script that gives a convenient, configurable and unified access point to the pipeline. Communication with this script is done via a socket that provides a transparent way for accessing LX-Suite remotely and for distributing processing load.

LX-VerbalLemmatizer, the tool in the LX-Suite analysis pipeline that handles morphological analysis of verbal tokens, was found to be a performance bottleneck. The problem was tracked down to the way the tool interfaces with an auxiliary Python script. Fixing this issue lead to a great speed improvement for the tool and, consequently, for LX-Suite.

LX-Suite uses TnT Brants [2000] for POS tagging. We found a bug in TnT where a blank line, which is used for separating sentences, is disregarded. This behavior means that a sentence may be tagged differently depending on the sentence that precedes it. We fixed this issue by finding a token sequence that is guaranteed to be recognized by TnT as a sentence separator, seamlessly introducing that sequence into the input stream of TnT, annotating and then seamlessly removing it from the output.

The in-domain evaluation reported in Deliverable D5.4 uncovered some systematic errors in POS tagging. A common source of error was the tagging of unknown English words (e.g. “router” and “wifi” being tagged as verbs) and the tagging of the first word in the sentence, in particular when that word is an unknown imperative verb (e.g. “Abra” (open), “Carregue” (press), etc.) Such words and contexts are common in the in-domain corpus, but they hardly occur in the corpus used to train the POS tagger. Manually adding these words with the correct POS tag to the lexicon of TnT, which is stored as a human-readable file, was a straightforward way to make them known to the tagger. This, in turn, meant that the tagger was now able to tag those words correctly. A similar procedure was done for LX-VerbalLemmatizer. This tool uses a list of attested verb lemmas to which we added the neologism “clicar” (to click).

A bug in LX-VerbalLemmatizer was found, where “pretérito-mais-que-perfeito” verb forms were being assigned the wrong inflection features. That bug was fixed.

We have implemented a tool for converting from the Portuguese CINTIL-style dependencies into Universal Stanford Dependencies (USD). This allows us to obtain Tectogrammatical representations by using the converter from USD into Tectogrammatical that already exists in Treex as a stepping stone.

The CINTIL-USD conversion tool uses some conversion rules that need information about the semantic role of relations, but the default dependency parser outputs relations tagged only with grammatical functions. To allow applying these conversion rules, the dependency parser was retrained over a corpus of dependencies where grammatical relations are extended with semantic roles, for instance “SJ-ARG1” for a subject that is the first argument and “M-LOC” for a modifier that refers to a location. We also took this opportunity to use additional training data that became available since the last time the parser was trained, a total of $20,046$ sentences and $231,671$ tokens. Under 10-fold cross-validation, this parser achieves 0.86 accuracy (LAS, or labeled accuracy score).

The curation of the analysis pipeline also involved the implementation and tuning of several Treex blocks. These include, for instance, a block for reordering dependencies in English analysis; and a block in Portuguese analysis that fixes the representation of imperatives that handles Portuguese politeness and turns subjunctive mood into imperative.

Similarly, the curation of the transfer module also required the development of several Treex blocks. For instance, EN-PT transfer needed a module for moving adjectives to post-nominal position while PT-EN transfer required the converse module, for moving adjectives to pre-nominal position.

The synthesis of Portuguese is done in Treex, with the support of LX-Inflector (used for nominal inflection) and LX-Conjugator (used for verbal conjugation). These supporting tools also suffered from the pipe buffering issue described above, and had to be fixed. LX-Inflector and LX-Conjugator were also taken into the LX-Suite wrapper script mentioned above, which made their integration into Treex much easier.

Naturally, synthesis is a language-specific task and required implementing several Treex blocks from scratch, followed by testing and fine-tuning. These Treex blocks are responsible for such diverse issues as word order, ensuring proper capitalization of words, inserting clitic pronouns, inserting articles, forming contracted forms, etc.

We note that part of the LRT curation was done in colaboration with the CUNI partner. Namely, (i) the driver[20] for converting the Portuguese tagset into Interset, by Dan Zeman and Martin Popel; (ii) the harmonization (dependency style conversion) Treex block[21] for Portuguese USD-style dependencies, by Dan Zeman and Zdenek Zabokrtsky; and (iii) the draft harmonization Treex block[22] for Portuguese CINTIL-style dependencies, by Martin Popel, which ended up not being directly applied in Pilot 1, since we convert CINTIL dependencies to USD, but part of its code was used in the USD harmonization block in (ii).

The Treex blocks that were created and developed for QTLeap purposes are published under an open-source license (Perl Artistic + GPL).[23]

The LRTs for Pilot 2 are documented in deliverable D5.9. Here we summarize only the curation aspects related to those LRTs.

The LX-NER tool, for the recognition and classification of named entites, and which already existed when the project started, was added to the LX-Suite analysis pipeline and integrated into Treex. As with the tools used in Pilot 1, doing this required some adjustements to the code of the tool to solve technical issues related to inter-process communication.

[20] https://metacpan.org/source/ZEMAN/Lingua-Interset-2.041/lib/Lingua/Interset/Tagset/PT/Cintil.pm

[21] https://github.com/ufal/treex/tree/master/libTreex/Block/HamleDT/PT/HarmonizeCintilUSD.pm

[22] https://github.com/ufal/treex/tree/master/libTreex/Block/HamleDT/PT/HarmonizeCintil.pm

[23] The common address is https://github.com/ufal/treex/tree/master/libTreex/, and the different blocks can be found in subfolders A2T/PT, A2W/PT, T2A/PT, T2T/EN2PT e W2A/PT.

The MWN.PT wordnet, which already existed when the project started, is used as knowledge base in the WSD task. MWN.PT is stored as a SQL database so it had first to be converted to the Princeton WordNet 3.0 format used by the WSD tool. This required the implementation of a special-purpose conversion tool. The NED, WSD and coreference LRTs were developed in the course of the project and did not require any particular curation.

The LRTs for Pilot 3, the DeepBank and the accompaning deep lexicon, are documented in Deliverable D4.12. These LRTs have been developed in the course of the project and did not require any particular curation.

3.7 Spanish to/from English

Our translation pipeline uses TectoMT, which handles English analysis, English synthesis and tectogrammatical transfer. Therefore, most of our effort in terms of LRTs has focused on Spanish analysis and Spanish synthesis. For Pilot 1, LRT curation mainly involved (i) integrating existing analysis tools into Treex and (ii) implementing Spanish synthesis in Treex. This Section describes such curation.

The *ixa-pipes* tools[24] consist of a set of modules that perform linguistic analysis from tokenization to parsing. Additionally, a set of external tools have been adapted to interact with them[25] adding extra functionality such as co-reference resolution, Semantic Role Labelling, and Named Entity Disambiguation. For Pilot 1, the tokenization and sentence splitting modules of Treex have been adapted to Spanish. For PoS tagging (*ixa-pipe-pos*) and dependency parsing (*ixa-pipe-srl*) tools from ixa-pipes have been integrated.

Treex integrates tokenization and sentence splitting based on non-breaking prefixes. Those blocks have been extended to work with Spanish. To this end, a list of Spanish non-breaking prefixes were added.

After tokenization, ixa-pipes modules for PoS tagging, lemmatization and dependency parsing have been integrated into Treex. The tools were already developed and ready to use. We integrated them as a wrapper block that, given a set of already tokenized sentences, creates the appropriate input in NAF format and calls the relevant tools. Once the tools complete their work, the output of the system is read and loaded in Treex documents.

These analyses, generated by ixa-pipes tools, follow the AnCora guidelines both for morphological tags and dependency tree structures. Therefore, to fully integrate the analyses into Treex, they must be modified to use the Interset tagset and follow Treex guidelines. To implement this change, we have used existing modules that have been improved for QTLeap purposes: (i) Interset driver for Spanish AnCora Treebank tagset by Dan Zeman and Zdenek Zabokrtsky. It is published under open-source license (Perl Artistic + GPL) at CPAN[26]; (ii) Harmonization Treex block for Spanish AnCora-style dependencies by Dan Zeman, Zdenek Zabokrtsky and Martin Popel. It is available under open-source license (Perl Artistic + GPL) at QTLeap git repository[27].

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

[25] http://ixa2.si.ehu.es/ixa-pipes/third-party-tools.html

[26] https://metacpan.org/source/ZEMAN/Lingua-Interset-2.041/lib/Lingua/Interset/Tagset/ES/Conll2009.pm

[27] https://redmine.ms.mff.cuni.cz/projects/qtleap/repository/\discretionary{-}{}{}revisions/\discretionary{-}{}{}master/changes/\discretionary{-}{}{}treex/\discretionary{-}{}{}lib/\discretionary{-}{}{}Treex/Block/HamleDT/ES/Harmonize.pm

For Spanish synthesis a specific rule-based block that deals with morphological inflection has been created. This rule-based block correctly manages the regular inflection schemes, as well as the more usual exceptions. Given the large amount of resources needed to build a complete rule-based module for synthesis, the possibility to train an statistical morphological generator such as Flect will be studied for future pilots. Additionally, a number of Treex modules have been created to deal with diverse issues such as word order and capitalization.

All the tools curated for Pilot 1 are published under open-source licenses: (1) PoS tagger (*ixa-pipes-pos*) is distributed under Apache 2.0 license; (2) the dependency parser (*ixa-pipe-srl*) is based on Mate-tools which is distributed under GPLv3 license; (3) Treex and all the extensions developed for EN-ES are distributed under Perl Artistic + GPL licenses.

4 Final remarks

In D2.5 two situations of LRT curation were described for Pilot 1. In the first one, the LRTs existed prior to QTLeap project, and were improved within the project: Basque-English (both parts); Dutch-English (both parts); Spanish-English (both parts); Czech-English (both parts); German-English (German); Portuguese-English (both parts) Bulgarian-English (Bulgarian part). In the second one, there were no appropriate LRTs for the QTLeap objectives, but were newly created within the project: Bulgarian-English (English part); Dutch-English (both parts updated for QTLeap).

This deliverable outlined also the developments on the multilingual QTLeap corpus as well as the developments of LRTs after D2.5 per language pair.

The main directions of the additional developments were as follows: the creation of an out-of-domain QTLeap corpus in the newsmedia domain, addition of new training data, and the improvements over the NLP pipelines and the MT systems.

References

Paige Holland Adams. Conversation Thread Extraction and Topic Detection in Text-Based Chat. Master's thesis, Naval Postgraduate School, Monterey, California, 2008.

Itziar Aduriz, María Jesús Aranzabe, Jose Mari Arriola, Aitziber Atutxa, Arantza Díaz de Ilarraza, Aitzpea Garmendia, and Maite Oronoz. Construction of a Basque Dependency Treebank. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, pages 201–204, 2003.

Rodrigo Agerri, Josu Bermudez, and German Rigau. IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4.

Iñaki Alegria, María Jesús Aranzabe, Aitzol Ezeiza, Nerea Ezeiza, and Ruben Urizar. Robustness and Customisation in an Analyser/lemmatiser for Basque. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC). Customizing knowledge in NLP applications Workshop*, page 1—6, 2002.

Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In Chu-Ren Huang and Dan Jurafsky, editors, *COLING*, pages 89–97. Tsinghua University Press, 2010.

Bernd Bohnet and Jonas Kuhn. The Best of BothWorlds – A Graph-based Completion Model for Transition-based Parsers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 77–87, Avignon, France, April 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E12-1009.

António Branco and João Silva. A suite of shallow processing tools for portuguese: LX-Suite. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 179–182, 2006.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2 (4):597–620, 2004. ISSN 1570-7075. doi: 10.1007/s11168-004-7431-3.

Thorsten Brants. Tnt – a statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 224–231, Seattle, Washington, USA, April 2000. Association for Computational Linguistics. doi: 10.3115/974147.974178. URL http://www.aclweb.org/anthology/A00-1031.

Michael Collins. Discriminative training methods for Hidden Markov Models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 1–8. Association for Computational Linguistics, 2002.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of LREC*, 2014.

Ondrej Dušek and Filip Jurcícek. Robust Multilingual Statistical Morphological Generation Models. *ACL 2013*, pages 158–164, 2013.

Micha Elsner and Eugene Charniak. Disentangling chat. *Comput. Linguist.*, 36(3):389–409, September 2010. URL http://dx.doi.org/10.1162/coli_a_00003.

E.N. Forsyth and C.H. Martell. Lexical and discourse analysis of online chat dialog. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 19–26, Sept 2007.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 1–18, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-29-9. URL http://dl.acm.org/citation.cfm?id=1596409.1596411.

Philipp Koehn and Hieu Hoang. Factored translation models. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 868–876, 2007. URL http://www.aclweb.org/anthology/D07-1091.

Sandra Kübler. The PaGe 2008 Shared Task on Parsing German. In *Proceedings of the Workshop on Parsing German*, PaGe '08, pages 55–63, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-15-2. URL http://dl.acm.org/citation.cfm?id=1621401.1621409.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.*, 19(2):313–330, June 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972470.972475.

Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *Proceedings of the 2007 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York, 2007. Association for Computational Linguistics.

Slav Petrov and Dan Klein. Parsing German with Latent Variable Grammars. In *Proceedings of the Workshop on Parsing German*, PaGe '08, pages 33–39, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-932432-15-2. URL http://dl.acm.org/citation.cfm?id=1621401.1621406.

Helmut Schmid. Trace Prediction and Recovery with Unlexicalized PCFGs and Slash Features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 177–184, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220198. URL http://dx.doi.org/10.3115/1220175.1220198.

Rico Sennrich and Beat Kunz. Zmorge: A German Morphological Lexicon Extracted from Wiktionary. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014. ISBN 978-2-9517408-8-4. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/116_Paper.pdf.

Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. A new hybrid dependency parser for German. *Proceedings of the German Society for Computational Linguistics and Language Technology*, pages 115–124, 2009.

Rico Sennrich, Martin Volk, and Gerold Schneider. Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis. In *RANLP*, pages 601–609, 2013.

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.

Heike Telljohann, Erhard Hinrichs, Sandra Kübler, Ra Kübler, and Universität Tübingen. The Tüba-D/Z Treebank: Annotating German with a Context-Free Backbone. In *In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004*, pages 2229–2235, 2004.

David C. Uthus and David W. Aha. The ubuntu chat corpus for multiparticipant chat analysis. In *Analyzing Microtext, Papers from the 2013 AAAI Spring Symposium, Palo Alto, California, USA, March 25-27, 2013*, 2013. URL http://www.aaai.org/ocs/index.php/SSS/SSS13/paper/view/5706.

Rui Wang, Petya Osenova, and Kiril Simov. Linguistically-augmented Bulgarian-to-English Statistical Machine Translation Model. In *Proceedings of the Joint Workshop on Exploiting Synergies Between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, EACL 2012, pages 119–128, Stroudsburg, PA, USA, 2012a. Association for Computational Linguistics. ISBN 978-1-937284-19-0. URL http://dl.acm.org/citation.cfm?id=2387956.2387972.

Rui Wang, Petya Osenova, and Kiril Simov. Linguistically-enriched Models for Bulgarian-to-English Machine Translation. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, SSST-6 '12, pages 10–19, Stroudsburg, PA, USA, 2012b. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2392936.2392939.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Eva Hasler, and Philipp Koehn. Edinburgh's Syntax-Based Systems at WMT 2014. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 207–214, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W14/W14-3324.

Daniel Zeman. Reusable Tagset Conversion Using Tagset Drivers. In *Proceedings of LREC*, 2008.

A Summary of availability

Name of LRT	language	QTLeap	License	URL
Alpino	EN, NL	No	LGPL	http://www.let.rug.nl/vannoord/alp/Alpino
Berkeley Parser	DE	No	GNU GPL v2	https://code.google.com/p/berkeleyparser/
BitPar	DE	No		http://www.cis.uni-muenchen.de/~schmid/tools/BitPar/
Harmonization Treex block for Specific language style	EU, BG, CS, DT, EN, DE, PT, ES	Yes	Perl Artistic + GPL	https://redmine.ms.mff.cuni.cz/projects/qtleap/repository/revisions/master/show/treex/lib/Treex/Block/HamleDT
Interset driver for language specific tagset	EU, BG, CS, DT, EN, DE, PT, ES	Yes	Perl Artistic + GPL	https://metacpan.org/source/ZEMAN/Lingua-Interset-2.041/lib/Lingua/Interset/Tagset/EU/Conll.pm
ixa-pipe-dep-eu	EU	Yes	GPL v3	http://ixa2.si.ehu.es/ixa-pipes/eu/ixa-pipe-dep-eu.tar.gz
ixa-pipe-pos	ES	No	Apache 2.0	https://github.com/ixa-ehu/ixa-pipe-pos/
ixa-pipe-pos-eu	EU	Yes	GPL v3	http://ixa2.si.ehu.es/ixa-pipes/eu/ixa-pipe-pos-eu.tar.gz
ixa-pipe-srl	ES	No	GPL v3	https://github.com/newsreader/ixa-pipe-srl
Treex	EU, BG, CS, DT, EN, DE, PT, ES	Yes	Perl Artistic + GPL	https://github.com/ufal/treex
LX-Suite of tools	PT	No	various, including MS-NC and proprietary	http://metashare.metanet4u.eu/ (search for “LX-Tokenizer” and “LX-Tagger”)
ParZu	DE	No	GNU GPL v2	https://github.com/rsennrich/parzu
QTLeap corpus	EU, BG, CS, DT, EN, DE, PT, ES	Yes	CC BY-NC-SA 4.0	http://metashare.metanet4u.eu/go2/qtleapcorpus
QTLeap News corpus	EU, BG, CS, DT, EN, DE, PT, ES	Yes	Available - freely available	http://metashare.metanet4u.eu/go2/qtleap-news-corpus
Wrapper for BTB pipeline and ixa-pipes	BG, EN	Yes	Web services	http://213.191.204.69:5080/smtbtbpipews/process curl -X POST –data-binary "BG text." http://213.191.204.69:9080/smtixapipews/process curl -X POST –data-binary "Testing the pipe."
Wrapper for MorphoDiTa	CS	Yes	Perl Artistic + GPL	https://redmine.ms.mff.cuni.cz/projects/qtleap/repository/revisions/master/changes/treex/lib/Treex/Tool/Tagger/MorphoDiTa.pm;
Wrapper for NameTag	CS	Yes	Perl Artistic + GPL	https://github.com/ufal/treex/tree/master/libTreex/Tool/NER/NameTag.pm

Table 3: Summary of publicly available LRTs mentioned in this deliverable. QTLeap column indicates with “yes” those LRTs which have been (partially) funded by QTLeap. QTLeap corpus is also available through CLARIN Lindat. (https://lindat.mff.cuni.cz/)