Cognate-aware morphological segmentation for multilingual neural translation

Stig-Arne Grönroos
stig-arne.gronroos@aalto.fi
Aalto University, Finland

Sami Virpioja
sami.virpioja@aalto.fi
Aalto University, Finland
Utopia Analytics, Finland

Mikko Kurimo
mikko.kurimo@aalto.fi
Aalto University, Finland

Abstract

This article describes the Aalto University entry to the WMT18 News Translation Shared Task. We participate in the multilingual subtrack with a system trained under the constrained condition to translate from English to both Finnish and Estonian. The system is based on the Transformer model. We focus on improving the consistency of morphological segmentation for words that are similar orthographically, semantically, and distributionally; such words include etymological cognates, loan words, and proper names. For this, we introduce Cognate Morfessor, a multilingual variant of the Morfessor method. We show that our approach improves the translation quality particularly for Estonian, which has less resources for training the translation model.

1 Introduction

Cognates are words in different languages, which due to a shared etymological origin are represented as identical or nearly identical strings, and also refer to the same or similar concepts. Ideally the cognate pair is similar orthographically, semantically, and distributionally. Care must be taken with "false friends", i.e. words with similar string representation but different semantics. Following usage in Natural Language Processing, e.g. (Kondrak, 2001), we use this broader definition of the term cognate, without placing the same weight on etymological origin as in historical linguistics. Therefore we accept loan words as cognates.

In any language pair written in the same alphabet, cognates can be found among names of persons, locations and other proper names. Cognates are more frequent in related languages, such as Finnish and Estonian. These additional cognates are words of any part-of-speech, which happen to have a shared origin.

In this work we set out to improve morphological segmentation for multilingual translation systems with one source language and two related target languages. One of the target languages is assumed to be a low-resource language. The motivation for using such a system is to exploit the large resources of a related language in order to improve the quality of translation into the low-resource language.

Consistency of the segmentations is important when using subword units in machine translation. We identify three types of consistency in the multilingual translation setting (see examples in Table 1):

(i) The benefit of consistency is most evident when the translated word is an identical cognate between the source and a target language. If the source and target segmentations are consistent, such words can be translated by sequentially copying subwords from source to target.

(ii) Language-internal consistency means that when a subword boundary is added, its location corresponds to a true morpheme boundary, and that if some morpheme boundaries are left unsegmented, the choices are consistent between words. This improves the productivity of the subwords and reduces the risk of introducing short, word-internal errors at the subword boundaries. In the example **saami + miseksi*, choosing the wrong second morph causes the letters *mi* to be accidentally repeated.

(iii) When training a multilingual model, a third form of consistency arises between the different target languages. An optimal segmentation would maximize the use of morphemes with cross-lingually similar string rep-

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 390–397

https://doi.org/10.18653/v1/W18-6410

type	consistent	en	fi	et
(i)	yes	On + y + sz + kie + wicz	On + y + sz + kie + wicz	On + y + sz + kie + wicz
(ii)	yes	gett + ing	saa + mise + ksi	saa + mise + ks
		work + ing	toimi + mise + ksi	toimi + mise + ks
(iii)	yes	work time	työ + aja + sta	töö + aja + st
(i)	no	On + y + sz + kie + wicz	Onys + zk + ie + wi + cz	O + nysz + ki + ewicz
(ii)	no	get + ting	saami + seksi	saami + seks
		work + ing	toimi + mise + ksi	toimi + miseks
(iii)	no	work time	työ + aja + sta	tööajast

Table 1: Example consistent and inconsistent segmentations.

resentations and meanings, whether they occur in cognate words or elsewhere. We hypothesize that segmentation consistency between target languages enables learning of better generalizing subword representations. This consistency allows contexts seen in the high-resource corpus to fill in for those missing from the low-resource corpus. This should lead to improved translation results, especially for the lower resourced target language.

Naïve joint training of a segmentation model, e.g. by training Byte Pair Encoding (BPE) (Sennrich et al., 2015) on the concatenation of the training corpora in different languages, can only address consistency when the cognates are identical (type *i*), or with some luck if the differences occur in the ends of the words. If a single letter changes in the middle of a cognate, consistent subwords that span over the location of the change are found only by chance. In order to encourage stronger consistency, we propose a segmentation model that uses automatically extracted cognates and fuzzy matching between cognate morphs.

In this work we also contribute two new features to the OpenNMT translation system: Ensemble decoding, and fine-tuning a pre-trained model using a compatible data set.[1]

1.1 Related work

Improving segmentation through multilingual learning has been studied before. Snyder and Barzilay (2008) propose an unsupervised, Bayesian method, which only uses parallel phrases as training data. Wicentowski (2004) present a supervised method, which requires lemmatization. The method of Naradowsky and Toutanova (2011) is also unsupervised, utilizing a hidden semi-Markov model, but it requires rich features on the input data.

The subtask of cognate extraction has seen much research effort (Mitkov et al., 2007; Bloodgood and Strauss, 2017; Ciobanu and Dinu, 2014). Most methods are supervised, and/or require rich features.

There is also work on cognate identification from historical linguistics perspective (Rama, 2016; Kondrak, 2009), where the aim is to classify which cognate candidates truly share an etymological origin.

We propose a language-agnostic, unsupervised method, which doesn't require annotations, lemmatizers, analyzers or parsers. Our method can exploit both monolingual and parallel data, and can use cognates of any part-of-speech.

2 Cognate Morfessor

We introduce a new variant of Morfessor for cross-lingual segmentation.[2] It is trained using a bilingual corpus, so that both target languages are trained simultaneously.

We allow each language to have its own subword lexicon. In essence, as a Morfessor model consists of a lexicon and the corpus encoded with that lexicon, we now have two separate complete Morfessor sub-models. The two models are linked through the training algorithm. We want the segmentation of non-cognates to tend towards the normal Morfessor Baseline segmentation, but place some additional constraints on how the cognates are segmented.

In our first experiments, we only restricted the number of subwords on both sides of the cognate pair to be equal. This criterion was

[1] Our changes are awaiting inclusion in OpenNMT. In the mean time, they are available from https://github.com/Waino/OpenNMT-py/tree/ensemble

[2] Available from https://github.com/Waino/morfessor-cognates

too loose, and we saw many of the longer cognates segmented with both 1-to-N and N-to-1 morpheme correspondences. For example

ty	+	ö	+	aja	+	sta
töö	+	aja	+	s	+	t

To further encourage consistency, we included a third component to the model, which encodes the letter edits transforming the subwords of one cognate into the other.

Cognate Morfessor is inspired by Allomorfessor (Kohonen et al., 2009; Virpioja et al., 2010), which is a variant of Morfessor that includes modeling of allomorphic variation. Simultaneously to learning the segmentations, Allomorfessor learns a lexicon of transformations to convert a morph into one of its allomorphs. Allomorfessor is trained on monolingual data.

We implement the new version as an extension of Morfessor Baseline 2.0 (Virpioja et al., 2013).

2.1 Model

The Morfessor Baseline cost function (Creutz and Lagus, 2002)

$$\mathrm{L}(\boldsymbol{\theta}, \boldsymbol{D}) = -\log p(\boldsymbol{\theta}) - \log p(\boldsymbol{D} \mid \boldsymbol{\theta}) \qquad (1)$$

is extended to

$$\begin{aligned}\mathrm{L}(\boldsymbol{\theta}, \boldsymbol{D}) = &-\log p(\boldsymbol{\theta}_1) - \log p(\boldsymbol{\theta}_2) - \log p(\boldsymbol{\theta}_E) \\ &- \log p(\boldsymbol{D}_1 \mid \boldsymbol{\theta}_1) - \log p(\boldsymbol{D}_2 \mid \boldsymbol{\theta}_2) \\ &- \log p(\boldsymbol{D}_E \mid \boldsymbol{\theta}_E) \qquad (2)\end{aligned}$$

dividing both lexicon and corpus coding costs into three parts: one for each language ($\boldsymbol{\theta}_1, \boldsymbol{D}_1$ and $\boldsymbol{\theta}_2, \boldsymbol{D}_2$) and one for the edits transforming the cognates from one language to the other ($\boldsymbol{\theta}_E, \boldsymbol{D}_E$).

The coding is redundant, as one language and the edits would be enough to reconstruct the second language. In the interest of symmetry between target languages, we ignore this redundancy.

The intuition is that the changes in spelling between the cognates in a particular language pair is regular. Coding the differences in a way that reduces the cost of making a similar change in another word guides the model towards learning these patterns from the data.

The coding of the edits is based on the Levenshtein (1966) algorithm. Let (w^a, w^b) be a cognate pair and its current segmentation $\big((m_1^a, \ldots, m_n^a), (m_1^b, \ldots m_n^b)\big)$. The morphs are paired up sequentially. Note that the restrictions on the search algorithm guarantee that both segmentations contain the same number of morphs, n. For a morph pair (m_i^a, m_i^b), the Levenshtein-minimal set of edits is calculated. Edits that are immediately adjacent to each other are merged. In order to improve the modeling of sound length change, we extend the edit in both languages to include the neighboring unchanged character, if one half of the edit is the empty string ϵ, and the other contains another instance of character representing the sound being lengthened or shortened. This extension encodes a sound lengthening as e.g. 'a→aa' instead of 'ϵ →a'. As the edits are cheaper to reuse once added to the edit lexicon, avoiding edits with ϵ on either side is beneficial to reduce spurious use. Finally, position information is discarded from the edits, leaving only the substrings, separated by a boundary symbol.

As an example, the edits found between *yhteenkuuluvuuspolitiikkaa* and *ühtekuuluvuspoliitika* are 'y→ü', 'een→e', 'uu→u', 'ti→it', and 'kka→k'.

The semi-supervised weighting scheme of Kohonen et al. (2010) can be applied to Cognate Morfessor. A new weighting parameter *edit_cost_weight* is added, and multiplicatively applied to both the lexicon and corpus costs of the edits.

The training algorithm is an iterative greedy local search very similar to the Morfessor Baseline algorithm. The algorithm finds an approximately minimizing solution to Eq 2. The recursive splitting algorithm from Morfessor Baseline is slightly modified. If a non-cognate is being reanalyzed, the normal algorithm is followed. Cognates are reanalyzed together. Recursive splitting is applied, with the restriction that if a morph in one language is split, then the corresponding cognate morph in the other language must be split as well. The Cartesian product of all combinations of valid split points for both languages is tried, and the pair of splits minimizing the cost function is selected, unless not splitting results in even lower cost.

3 Extracting cognates from parallel data

Finnish–Estonian cognates were automatically extracted from the shared task training data. As we needed a Finnish–Estonian parallel data set, we generated one by triangulation from the English–Finnish and English–Estonian parallel data. This resulted in a set of 679 252 sentence pairs (ca 12 million tokens per language).

FastAlign (Dyer et al., 2013) was used for word alignment in both directions, after which the alignments were symmetrized using the *grow-diag-final-and* heuristic. All aligned word pairs were extracted based on the symmetrized alignment. Words containing punctuation, and pairs aligned to each other fewer than 2 times were removed. The list of word pairs was filtered based on Levenshtein distance. If either of the words consisted of 4 or fewer characters, an exact match was required. Otherwise, a Levenshtein distance up to a third of the mean of the lengths, rounding up, was allowed. This procedure resulted in a list of 40 472 cognate pairs. The list contains words participating in multiple cognate pairs. Cognate Morfessor is only able to link a word to a single cognate. We filtered the list, keeping only the pairing to the most frequent cognate, which reduces the list to 22 226 pairs.

The word alignment provides a check for semantic similarity in the form of translational equivalence. Even though the word alignment may produce some errors, accidentally segmenting false friends consistently should not be problematic.

4 Data

After filtering, we have 9 million multilingual sentence pairs in total. 6.3M of this is English–Finnish, of which 2.2M is parallel data, and 4.1M is synthetic backtranslated data. Of the 2.8M total English–Estonian, 1M is parallel and 1.8M backtranslated. The sentences backtranslated from Finnish were from the news.2016.fi corpus, translated with a PB-SMT model, trained with WMT16 constrained settings. The backtranslation from Estonian was freshly made with a BPE-based system similar to our baseline system, trained on the WMT18 data. The sentences were selected from the news.20{14-17}.et corpora, using a language model filtering technique.

4.1 Preprocessing

The preprocessing pipeline consisted of filtering by length[3] and ratio of lengths[4], fixing encoding problems, normalizing punctuation, removing of rare characters[5], deduplication, tokenizing, truecasing, rule-based filtering of noise, normalization of contractions, and filtering of noise using a language model.

The language model based noise filtering was performed by training a character-based deep LSTM language model on the in-domain monolingual data, using it to score each target sentence in the parallel data, and removal of sentences with perplexity per character above a manually picked threshold. A lenient threshold[6] was selected in order to filter noise, rather than for aiming for domain adaptation. The same process was applied to filter the Estonian news data for backtranslation.

Our cognate segmentation resulted in a target vocabulary of 42 386 subwords for Estonian and 46 930 subwords for Finnish, resulting in 64 396 subwords when combined.

For segmentation of the English source, a separate Morfessor Baseline model was trained. To ensure consistency between source and target segmentations, we used the segmentation of the Cognate Morfessor model for any English words that were also present in the target side corpora. The source vocabulary consisted of 61 644 subwords.

As a baseline segmentation, we train a shared 100k subword vocabulary using BPE. To produce a balanced multilingual segmentation, the following procedure was used: First, word counts were calculated individually for English and each of the target languages Finnish and Estonian. The counts were normalized to equalize the sum of the counts for each language. This avoided imbalance in the amount of data skewing the segmentation in favor of some language. BPE was trained on the balanced counts. Segmentation boundaries around hyphens were forced, overriding the BPE.

[3] 1–100 tokens, 3–600 chars, $\leq$ 50 chars/token.
[4] Requiring ratio 0.5–2.0, if either side $>$ 10 chars.
[5] $<$ 10 occurrences
[6] 96% of the data was retained.

ϵ	→	n	27919	g	→	k	3000	il	→	ϵ	2077
ϵ	→	a	17082	ü	→	y	2979	m	→	mm	2016
ϵ	→	i	15725	oo	→	o	2790	s	→	n	2005
d	→	t	12599	t	→	a	2674	ee	→	e	1950
l	→	ll	5236	ϵ	→	k	2583	i	→	ϵ	1889
ϵ	→	ä	4437	aa	→	a	2536	ϵ	→	e	1803
s	→	ssa	3907	õ	→	o	2493	u	→	o	1724
t	→	tt	3863	a	→	ä	2479	ϵ	→	d	1496
o	→	u	3768	s	→	ϵ	2173	il	→	t	1486
e	→	i	3182	t	→	ϵ	2158	d	→	ϵ	1433

Table 2: 30 most frequent edits learned by the model. The direction is Estonian→Finnish. The numbers indicate how many times the edit was applied in the morph lexicon. ϵ indicates the empty string.

EN-ET	chrF-1.0 dev	BLEU% dev
BPE	56.52	17.93
monolingual	53.44	15.82
Cognate Morfessor	57.05	18.40
+finetuned	57.23	18.45
+ensemble-of-5	**57.75**	**19.09**
+ensemble-of-3	57.64	18.96
+linked embeddings	56.20	17.48
−LM filtering	52.94	14.65
6+6 layers	57.35	18.84

Table 3: Development set results for English–Estonian. character-F and BLEU scores in percentages. +/− stands for adding/removing a component. Multiple modifications are indicated by increasing the indentation.

Multilingual translation with target-language tag was done following (Johnson et al., 2016). A pseudo-word, e.g. <TO_ET> to mark Estonian as the target language, was prefixed to each paired English source sentence.

5 NMT system

We use the OpenNMT-py (Klein et al., 2017) implementation of the Transformer.

5.1 Transformer

The Transformer architecture (Vaswani et al., 2017) relies fully on attention mechanisms, without need for recurrence or convolution. A Transformer is a deep stack of layers, consisting of two types of sub-layer: multi-head (MH) attention (Att) sub-layers and feed-forward (FF) sub-layers:

$$\begin{aligned} \mathrm{Att}(Q,K,V) &= \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \\ a_i &= \mathrm{Att}(QW_i^Q, KW_i^K, VW_i^V) \\ \mathrm{MH}(Q,K,V) &= [a_1;\ldots;a_h]W^O \\ \mathrm{FF}(x) &= \max(0, xW_1+b_1)W_2+b_2 \end{aligned} \quad (3)$$

where Q is the input query, K is the key, and V the attended values. Each sub-layer is individually wrapped in a residual connection and layer normalization.

When used in translation, Transformer layers are stacked into an encoder-decoder structure. In the encoder, the layer consists of a self-attention sub-layer followed by a FF sub-layer. In self-attention, the output of the previous layer is used as queries, keys and values $Q = K = V$. In the decoder, a third context attention sub-layer is inserted between the self-attention and the FF. In context attention, Q is again the output of the previous layer, but $K = V$ is the output of the encoder stack. The decoder self-attention is also masked to prevent access to future information. Sinusoidal position encoding makes word order information available.

5.2 Training

Based on some preliminary results, we decided to reduce the number of layers to 4 in both encoder and decoder; later we found that the decision was based on too short training time. Other parameters were chosen following the OpenNMT FAQ (Rush, 2018): 512-dimensional word embeddings and hidden states, dropout 0.1, batch size 4096 tokens, label smoothing 0.1, Adam with initial learning rate 2 and β_2 0.998.

Fine-tuning for each target language was performed by continuing training of a multilingual model. Only the appropriate monolingual subset of the training data was used in this phase. The data was still prefixed for target language as during multilingual training. No vocabulary pruning was performed.

In our ensemble decoding procedure, the predictions of 3–8 models are combined by averaging after the softmax layer. Best results are achieved when the models have been independently trained. However, we also try combinations where a second copy of a model is further trained with a different configuration (monolingual finetuning).

	chrF-1.0				BLEU%			
EN-FI	nt2015	nt2016	nt2017	nt2017AB	nt2015	nt2016	nt2017	nt2017AB
BPE	58.59	59.76	62.00	63.06	21.09	21.04	23.49	26.55
monolingual	57.94	59.11	61.33	62.41	20.87	20.70	23.11	26.12
Cognate Morfessor	58.18	59.81	62.15	63.24	20.73	21.18	23.37	26.26
+finetuned	58.48	59.89	62.17	63.28	21.08	21.41	23.45	26.52
+ensemble-of-8	**59.07**	**60.69**	**62.94**	**64.07**	**21.50**	**22.34**	**24.59**	**27.55**
−LM filtering	58.19	59.39	61.78	62.82	20.62	20.77	23.38	26.36
+linked embeddings	57.79	59.45	61.52	62.58	19.95	20.84	22.70	25.69
6+6 layers	58.68	60.26	62.37	63.52	21.05	21.81	23.93	27.08

Table 4: Results for English–Finnish. character-F and BLEU scores in percentages. +/− stands for adding/removing a component. Newstest is abbreviated nt. Both references are used in nt2017AB.

We experimented with partially linking the embeddings of cognate morphs. In this experiment, we used morph embeddings concatenated from two parts: a part consisting of normal embedding of the morph, and a part that was shared between both halves of the cognate morph pair. Non-cognate morphs used an unlinked embedding also for the second part. After concatenation, the linked embeddings have the same size as the baseline embeddings.

We evaluate the systems with cased BLEU using the mteval-v13a.pl script, and characterF (Popovic, 2015) with β set to 1.0. The latter was used for tuning.

6 Results

Based on preliminary experiments, the Morfessor corpus cost weight α was set to 0.01, and the edit cost weight was set to 10. The most frequent edits are shown in Table 2.

Table 3 shows the development set results for Estonian. Table 4 shows results for previous year's test sets for Finnish.

The tables show our main system and the two baselines: a multilingual model using joint BPE segmentation, and a monolingual model using Morfessor Baseline.

Cognate Morfessor outperforms the comparable BPE system according to both measures for Estonian, and according to chrF-1.0 for Finnish. For Finnish, results measured with BLEU vary between test sets. The cross-lingual segmentation is particularly beneficial for Estonian.

In the monolingual experiment, the cross-lingual segmentations are replaced with monolingual Morfessor Baseline segmentation, and only the data sets of one language pair at a time is used. These results show that even the higher resourced language, Finnish, benefits from multilingual training.

The indented rows show variant configurations of our main system. Monolingual finetuning consistently improves results for both languages. For Estonian, we have two ensemble configurations: one combining 3 monolingually finetuned independent runs, and one combining 5 monolingually finetuned savepoints from 4 independent runs. Selection of savepoints for the ensemble was based on development set chrF-1. In the ensemble-of-5, one training run contributed two models: starting finetuning from epochs 14 and 21 of the multi-lingual training. The submitted system is the ensemble-of-3, as the ensemble-of-5 finished training after the deadline. For Finnish, we use an ensemble of 4 finetuned and 4 non-finetuned savepoints from 4 independent runs.

To see if further cross-lingual learning could be achieved, we performed an unsuccessful experiment with linked embeddings. It appears that explicit linking does not improve the morph representations over what the translation model is already capable of learning.

After the deadline, we trained a single model with 6 layers in both the encoder and decoder. This configuration consistently improves results compared to the submitted system.

All the variant configurations (ensemble, finetuning, LM filtering, linked embeddings, number of layers) used with Cognate Morfessor are compatible with each other. We did not not explore the combinations in this work, except for combining finetuning with ensembleing: all of the models in the Estonian ensembles, and 4 of the models in the Finnish

ensemble are finetuned. All the variant configurations except for linked embeddings could also be used with BPE.

7 Conclusions and future work

The translation system trained using the Cognate Morfessor segmentation outperforms the baselines for both languages. The benefit is larger for Estonian, the language with less data in this experiment.

One downside is that, due to the model structure, Cognate Morfessor is currently not applicable to more than two target languages.

Cognate Morfessor itself learns to model the frequent edits between cognate pairs. However, in the preprocessing cognate extraction step of this work, we used unweighted Levenshtein distance, which does not distinguish edits by frequency. In future work, weighted or graphonological Levenshtein distance could be applied (Babych, 2016).

Acknowledgments

This research has been supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 780069. Computer resources within the Aalto University School of Science "Science-IT" project were used. We wish to thank Peter Smit for groundlaying work that led to Cognate Morfessor.

References

Bogdan Babych. 2016. Graphonological Levenshtein edit distance: Application for automated cognate identification. *Baltic Journal of Modern Computing*, 4(2):115–128.

Michael Bloodgood and Benjamin Strauss. 2017. Using global constraints and reranking to improve cognates detection. In *Proc. ACL*, volume 1, pages 1983–1992.

Alina Maria Ciobanu and Liviu P Dinu. 2014. Automatic detection of cognates using orthographic alignment. In *Proc. ACL*, volume 2, pages 99–105.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proc. SIGPHON*, pages 21–30, Philadelphia, PA, USA. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proc. NAACL*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Oskar Kohonen, Sami Virpioja, and Mikaela Klami. 2009. Allomorfessor: Towards unsupervised morpheme analysis. In *Evaluating Systems for Multilingual and Multimodal Information Access*, volume 5706 of *Lecture Notes in Computer Science*, pages 975–982. Springer Berlin / Heidelberg.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proc. SIGMORPHON*, pages 78–86, Uppsala, Sweden. Association for Computational Linguistics.

Grzegorz Kondrak. 2001. Identifying cognates by phonetic and semantic similarity. In *Proc. NAACL*, pages 1–8. Association for Computational Linguistics.

Grzegorz Kondrak. 2009. Identification of cognates and recurrent sound correspondences in word lists. *TAL*, 50(2):201–235.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Ruslan Mitkov, Viktor Pekar, Dimitar Blagoev, and Andrea Mulloni. 2007. Methods for extracting and classifying pairs of cognates and false friends. *Machine translation*, 21(1):29.

Jason Naradowsky and Kristina Toutanova. 2011. Unsupervised bilingual morpheme segmentation and alignment with context-rich hidden semi-Markov models. In *Proc. ACL: HLT*, pages 895–904. Association for Computational Linguistics.

Maja Popovic. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *WMT15*, pages 392–395.

Taraka Rama. 2016. Siamese convolutional networks for cognate identification. In *Proc. COLING*, pages 1018–1027.

Alexander M. Rush. 2018. OpenNMT FAQ – How do i use the Transformer model? `http://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model`. Accessed: 27.7.2018.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *Proc. ACL16*.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proc. ACL: HLT*, pages 737–745.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. NIPS*, pages 6000–6010.

Sami Virpioja, Oskar Kohonen, and Krista Lagus. 2010. Unsupervised morpheme analysis with Allomorfessor. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, volume 6241 of *Lecture Notes in Computer Science*, pages 609–616. Springer Berlin / Heidelberg.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline. Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, Department of Signal Processing and Acoustics, Aalto University.

Richard Wicentowski. 2004. Multilingual noise-robust supervised morphological analysis using the WordFrame model. In *Proc. SIGPHON*, pages 70–77. Association for Computational Linguistics.