OpenSubtitles2018: Statistical Rescoring of Sentence Alignments in Large, Noisy Parallel Corpora

Pierre Lison[*], **Jörg Tiedemann**[†], **Milen Kouylekov**[‡]
[*]Norwegian Computing Center
plison@nr.no

[†] Department of Modern Languages, University of Helsinki
jorg.tiedemann@helsinki.fi

[‡] Department of Informatics, University of Oslo
milen.kouylekov@usit.uio.no

Abstract

Movie and TV subtitles are a highly valuable resource for the compilation of parallel corpora thanks to their availability in large numbers and across many languages. However, the quality of the resulting sentence alignments is often lower than for other parallel corpora. This paper presents a new major release of the OpenSubtitles collection of parallel corpora, which is extracted from a total of 3.7 million subtitles spread over 60 languages. In addition to a substantial increase in the corpus size (about 30 % compared to the previous version), this new release associates explicit *quality scores* to each sentence alignment. These scores are determined by a feedforward neural network based on simple language-independent features and estimated on a sample of aligned sentence pairs. Evaluation results show that the model is able predict lexical translation probabilities with a root mean square error of 0.07 (coefficient of determination R^2 = 0.47). Based on the scores produced by this regression model, the parallel corpora can be filtered to prune out low-quality alignments.

Keywords: Parallel Corpora, Machine Translation, Bitext alignment

1 Introduction

Movie and TV subtitles are used in a wide range of language technology applications. Their availability in a large number of languages makes them well-suited for the creation of parallel multilingual corpora. These corpora are a central resource for learning machine translation models (Koehn, 2009) but can also be used for corpus-driven lexicography, cross-lingual NLP or translation research (Paetzold, 2016; Akbik et al., 2016; Mikhailov and Cooper, 2016). Recent work on neural conversation models also showed that subtitles can be used to train dialogue agents (Vinyals and Le, 2015; Lison and Bibauw, 2017).

Parallel corpora derived from subtitles have a number of benefits. The first one is their size: the OpenSubtitles dataset is (to the best of our knowledge) the world's largest open collection of parallel corpora. The latest release, which is presented in this paper, contains no less than 3.4 billion sentences (amounting to 22.2 billion tokens) spread over 60 languages and a total of 1782 language pairs. As subtitles are annotated with timestamps, they can also be efficiently synchronised using a linear-time algorithm (Tiedemann, 2008). Finally, their conversational nature make them ideal for exploring dialogue phenomena and properties of everyday language (Paetzold and Specia, 2016; van der Wees et al., 2016).

However, the extraction of parallel corpora from subtitles must also face some challenges. One difficulty stems from the fact that subtitles are typicall not direct translations of one another. Rather, they should better be viewed as boiled-down *transcriptions* of the same conversations across several languages. Subtitles will inevitably differ in how they "compress" the conversations, notably due to structural divergences between languages, cultural differences and disparities in subtitling traditions/conventions. As a consequence, sentence alignments extracted from subtitles often have a higher degree of insertions and deletions compared to alignments derived from other sources.

We present in this paper a new release of the OpenSubtitles collection. In addition to increasing the global volume of the dataset (+30 % of the total number of sentences), the release includes several technical improvements in the preprocessing and alignments of the sentences. The most important improvement is the calculation of explicit *quality scores* for all sentence pairs. As explained in Section 3, these quality scores are determined by a neural model based on simple features extracted from the sentence pairs. This regression model is fitted based on a sample of sentence pairs and can be subsequently applied to the full collection of bilingual corpora.

The paper is organised as follows. Section 2 describes the preprocessing and alignment steps involved in compiling the parallel corpora, while Section 3 presents the alignment (re)scoring model. Section 5 provides our conclusions.

2 Dataset

2.1 Source Data

The raw data consists of a full database dump of the OpenSubtitles website[1], encompassing a total of 3.98 million subtitle files. In addition to the files themselves, the database dump contains information about the source material (through IMDB identifiers[2]), the subtitling language and format (usually `.srt` format), as well as miscellaneous meta-data such as the upload date and user ratings. The dataset covers a total of 208 K movies or TV episodes (as determined by their IMDb identifier). 69 % of the IMDb

[1] http://www.opensubtitles.org

[2] http://www.imdb.com

Language	2016 release Subtitle files	2018 release Subtitle files	Covered IMDbs	Sentences	Tokens
Afrikaans	32	63	57	61.3K	450K
Albanian	3.0K	3.1K	2.0K	3.6M	24.4M
Arabic	67.3K	94.1K	45.0K	83.6M	458M
Armenian	1	9	9	4.1K	33.0K
Basque	188	0.9K	0.9K	1.0M	5.8M
Bengali	76	0.5K	440	0.7M	3.7M
Bosnian	30.5K	37.3K	21.1K	34.1M	216M
Breton	32	32	28	23.1K	165K
Bulgarian	90.4K	108K	59.1K	94.6M	0.6G
Catalan	0.7K	0.8K	0.8K	0.6M	4.7M
Chinese (simplified)	22.4K	29.1K	15.2K	31.2M	191M
Chinese (traditional)	6.7K	9.9K	6.0K	10.7M	66.2M
Bilingual Chinese-English	4.5K	8.6K	4.7K	9.2M/8.3M	56.0M/73.9M
Croatian	96.8K	126K	52.9K	113M	0.7G
Czech	125K	157K	63.8K	136M	0.9G
Danish	24.1K	32.4K	19.5K	30.2M	208M
Dutch	98.2K	125K	58.9K	105M	0.8G
English	322K	447K	140K	441M	3.2G
Esperanto	89	103	95	93.1K	0.6M
Estonian	23.5K	28.8K	16.2K	27.5M	168M
Finnish	44.6K	64.4K	45.2K	52.0M	282M
French	105K	127K	66.7K	107M	0.8G
Galician	370	449	424	309K	2.4M
Georgian	271	293	268	281K	1.7M
German	27.7K	46.5K	34.5K	41.6M	288M
Greek	114K	143K	61.1K	126M	0.9G
Hebrew	79.7K	98.7K	43.5K	87.5M	0.5G
Hindi	57	102	92	144K	1.0M
Hungarian	99.3K	131K	66.7K	104M	0.6G
Icelandic	1.3K	1.5K	1.3K	1.9M	12.2M
Indonesian	11.0K	21.6K	12.2K	22.8M	138M
Italian	96.5K	135K	55.8K	105M	0.8G
Japanese	2.6K	3.5K	3.0K	3.2M	23.7M
Kazakh	0	4	4	4.1K	19.8K
Korean	0.7K	2.2K	1.9K	2.3M	10.2M
Latvian	392	493	459	0.6M	3.5M
Lithuanian	1.5K	2.0K	1.8K	2.1M	11.6M
Macedonian	5.6K	7.9K	4.6K	7.9M	50.3M
Malay	1.0K	3.2K	2.2K	3.8M	22.9M
Malayalam	251	421	379	0.5M	2.8M
Norwegian	8.9K	14.2K	11.8K	13.0M	86.8M
Persian	6.5K	12.2K	8.0K	13.0M	78.8M
Polish	161K	279K	66.5K	237M	1.4G
Portuguese	96.3K	131K	48.1K	118M	0.8G
Portuguese (BR)	220K	289K	101K	252M	1.7G
Romanian	162K	205K	72.3K	193M	1.3G
Russian	38.7K	56.0K	39.9K	44.9M	291M
Serbian	148K	180K	67.8K	168M	1.1G
Sinhalese	0.5K	0.9K	0.8K	1.0M	5.7M
Slovak	14.7K	18.1K	12.4K	16.1M	104M
Slovenian	52.6K	60.4K	27.1K	59.6M	361M
Spanish	192K	234K	91.7K	214M	1.5G
Swedish	27.3K	41.1K	26.2K	36.2M	245M
Tagalog	52	60	59	19.3K	130K
Tamil	17	32	30	40.2K	206K
Telugu	20	22	22	30.4K	160K
Thai	10.2K	11.0K	5.8K	9.1M	18.8M
Turkish	159K	189K	65.0K	173M	1.0G
Ukrainian	1.0K	1.6K	1.4K	1.3M	7.9M
Urdu	14	35	32	46.5K	358K
Vietnamese	3.1K	5.2K	4.1K	5.1M	41.9M
Total	2.8M	3.7M		3.4G	22.2G

Table 1: Statistics for the 60 languages in the extracted corpus. The *subtitles files* corresponds to the number of converted subtitles (which may be lower than the number of raw subtitles in the database due to discarded files). The *covered IMDbs* represent the number of distinct movies or TV episodes (denoted by their IMDb identifier) covered by the subtitles.

identifiers are associated with subtitles in at least two languages and 29 % with at least 10 languages.

2.2 Preprocessing

A number of steps are required to preprocess the subtitle files, as detailed in (Lison and Tiedemann, 2016):

1. **Format conversion**: The `.srt` subtitles are parsed to extract their constitutive blocks. This step includes detecting the file encoding.

2. **Sentence segmentation**: There no one-to-one correspondence between sentences and subtitle blocks displayed on the screen, as illustrated in this small example (where the first sentence is spread over 2 subtitle blocks, while the third block contains 2 sentences):

```
140
00:07:12,502 --> 00:07:15,812
Quando abbiamo estratto l'energia
blu positiva dal frammento

141
00:07:15,902 --> 00:07:19,019
ci siamo ritrovati con questo
sottoprodotto altamente instabile.

142
00:07:19,102 --> 00:07:21,935
- l'energia rossa negativa.
- Ah, quella mi piace.
```

   The sentences are segmented using language-specific heuristics based on punctuation markers, time gaps between blocks, and capitalisation.

3. **Tokenisation**: Once the sentences are segmented, they are tokenised, using either the tokenisation scripts from Moses (Koehn et al., 2007) or the Kytea toolkit for Chinese word segmentation (Neubig et al., 2011).

4. **Correction of OCR errors**: Some subtitles are extracted from video streams using OCR (Optical Character Recognition), generating a number of recognition errors. A noisy channel approach is presented in (Lison and Tiedemann, 2016) to correct these errors based on language models derived from the Google N-grams. This spellchecking model is also used here with some minor improvements to better handle e.g. accented characters and proper names. In total, more than 9 million tokens were corrected using this approach (with 4 million tokens just for English).

5. **Inclusion of meta-data**: Finally, the subtitles are enriched with meta-data extracted from IMDb, providing details such as the film genre and the original (spoken) language of the movie or TV episode. The new release contains additional information such as the version number of the subtitles and flags indicating (a) whether the subtitle is intended for hearing-impaired audiences and (b) whether the subtitles were generated automatically using machine translation.

After preprocessing, we obtain a total of 3.7 million subtitles (180 thousand subtitles were discarded due to formatting errors or erroneous meta-data). Each subtitle is encoded in a separate XML file including the tokenised sentences (annotated with timestamps) together with meta-data about the subtitle and its associated movie / TV episode.

2.3 Alignment

Sentence alignment is done using the time-overlap algorithm proposed by (Tiedemann, 2008). The procedure searches for the alignment that maximises the time-overlap between aligned units based on the time stamps given in the subtitles. Time information is extrapolated in correlation to string length in cases where it is not available at the sentence boundary. To further improve the synchronisation, we use lexical cues to estimate offset and speed parameters using bilingual dictionaries extracted from word-aligned subtitles (Tiedemann, 2008). In contrast to our earlier releases, we now also keep alignments between alternative subtitle files besides the ones that show the best match according to an overlap measure. Those alternative links are stored in separate alignment files and may be used to complement the selection of top-ranked subtitle pairs. Furthermore, intralingual links will also be offered again based on the procedures of (Tiedemann, 2016).

3 Rescoring model

As mentioned in the introduction, sentence alignments extracted from subtitle are often less literal than alignments from other types of bilingual corpora. Subtitle must indeed obey strong space and time constraints: a maximum of two lines with at most 40-50 characters per line and an on-screen display between 1 and 6 seconds (Aziz et al., 2012). Subtitles must therefore be crisp and boil down the spoken conversations to a small number of words. The everyday language used in subtitles also leaves more room for translation choices than technical or legal texts. Here are two examples of non-literal alignments:

English:	Oh, I bet it does
French:	Le contraire m'aurait surpris.
	[The contrary would have surprised me]
Arabic:	أنت الأهدأ في مركز العاصفةً
	[You are the calmest in the center of the storm]
Spanish:	Dijeron que no tenías nervios.
	[They said you had no nerves.]

These less literal alignments (along with other types of misalignments due to e.g. timing differenes) may lead to problems for downstream NLP tasks. Fortunately, some surface cues that can be exploited to predict whether a sentence pair is closely aligned or not. For instance, a large difference in the number of tokens in the source and target language may be indicative of a low-quality alignment. On the other side, the presence of cognates or the use of identical punctuation markers increases the likelihood of a good alignment.

3.1 Measures of alignment quality

The first step towards building the rescoring model is to determine a measure of alignment quality that can be used

as target variable. The approach used in this paper relies on extracting a sample of sentence pairs from the bilingual corpora, computing their lexical translation probabilities (in both directions) based on existing lexical translation tables and using these probabilities as a proxy for the alignment quality[3]. More specifically, we rely on the expectation formula of IBM Model 1 (Brown et al., 1993) to compute the log-probabilities of the target sentence t given the source s and the source s given the target t:

$$\log P(s|t) = \alpha \sum_{j=1}^{l_s} \log \left(\sum_{i=0}^{l_t} t(s_j|t_i) \right) \quad (1)$$

$$\log P(t|s) = \alpha \sum_{j=1}^{l_t} \log \left(\sum_{i=0}^{l_s} t(t_j|s_i) \right) \quad (2)$$

In the two formulae above, α represents a normalising factor and $t(x|y)$ the translation probability of token x from token y given by the lexical translation table. As done in IBM Model 1, s_0 and t_0 represent a default "null" value allowing for tokens to appear in the target without direct equivalent in the source language. To obtain the lexical translation tables, word alignments are first generated by running GIZA (Och and Ney, 2000) on the existing bitexts from OpenSubtitles and estimating the probability of each word pair through maximum likelihood.

In order to be useful as measures of alignment quality, the log-probabilities in (1) and (2) must, however, be slightly modified. First, the log-probabilities will typically be lower for a long source sentence than for a short one, since the number of translation choices increases with the sentence length. This is unfortunate, as we do not want to penalise long sentence pairs in the scoring model. To address this issue, the log-probabilities in (1) and (2) are divided by the length of the source sentence, such that the average log-probability remains roughly constant as a function of the sentence length. We can then define a raw score of alignment quality between the two sentences s and t as the minimum of these two rescaled log-probabilities:

$$\text{score}_{\text{raw}}(s,t) = \min \left(\frac{\log P(t|s)}{l_s}, \frac{\log P(s|t)}{l_t} \right) \quad (3)$$

Furthermore, the lexical translation probabilities may also vary according to the language pair. This variation may be due to the size of the bitext on which the translation tables were trained (a larger bitext will lead to a higher number of alternative translations for each token), or to the linguistic distance between the source and target language. To avoid penalising language pairs with lower average translation probabilities, the raw scores of (3) are rescaled separately for each language pair using quantile normalisation (Bolstad et al., 2003). Quantile normalisation is a non-linear transformation that matches the quantiles of the original distribution to the quantiles of a target distribution, in this case a normal distribution. After this quantile transform, the scores are mapped to a range of $[0, 1]$. The final scores are thus computed as:

$$\text{score}_{\text{final}}(s,t) = \text{scale}_{L_s,L_t} \left(\text{score}_{\text{raw}}(s,t) \right) \quad (4)$$

where scale_{L_s,L_t} is the quantile transform of the raw scores for the source and target languages L_s and L_t followed by rescaling to [0,1]. A score of 0.5 will therefore indicate a sentence pair whose log-probabilities (per token) revolve around the arithmetic mean for that particular language pair.

3.2 Features

Three families of features are extracted from the sentence alignments:

- Features extracted at the level of sentence pairs, such as the ratio of sentence length (measured in number of tokens or characters) in the source and target, or the use of similar punctuations.
- Features extracted at the subtitle level, such as the number of empty alignments or the ratio between the total number of tokens in the two subtitles.
- Features extracted from meta-data, in particular the languages used, movie or TV genre, release year and user rating.

These three families of features are enumerated in Table 2. All features are rescaled separately for each language pair. This per-language rescaling is necessary since the distribution of many features will correlate with the language pair – for instance, the number of cognates will be higher for Spanish-Catalan than for Arabic-Chinese. It is worth noting that the features defined in Table 2 are all based on simple, surface-level measures that do not rely on external language resources or NLP tools. This is important as many of the languages found in the OpenSubtitles corpus have relatively few available linguistic resources.

4 Evaluation

4.1 Model selection

A set of 8.3 million sentence pairs was extracted from the OpenSubtitles corpus, covering 760 distinct language pairs. This set corresponds to 0.24 % of the total number of sentences in the corpus. The features and quality scores were extracted for these sentence pairs based on the approach described in the previous section. Several machine learning models for regression were tested:

- Ridge regression, which is a simple linear model with L_2 regularisation.
- Lasso regression, another linear model with L_1 regularisation.
- Gradient boosting, which builds a predictor from an ensemble of simpler models, here regression trees.

[3] A previous version of this paper used position-independent word error rates computed from Google translations as response variable. However, the average error rates were too high to be practically useful to estimate the quality of sentence pairs.

Feature	Description
Features extracted from the sentence pair:	
tokens_{ratio,ndiff}	Ratio and normalised difference between the number of tokens in source and target sentences.
chars_{ratio,ndiff}	Ratio and normalised difference between the nb. of characters in the two sentences.
nb_identical_{all,cap}	Nb. of identical tokens in the two sentences, considering all tokens or only capitalized tokens.
nb_cognates	Nb. of near-identical tokens (shared substring of at least 4 characters) in the two sentences.
nb_corrected	Nb. of tokens corrected by the spellchecker(s) in the two sentences.
same_timings	Whether the start and end timestamps of the sentences in the two subtitles are identical or not.
time_overlap	Time overlap between the source and target sentences (as given by the timestaps).
nb_aligns	Total nb. of sentences in the pair (2 for 1:1 alignments, 3 for 1:2, etc.)
final_punct	Whether the final punctuation is the same in the source and target sentence or not.
llcsr	Length of the longest common substring between the two sentences, normalised by length.
nonalpha_seq	Length of longest common subsequence for non-letter characters (punctuation and numbers).
Features extracted at subtitle-level:	
sentences_{ratio,ndiff}	Ratio and normalised difference between the total number of sentences in the two subtitles.
tokens_{ratio, ndiff}	Ratio and normalised difference between the total number of tokens.
duration_{ratio, ndiff}	Ratio and normalised difference between the duration (in seconds)
corrected_words	Total number of tokens corrected by the spellcheckers (when available).
unknown_words	Total number of tokens unknown to the spellcheckers (when available).
ratio_{0:1,1:1,1:2,1:3}	Ratio of alignments of a particular type among the full list of alignments.
Meta-data features:	
language	Source and target languages (one-hot encoding)
genre	Movie or TV genre (one-hot encoding).
year	Release year of movie or TV episode.
original	Original language used in the movie or TV episode (one-hot encoding)
MT-translated	Whether zero, one or both subtitles are marked as translated by MT engines
confidence	Confidence score from language identification tool on the subtitles
rating_{min,max,avg}	Minimum, maximum and average rating of the two subtitles.

Table 2: Features used in the scoring model.

- Feedforward neural networks (multilayer perceptron) including either one or two hidden layers.

The performance of these models are evaluated through 10-fold cross validation, using the mean-square error, root-mean-square error and coefficient of determination (R^2) as evaluation metrics. The baseline is simply the prediction of the mean value for the quality score.

Table 3 summarises the results. We can observe that the best performing model is a feedforward neural network with two hidden layers of 100 units each. The neural network obtains a R^2 for coefficient of determination for 0.47, which means that 47 % of the variance in the quality score can be predicted from the input features using this model. The good performance of neural networks seems to indicate the presence of complex, non-linear relations between the features and the quality score which cannot be accounted for by simpler models. The bad performance of Lasso regression (which favours using a small number of features due to the L_1 regularisation) also shows that there is no single feature that works as a good predictor for the task.

Once learned on the dataset of 8.3 M sentence pairs, the

Model	MSE	RMSE	R^2
Baseline (predict mean)	0.009	0.096	0.0
Lasso regression ($\alpha = 0.01$)	0.008	0.092	0.091
Lasso regression ($\alpha = 0.001$)	0.006	0.081	0.303
Ridge regression ($\alpha = 1$)	0.006	0.077	0.356
Gradient boosting (10 regression trees)	0.007	0.085	0.224
Feedforward NN (one hidden layer, dim=100)	0.005	0.071	0.457
Feedforward NN (two hidden layers, dim=100)	**0.005**	**0.070**	**0.470**

Table 3: Evaluation results for various machine learning models on the task of predicting the value of $\text{score}_{\text{final}}(s,t)$ from the features listed in Table 2. MSE stands for mean-square error, RMSE for the root mean-square error and R^2 for coefficient of determination.

neural network can then be straightforwardly applied on the full list of aligned sentences in OpenSubtitles in order to assign each alignment to a quality score.

We also conducted a small-scale manual analysis of the quality scores predicted by the neural network. Here are a few examples of alignments assigned to a low quality score (<0.25) by the neural network:

Afrikaans:	Kalmeer
	[Calm down]
Polish:	Dlatego byłem w Wiedniu.
	[That's why I was in Vienna]
Bosnian:	Tačno tako
	[Exactly]
Danish:	Og du er tidligere straffet?
	[And you had previous convictions?]
Greek:	Θεέ μου
	[Oh my god]
Portuguese:	Residência Mainwaring.
	[Mainwaring Residence.]
German	(Mystische Musik)
	[(Mystical music)]
Turkish	Lordum...
	[My Lord...]

4.2 MT Experiments

In this section we look at machine translation models trained on filtered data sets in order to test the impact of rescoring on a downstream task. For this purpose, we apply an attentional sequence-to-sequence model implemented in the Helsinki neural MT system (Östling et al., 2017) with byte-pair encoding (BPE) to the language pairs of news translation task at WMT 2017. Our models are trained on OpenSubtitles data only from the 2018 release presented in this paper and we leave subtitles released in 2017 as held-out data from which we extract 10,000 sentences and their alignments as an in-domain test set for each language pair. We restrict the test data to one-to-one sentence pairs with a time-overlap of over 80% to reduce noise in the data.

Using this setup we can now compare two different systems for each language pair: One that is trained on all data (excluding the heldout data) and one that is trained on filtered data using a rescoring threshold of 0.6. We then apply both models to in-domain test data from the subtitle corpus and to out-of-domain news data from WMT 2017.

All systems apply the same setup with the same number of training batches without any language-specific tuning of any of them. In particular, we use 256 dimensions for word embeddings, 512 dimensions for the hidden LSTM-layer in the encoder, 1024 dimensions for the decoder LSTM's and 256 dimensions for the attention layer. We use savepoint ensembling of the last five models (each of them saved after 5,000 batches of size 16) and stop training after 45,000 batches. We use a vocabulary of 50,000 items for each language and split the data using BPE (Sennrich et al., 2016) trained on the subtitle trainng data with 50,000 merge operations. We also apply the hybrid encoder of HNMT to avoid unknown words in translation.

Note that the results on news data will be much lower than official results from WMT 2017 due to the domain mismatch and limited training that we apply in our experiments. We only give BLEU scores here to provide some indication of the translation quality that can be expected. Further analyses of the MT results is outside of the scope of this paper. Table 4 summarises the results of our experiments.

	2016		2018		filtered	
system	subs	news	subs	news	subs	news
en-cs	28.36	12.02	**28.76**	**12.94**	28.35	12.05
en-fi	23.51	11.00	24.00	11.13	**24.12**	**11.49**
en-de	28.71	14.48	**28.92**	**16.07**	**28.92**	14.71
en-ru	23.21	14.21	**23.74**	**15.94**	23.68	15.25
en-tr	**18.67**	6.46	18.58	**7.36**	18.24	6.81
cs-en	38.14	17.18	38.34	**17.26**	**38.37**	16.90
fi-en	26.58	13.80	26.94	10.77	**27.08**	**15.88**
de-en	33.02	18.88	**33.40**	19.16	33.01	**19.24**
ru-en	30.52	18.40	30.15	17.67	**30.58**	**18.71**
tr-en	**25.84**	10.34	25.64	**10.79**	25.32	10.65

Table 4: NMT models trained on subtitle data with and without filtering. BLEU scores in % on heldout data and news test data from WMT 2017.

We can see that the differences are small and the effect of filtering is not always beneficial. There are several reasons why the results are inconclusive. First of all, our training procedures were rather limited and all models have only seen a fraction of the entire data. Furthermore, our choice of leaving out all subtitles from 2017 as heldout data removed a large portion of the additional data that we include in our new release and the positive effect is not as visible as it could be. Finally, we did not perform a systematic study on optimising the threshold for filtering the training data. It is possible that too much of the valuable training data is left out and the coverage is reduced.

5 Conclusion

This paper presented OpenSubtitles2018, a new major release of the OpenSubtitles collection of parallel corpora. One important addition to this release is the estimation of a quality score associated with each sentence pair. The scores are determined through a feedforward neural network estimated from a sample of sentence pairs.

To train the neural network, lexical translation probabilities are computed for each sentence pair and employed as an indirect measure of alignment quality. The approach relies on simple, generic features such as the relative sentence length, number of empty alignments, number of cognates, or similar punctuation.

The neural network is able to explain 47 % of the variance of the quality score based on these features. A subsequent manual analysis also showed that sentence pairs assigned with a low quality score were indeed the result of misalignments. However, initial experiments with neural machine translation models do not demonstrate a conclusive advantage to filtering the bitexts based on these quality scores, suggesting that more work is needed to find the right filtering threshold.

References

Akbik, A., Kumar, V., and Li, Y. (2016). Towards semi-automatic generation of proposition banks for low-resource languages. In *EMNLP*, pages 993–998.

Aziz, W., de Sousa, S. C. M., and Specia, L. (2012). Cross-lingual sentence compression for subtitles. In *16th Annual Conference of the European Association for Machine Translation (EAMT 2012)*, pages 103–110, Trento, Italy.

Bolstad, B. M., Irizarry, R. A., Åstrand, M., and Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C. J., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, pages 177–180, Prague, Czech Republic.

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Lison, P. and Bibauw, S. (2017). Not all dialogues are created equal: Instance weighting for neural conversational models. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2017)*, pages 384–394, Saarbrücken, Germany. ACL.

Lison, P. and Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Mikhailov, M. and Cooper, R. (2016). *Corpus Linguistics for Translation and Contrastive Studies: a guide for research*. Routledge.

Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 529–533, Portland, Oregon, USA.

Och, F. J. and Ney, H. (2000). Giza++: Training of statistical translation models.

Östling, R., Scherrer, Y., Tiedemann, J., Tang, G., and Nieminen, T. (2017). The Helsinki neural machine translation system. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September. Association for Computational Linguistics.

Paetzold, G. and Specia, L. (2016). Collecting and exploring everyday language for predicting psycholinguistic properties of words. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1669–1679, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Paetzold, G. H. (2016). *Lexical Simplification for Non-Native English Speakers*. Ph.D. thesis, University of Sheffield.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Tiedemann, J. (2008). Synchronizing translated movie subtitles. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*.

Tiedemann, J. (2016). Finding alternative translations in a large corpus of movie subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC-2016)*.

van der Wees, M., Bisazza, A., and Monz, C. (2016). Measuring the effect of conversational aspects on machine translation quality. In *COLING*, pages 2571–2581.

Vinyals, O. and Le, Q. (2015). A Neural Conversational Model. *CoRR*, abs/1506.05869.