CUNI Experiments for WMT17 Metrics Task

David Mareček **Ondřej Bojar**
Ondřej Hübsch **Rudolf Rosa** **Dušan Variš**

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
surname@ufal.mff.cuni.cz, except hubschondrej@gmail.com

Abstract

In this paper, we propose three different methods for automatic evaluation of the machine translation (MT) quality. Two of the metrics are trainable on direct-assessment scores and two of them use dependency structures. The trainable metric AutoDA, which uses deep-syntactic features, achieved better correlation with humans compared e.g. to the chrF3 metric.

1 Introduction

With the ongoing research of the machine translation (MT) systems in the past the need for accurate automatic evaluation of the translation quality became unquestionable. Even though the human judgment of the MT system outputs still holds as the most reliable form of evaluation, the high cost of human evaluation together with the amount of time required for such evaluation makes human judgment unsuitable for large scale experiments where we need to evaluate many different system configurations in a relatively short timespan. An additional important limitation of human evaluation is that it cannot be exactly repeated. This led to development of various methods for automatic MT evaluation in the past with the aim to eliminate the need for the expensive human assessment of the developed MT systems.

In this paper we suggest three novel methods for automatic MT evaluation together with their direct comparison:

1. AutoDA: A linear regression model using semantic features trained on WMT Direct Assessment scores (Bojar et al., 2016) or HUMEseg scores (Birch et al., 2016).

2. TreeAggreg: N-gram based metric computed over aligned syntactic structures instead of the linear representation of the translated sentences.

3. NMTScorer: A neural sequence classifier which assigns correct/incorrect flags to the evaluated sentence segments.

Table 1 shows the main properties of the proposed methods. Some of them were mainly developed for Czech as the target language and were later modified to be applied to other languages. The differences in the data preprocessing and their impact on the resulting evaluator are also described in this paper.

2 AutoDA: Automatic Direct Assessment

AutoDA is a sentence-level metric trainable on any direct assessment scores. The metric is based on a simple linear regression combining several features extracted from the automatically aligned translation-reference pair. There may be also other established metrics within the features.

The training data with golden direct-assessment scores available are shown in Table 2.

We describe two variants. The first one works only on Czech and uses many semantic features based of rich Czech tectogrammatical annotation (Böhmová et al., 2003). The second one uses much fewer features, however, it is language universal and needs only a dependency parsing model available.

2.1 AutoDA Using Czech Tectogrammatics

This metric automatically parses the Czech translation candidate and the reference translation and uses various semantic features to compute the final score.

2.1.1 Word Alignment

AutoDA relies on automatic alignment between the translation candidate and the reference trans-

Method	Resource Type	Trainable	Metric Type
AutoDA	Monolingual/Bilingual*	Yes	Segment-level Linear Regression
TreeAggreg	Monolingual	No	Tree Segment-level ChrF**
NMTScorer	Bilingual	Yes	Segment-level Classification

Table 1: Overview of the examined methods. Currently, AutoDA uses only monolingual resources even though extracting additional features from the bilingual data (*) is possible. TreeAggreg can use any string-level metric for score computation instead of ChrF (**).

Dataset	Source	Target	# Sentences
WMT16 DAseg	TR/FI/CS/RO/RU/DE	EN	560
	EN	RU	
WMT15 DAseg	DE/RU/FI/CS	EN	500
	EN	RU	
WMT16 HUMEseg	EN	CS/DE/PL/RO	∼350

Table 2: Overview of the available data for training AutoDA.

lation. The easiest way of obtaining word alignments is to run GIZA++ (Och and Ney, 2000) on the set of sentence pairs. GIZA++ was designed to align documents in two languages and it can obviously also align documents in a single language, although it does not benefit in any way from the fact that many words are identical in the aligned sentences. GIZA++ works well if the input corpus is sufficiently large, to allow for extraction of reliable word co-occurrence statistics. While the test sets alone are too small, we have a corpus of paraphrases for Czech (Bojar et al., 2013). We thus run GIZA++ on all possible paraphrase combinations together with the reference-translation pairs we need to align and then extract alignments only for the sentences of interest.

2.1.2 Tectogrammatical Parsing

We use Treex[1] framework (Popel and Žabokrtský, 2010) to do the tagging, parsing and tectogrammatical annotation. Tectogrammatical annotation of sentence is a dependency tree, in which only content words are represented by nodes. The main label of the node is a tectogrammatical lemma – mostly the same as the morphological lemma, sometimes combined with a function word in case it changes its meaning. Other function words and grammatical features of the words are expressed by other attributes of the tectogrammatical node. An example of a pair of tectogrammatical trees is provided in Figure 1. The main attributes are:

- **tectogrammatical lemma (t-lemma)**: the lexical value of the node,
- **functor**: the semantic value of the syntactic dependency relation. Functors express the functions of individual modifications in the sentence, e.g. ACT (Actor), PAT (Patient), ADDR (Addressee), LOC (Location), MANN (Manner),
- **sempos**: semantic part of speech: n (noun), adj (adjective), v (verb), or adv (adverbial),
- **formeme**: morphosyntactic form of the node. The formeme includes for example prepositions and cases of the nouns, e.g. *n:jako+1* for nominative case with preposition *jako*.
- **grammatemes**: tectogrammatical counterparts of morphological categories, such as number, gender, person, negation, modality, aspect, etc.

2.1.3 Scores for Matching Attributes Ratios

Given the word- (or node-) alignment links between tectogrammatical annotations of the translation and reference sentences, we can count the percentage of links where individual attributes agree, e.g. the number of pairs of tectogrammatical nodes that have the same tectogrammatical lemma. These scores capture only a portion of what the tectogrammatical annotations offer, for instance, we they do not consider the structure of the trees at all. For the time being, we take these scores as individual features and use them in a combined model.

[1] http://ufal.mff.cuni.cz/treex

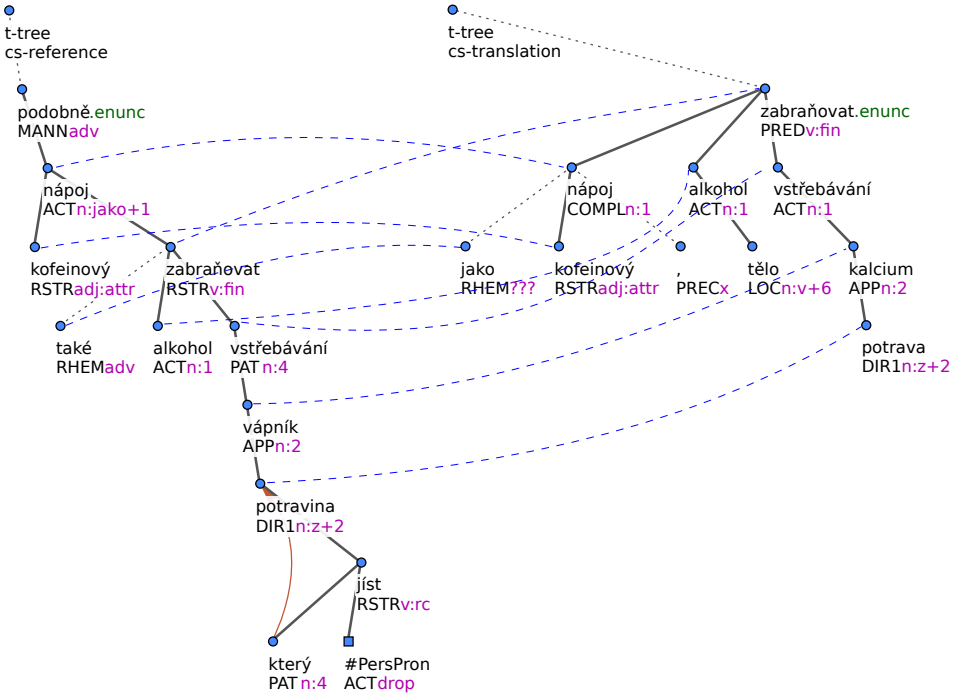

Figure 1: Example of aligned tectogrammatical trees of the reference *"Podobně jako kofeinový nápoj také alkohol zabraňuje vstřebávání vápníku z potravin, které jíme."* and the candidate translation *"Jako kofeinový nápoj, alkohol v těle zabraňuje vstřebávání kalcia z potravy."*

2.1.4 Linear Regression Training

We collect 83 various features based on matching tectogrammatical attributes computed on all nodes or a subsets defined by particular semantic part-of-speech tags. To this set of features, we add two BLEU scores (Papineni et al., 2002) computed on forms and on lemmas and two chrF3 scores (Popovic, 2015) computed on trigrams and sixgrams, so we have 87 features in total.

We train a linear regression model to obtain a weighted mix of features that fits best the WMT16 HUMEseg scores. Since the amount of annotated data available is low, we use the jackknife strategy:

- We split the annotated data into ten parts.
- For each tenth, we train the regression on all the rest data and apply it to this tenth.

By this procedure, we obtain automatically assigned scores for all sentences in the data. The correlation coefficients are shown in Table 3, along with the individual features.

In addition to the regression using all 87 features, we also did a feature selection, in which we manually chose only 23 features with a positive impact on the overall correlation score. For instance, we found that the BLEU scores can be easily omitted without worsening the correlation. Conversely, the chrF scores are very valuable and omiting them would lower the correlation significantly.

We see that chrF3 alone performs reasonably well (Pearson of 0.54), If we combine it with a selected subset our features, we are able to achieve the correlation of up to 0.659.

metric	en-cs
aligned-tnode-tlemma-exact-match	0.449
aligned-tnode-formeme-match	0.429
aligned-tnode-functor-match	0.391
aligned-tnode-sempos-match	0.416
lexrf-form-exact-match	0.372
lexrf-lemma-exact-match	0.436
BLEU on forms	0.361
BLEU on lemmas	0.395
chrF3	0.540
AutoDA (87 features)	0.625
AutoDA (selected 23 features)	**0.659**

Table 3: Selected Czech deep-syntactic features and their correlation against WMT16 HUMEseg dataset. Comparison with BLEU, chrF3, and our trainable AutoDA (using chrF3 as well).

2.2 Language Universal AutoDA

We have seen that deep-syntactic features help to train an automatic metric with higher correlation for Czech. Even though we have no similar tools for other languages so far, we try to extract similar features for them as well.

2.2.1 Universal Parsing

We use Universal Dependencies (UD) by Nivre et al. (2016b), a collection of treebanks in a common annotation style, where all our testing languages are present – version 1.3 covers 40 languages (Nivre et al., 2016a). For syntactic analysis, we use UDPipe by Straka et al. (2016), a tokenizer, tagger, and parser in one tool, which is trained on UD. The UD tagset consists of 17 POS tags; the big advantage is that the tagset is the same for all the languages and therefore we can easily extract e.g. content words, prepositional phrases, etc.

2.2.2 Monolingual Alignment

Unlike from Czech, we did not known about the existing corpus of paraphrases available across other languages,[2] so we used a simple monolingual aligner based on word similarities and relative positions in the sentence. Our implementation is inspired by the heuristic Monolingual Greedy Aligner written by Martin Popel (Rosa et al., 2012), which is available in the Treex framework.[3]

First, we compute scores for all possible alignment connections between tokens of the reference and translated sentence:

$$\begin{aligned} score(i,j) = w_1 &\text{JaroWinkler}(W_i^t, W_j^r) \\ &+ w_2 I(T_i^t = T_j^r) \\ &+ w_3(1 - |(i/len(t) - j/len(r)|), \end{aligned} \quad (1)$$

where JaroWinkler(W_i^t, W_j^r) defines similarity between the given words (Winkler, 1990), $I(T_i^t = T_j^r)$ is a binary indicator testing the identity of POS tags, and $(1 - |(i/len(t) - j/len(r)|)$ tells us how close are the two words according to their relative positions in the sentences. The weights were set manually to $w_1 = 8$, $w_2 = 3$, and $w_3 = 3$; they were not tuned for this specific task. When we have the scores, we can simply produce unidirectional alignments (i.e. find the best token in the translation for each token in the reference and vice versa) and then symmetrize them to create intersection (one-to-one) or union (many-to-many) alignments. We finally use union symmetrization, since it achieved slightly better correlation with humans.

2.2.3 Extracting Features

We distinguish content words from function ones by the POS tag. The tags for nouns (NOUN, PROPN), verbs (VERB), adjectives (ADJ), and adverbs (ADV) correspond more or less to content words. Then there are pronouns (PRON), symbols (SYM), and other (X), which may be sometimes content words as well, but we do not count them. The rest of POS tags represent function words.

Now, using the alignment links and the content words, we can compute numbers of matching content word forms and matching content word lemmas. The universal annotations contains also morphological features of words: case, number, tense, etc. Therefore, we also create equivalents of tectogrammatical formemes or grammatemes. Our features can thus check for instance the percentage of aligned words with matching morphological number or tense.

2.2.4 Regression and Results

We compute all the scores proposed in the previous section on the four languages and test the correlation on WMT16 HUMEseg dataset (Birch et al., 2016). German UD annotation does not contain lemmas and morphological features, so some scores for German could not be computed.

Similarly as in Section 2.1.4, we trained a linear regression on all the features together with *chrF3* score. The results computed by 10-fold cross-validation on WMT16 HUMEseg dataset and comparison with chrF and NIST[4] scores is shown in Table 4.

3 Tree Aggregated Evaluation

TreeAggreg is a simple sentence-level metric, remotely inspired by HUME. Rather than being a full standalone metric, it can be regarded as

[2] Multilingual corpus of paraphrases has been released by Chris Callison-Burch's group and is available here: http://paraphrase.org/#/download

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

[4] Unlike in previous experiment, we compare the results using NIST rather than BLEU since it is better suited for segment-level evaluation.

metric	en-cs	en-de	en-pl	en-ro
NIST	0.436	0.481	0.418	0.611
NIST cased	0.421	0.481	0.410	0.611
chrF1	0.505	0.497	0.428	0.608
chrF3	0.540	0.511	0.419	0.638
NIST on content lemmas	0.416	–	0.361	0.542
matching lemmas	0.431	–	0.393	0.565
matching forms	0.372	0.478	0.405	0.576
matching content lemmas	0.359	–	0.408	0.536
matching content forms	0.321	0.470	0.427	0.552
matching formemes	0.347	0.170	0.357	0.420
matching tense	-0.094	–	-0.118	0.079
matching number	0.286	–	0.205	0.404
AutoDA (linear regression)	**0.604**	**0.525**	**0.453**	**0.656**

Table 4: Pearson correlations of different sentence-level metrics on WMT16 HUMEseg dataset. Standard NIST and chrF metrics are compared with our individual features matching. AutoDA combines all the features together with the chrF3 score and the NIST score computed on content lemmas only. Other NIST scores are not included in AutoDA, since they do not bring any improvement.

a *metric template*, for in principle, any string-based MT metric can be plugged into it; we used chrF3 (Popovic, 2015) in our work.

In TreeAggreg, we are trying to improve an existing string-based metric by applying it in a syntax-tree-based context. This is motivated by our belief that dependency trees are a good means of capturing sentence structure, which may be relevant for MT evaluation metrics, as the MT output should presumably transfer the information present in the source sentence into a similar syntactic structure as the reference translation uses. However, in string-based MT metrics, the syntactic structure of a sentence is typically ignored.

In our rather light-weight attempt to employ syntactic analysis in MT evaluation, we segment the sentences into phrases based on their dependency parse trees, and evaluate these phrases independently with the string-based MT metric. The resulting scores are then aggregated into a final sentence-level score using a simple weighted average.

Our source codes are available online.[5]

3.1 Method

To be able to apply TreeAggreg to measuring the correspondence of a translation t to the reference r, we first need to apply a set of NLP tools in a pre-processing pipeline:

1. align reference and translation
2. parse reference
3. parse translation

We use the monolingual aligner presented in Section 2.2.2, using the unidirectional alignment from reference to translation; i.e. for each reference word we get exactly one translation word aligned to it (not necessarily unique). We use the UDPipe tool to provide the dependency parse trees (see Section 2.2.1).

Next, both the reference and the translation are split into the following types of segments:

1. the whole sentence (s_r, s_t)
2. the sentence root (r_r, r_t)
3. for each immediate dependent (d_r^i, d_t^i) of the root, the continuous span defined by its subtree (p_r^i, p_t^i)

Whole sentence This is simply the base string-based MT metric applied in the standard way.

Sentence root The sentence root is selected according to the parse trees; usually this is the main verb in the sentence.

Subtree spans As we expect the dependency analysis of the reference to be much more accurate than that of the translation, we only use the reference parse tree to identify the root dependents'

[5] https://github.com/ufal/auto-hume/tree/rudolf

spans, and the word alignment to identify the corresponding spans in the translation:

- p_r^i contains all words from s_r that are transitively dependent on d_r^i, the ith dependent of r_r; p_r^i includes d_r^i but excludes r_r
- p_t^i contains the first and last word from s_t which are aligned to any of the words in p_r^i, and all of the words between them

The string-level metric $m(r, t)$ is then computed on each corresponding pair of the reference and translation segments. A weighted average of the segment-level scores is computed, where longer segments are given higher weight: the weight is the sum of the numbers of words in the reference segment and in the translation segment. Additionally, for the (s_r, s_t) segment pair, which is still the most important component of the metric, we use a double weight. Thus, the final score m is computed as follows:

$$
\begin{aligned}
m_s &= m(s_r, s_t) \cdot (|s_r| + |s_t|) \cdot 2 \\
m_r &= m(r_r, r_t) \cdot 2 \\
m_p^i &= m(p_r^i, p_t^i) \cdot (|p_r^i| + |p_t^i|) \\
m &= \frac{m_s + m_r + \sum_{i \in Dep(r_r)} m_p^i}{2|s_r| + 2|s_t| + 2 + \sum_{i \in Dep(r_r)} |p_r^i| + |p_t^i|}
\end{aligned}
$$

$Dep(r_r)$ are all immediate dependents of r_r.

3.2 Development

When developing the TreeAggreg metric, we tried multiple configurations, evaluating each of them on the WMT16 HUMEseg dataset for correlation with human judgments, and then selected the one that performed best, which we have just described.

For example, we also experimented with more fine-grained segmentations, such as taking each node together only with its immediate dependents as a span. However, such setups performed poorer, probably because they depend more heavily on the high structural similarity of the translation to the reference. Still, it seems reasonable to assume that at least the arguments of the root node should usually correspond well between the reference and the candidate translation.

We also tried to put more weight to certain words that we expected to be more important, such as d_r^i (immediate dependents of the root r_r) However, this always led to a deterioration in the correlation of the metric to human judgments. Thus, an important property of our metric seems to be that each reference word is taken into account exactly twice.[6]

Lang.	chrF3	TreeAggreg	Difference
en-cs	0.5403	**0.5473**	+0.0070
en-de	**0.5111**	0.5078	−0.0033
en-pl	0.4186	**0.4266**	+0.0080
en-ro	**0.6314**	0.6226	−0.0088
Average	0.5254	**0.5261**	+0.0007

Table 5: Evaluation of TreeAggreg (our metric) and chrF3 (baseline) with Pearson's correlation to human judgments.

3.3 Evaluation

To evaluate our metric, we measured Pearson's correlation of chrF3-based TreeAggreg scores with sentence-level human judgments on the WMT16 HUMEseg dataset. For comparison, we also measure the correlation of a baseline metric, which is the vanilla sentence-level chrF3.

As shown in Table 5, our metric performs comparably to the chrF3 baseline, leading to a slight improvement for two language pairs, and a slight deterioration for the other two.

Thus, our approach of employing sentence syntactic structure into a string-based MT metric seems to affect the metric only minimally. Moreover, the TreeAggreg metric was developed and evaluated on the same data and therefore the comparison in Table 5 is not quite fair, however, the number of configurations tested was very little.

4 Neural MT Scorer

Neural MT Scorer is a model that predicts a probability for a given source/target translation pair using a simplified architecture that is based on existing NMT models with attention. The predicted number should reflect how much the meaning of source and target matches. We used that model for a different task (scoring phrase table entries in PBMT) where it performed well. Note that as of now, Neural MT Scorer indeed does not make any use of the reference translation, so it is effectively a quality estimation method.

The training data for the model are bilingual corpus (set of sentences that should be classified

[6] The same holds for words in the translation only if the p_t^i spans do not overlap, are contiguous, include both the first and the last word in the sentence, and do not include r_r.

as entirely correct) as well as a set of sentences that should be classified as incorrect (we obtain these by performing some random operations on the bilingual corpus). We do not train it on data specific for the metrics task (i.e. the model is only trained to recognize correct and incorrect translations, but small differences among different translations of the same sentence might not be recognized), therefore there is a room for potential improvement.

4.1 Architecture

We use two LSTM encoders, one for source and one for target side. The vector representations of the source words are fed into the source LSTM encoder to obtain one representation p_s of the entire sentence. Also, the intermediate outputs of the source LSTM encoder are used in an attentional layer when processing the target sentence in the target LSTM encoder. The final cell states p_s and p_t are used to measure the bilingual similarity by $\sigma(p_s^T p_t)$. The entire architecture is very similar to (Bahdanau et al., 2014), except that we use the attention mechanism while encoding the target side. Note that there is also no softmax layer over the word dictionary – we know the entire source and target sentences and so we do not need to predict the next word; we just need one score between 0 and 1. This should allow for faster training of the model; however, we need to provide labeled training data. We currently generate wrong sentences using these basic operations:

- change a few words to completely random ones from the source/target dictionary
- take a translation of a completely different sentence
- utilize WordNet to change the polarity of a sentence
- remove/add some random words at a random place

4.2 Evaluation

We evaluated the model on the WMT16 HUMEseg dataset, but currently it performs poorly. It should be possible to improve it significantly by optimizing the training process for the metrics task (for example by adding another layer that uses the final representations p_s and p_t to predict human scores and finetune the entire model on some manually evaluated datasets). The Pearson correlation coefficients to human judgements are shown in Table 6.

Languages	NMT Scorer
en-cs	0.4099
en-de	0.3462
en-pl	0.3261
en-ro	0.4792
Average	0.3903

Table 6: Evaluation of NMT Scorer with Pearson correlation to human judgments.

5 Conclusion

We presented three metrics. AutoDA is a trainable metric combining syntactic features matching and chrF and naturally significantly outperforms chrF on all four tested languages.

In TreeAggreg, we tried to enrich a string-based MT metric with light-weight information about the syntactic structure of the sentences, but the results seem rather disappointing.

NMTScorer in which we used two LSTM encoders for source sentence and candidate translation and predicted sentence similarity also did not prove to work well.

Acknowledgments

This work has been in part supported by the European Union's Horizon 2020 research and innovation programme under grant agreements No 644402 (HimL) and 645452 (QT21), by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16_013/0001781), by the grant No. DG16P02B048 of the Ministry of Culture of the Czech Republic and by the SVV 260 453 grant.

References

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

Alexandra Birch, Omri Abend, Ondrej Bojar, and Barry Haddow. 2016. HUME: Human UCCA-based evaluation of machine translation. *arXiv preprint arXiv:1607.00030* .

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague dependency treebank. In *Treebanks*, Springer, pages 103–127.

Ondřej Bojar, Matouš Macháček, Aleš Tamchyna, and Daniel Zeman. 2013. Scratching the Surface of Possible Translations. In *Proc. of TSD 2013*. Západočeská univerzita v Plzni, Springer Verlag, Berlin / Heidelberg, Lecture Notes in Artificial Intelligence.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 131–198. http://www.aclweb.org/anthology/W/W16/W16-2301.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Yevgeni Berzak, Riyaz Ahmad Bhat, Cristina Bosco, Gosse Bouma, Sam Bowman, Gülşen Cebirolu Eryiit, Giuseppe G. A. Celano, Çar Çöltekin, Miriam Connor, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, T. Erjavec, R. Farkas, J. Foster, D. Galbraith, S. Garza, F.and Goenaga I. Ginter, K. Gojenola, M. Gokirmak, Y. Goldberg, X. Gómez Guinovart, B. Gonzáles Saavedra, N. Grūzītis, B. Guillaume, J. Hajič, D. Haug, B. Hladká, R. Ion, E. Irimia, A. Johannsen, H. Kaşkara, H. Kanayama, J. Kanerva, B. Katz, J. Kenney, S. Krek, V. Laippala, L. Lam, A. Lenci, N. Ljubešić, O. Lyashevskaya, T. Lynn, A. Makazhanov, C. Manning, C. Mărănduc, D. Mareček, H. Martínez Alonso, J. Mašek, Y. Matsumoto, R. McDonald, A. Missilä, V. Mititelu, Y. Miyao, S. Montemagni, K. S. Mori, S. Mori, K. Muischnek, N. Mustafina, K. Müürisep, V. Nikolaev, H. Nurmi, P. Osenova, L. Øvrelid, E. Pascual, M. Passarotti, C. Perez, S. Petrov, J. Piitulainen, B. Plank, M. Popel, L. Pretkalnia, P. Prokopidis, T. Puolakainen, S. Pyysalo, L. Ramasamy, L. Rituma, R. Rosa, S. Saleh, B. Saulīte, S. Schuster, W. Seeker, M. Seraji, L. Shakurova, M. Shen, N. Silveira, M. Simi, R. Simionescu, K. Simkó, K. Simov, A. Smith, C. Spadine, A. Suhr, U. Sulubacak, Z. Szántó, T. Tanaka, R. Tsarfaty, F. Tyers, S. Uematsu, L. Uria, G. van Noord, V. Varga, V. Vincze, Jing Xian Wang, J. N. Washington, Z. Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2016a. Universal dependencies 1.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. http://hdl.handle.net/11234/1-1699.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016b. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Portorož, Slovenia.

Franz Josef Och and Hermann Ney. 2000. A Comparison of Alignment Models for Statistical Machine Translation. In *Proceedings of the 17th conference on Computational linguistics*. Association for Computational Linguistics, pages 1086–1090.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02, pages 311–318. https://doi.org/10.3115/1073083.1073135.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: Modular NLP framework. In Hrafn Loftsson, Eirikur Rögnvaldsson, and Sigrun Helgadottir, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL 2010)*. Iceland Centre for Language Technology (ICLT), Springer, Berlin / Heidelberg, volume 6233 of *Lecture Notes in Computer Science*, pages 293–304.

Maja Popovic. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the 10th Workshop on Statistical Machine Translation (WMT-15)*. pages 392–395.

Rudolf Rosa, Ondřej Dušek, David Mareček, and Martin Popel. 2012. Using parallel features in parsing of machine-translated sentences for correction of grammatical errors. In *Proceedings of Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-6), ACL*. ACL, Jeju, Korea, pages 39–48.

Milan Straka, Jan Hajič, and Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Paris, France.

William E. Winkler. 1990. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods (American Statistical Association)*. pages 354–359.