LIMSI@IWSLT'16: MT Track

Franck Burlot, Matthieu Labeau, Elena Knyazeva,
Thomas Lavergne, Alexandre Allauzen, François Yvon

LIMSI, CNRS, Univ. Paris-Sud, Université Paris Saclay, 91 403 Orsay, France
firstname.lastname@limsi.fr

Abstract

This paper describes LIMSI's submission to the MT track of IWSLT 2016. We report results for translation from English into Czech. Our submission is an attempt to address the difficulties of translating into a morphologically rich language by paying special attention to the morphology generation on target side. To this end, we propose two ways of improving the morphological fluency of the output: 1. by performing translation and inflection of the target language in two separate steps, and 2. by using a neural language model with characted-based word representation. We finally present the combination of both methods used for our primary system submission.

1. Introduction

This paper documents LIMSI's participation to the MT Track of IWSLT 2016 for English-to-Czech. The reported experiments are an attempt to address the difficulties of translating into a morphologically rich language. In Statistical Machine Translation (SMT), the generated target language contains several incorrect word forms that show errors in agreement within a noun phrase, that encode the wrong grammatical function of the word in the sentence, or that simply convey the wrong meaning from the source. Our attempt to tackle this issue consists in two methods that improve the morphological fluency of the output.

The first method implies a specific representation of the target morphologically rich language. Words are normalized, i.e. morphological information that is redundant with respect to English is removed, such as case for nouns, that has no equivalent in English. The MT system then translates from English into normalized Czech. In a second step, a classifier is used to predict the previously removed information and helps to generate a word form.

The second method is a neural language model that is used to re-score the n-best hypothesis from the SMT system. This model builds word representations from the character level, which makes it able to take into account morphology.

In the first section, we introduce our SMT baseline setup and the data pre-processing we applied. We then describe successively the two methods above to handle target morphology. Finally, we present the results obtained by combining them.

2. Baseline System Overview

Our experiments mainly use NCODE,[1] an open source implementation of the n-gram approach, as well as MOSES[2] for some contrastive experiments. For more details about these toolkits, the reader can refer to [1] for MOSES and to [2] for NCODE.

2.1. Data Selection

For this task, we used the data provided at IWSLT 2016, as well as the permissible data provided at WMT 2016.[3] The parallel data used to train the translation model consists in the TED corpus, the QED corpus[4] and Europarl [3], which sum up to 885k parallel sentences. The monolingual data consists in the target side of the parallel data, the Czech news corpora provided at WMT 2016 and a subset of the Czeng1.6-pre corpus [4] labelled as "subtitles", which brings us to nearly 90M monolingual sentences.

The baseline system is optimized on a concatenation of English-to-Czech TED test sets 2010 and 2011, and tested over the official test sets from IWSLT 2016 (TED-2015, TED-2016 and QED-2016). We held out the concatenation of English-to-Czech TED test sets 2012 and 2014 as a development set to re-rank n-best hypothesis from the MT system (see Sections 3.3 and 4).

2.2. Data Pre-processing and System Setup

All the English data has been cleaned by normalizing character encoding. Its tokenization and truecasing relies on in-house text processing tools [5]. The Czech data is tokenized and truecased using scripts from the Moses toolkit.

Symmetrized word alignments are trained using `fast_align`. Our NCODE and MOSES systems are optimized with `mira`. 4-gram language models are trained with removed singletons using KenLM [6].

2.3. NCODE

NCODE implements the bilingual n-gram approach to SMT [7, 8, 9] that is closely related to the standard phrase-based

[1] http://ncode.limsi.fr
[2] http://www.statmt.org/moses/
[3] http://www.statmt.org/wmt16
[4] http://alt.qcri.org/resources/qedcorpus

approach [10]. In this framework, the translation is divided into two steps. To translate a source sentence **f** into a target sentence **e**, the source sentence is first reordered according to a set of rewriting rules so as to reproduce the target word order. This generates a word lattice containing the most promising source permutations, which is then translated. Since the translation step is monotonic, the peculiarity of this approach is to rely on the n-gram assumption to decompose the joint probability of a sentence pair in a sequence of bilingual units called tuples.

$$e^* = \arg\max_{\mathbf{e},\mathbf{a}} \sum_{k=1}^{K} \lambda_k f_k(\mathbf{f}, \mathbf{e}, \mathbf{a})$$

where K feature functions (f_k) are weighted by a set of coefficients (λ_k) and **a** denotes the set of hidden variables corresponding to the reordering and segmentation of the source sentence. Along with the n-gram translation models and target n-gram language models, 13 conventional features are combined: 4 *lexicon models* similar to the ones used in standard phrase-based systems; 6 *lexicalized reordering models* [11, 2] aimed at predicting the orientation of the next translation unit; a "weak" distance-based *distortion model*; and finally a *word-bonus model* and a *tuple-bonus model* which compensate for the system preference for short translations. Features are estimated during the training phase. Training source sentences are first reordered so as to match the target word order by unfolding the word alignments [12]. Tuples are then extracted in such a way that a unique segmentation of the bilingual corpus is achieved [9] and n-gram translation models are then estimated over the training corpus composed of tuple sequences made of surface forms or PoS tags. Reordering rules are automatically learned during the unfolding procedure and are built using part-of-speech (PoS), rather than surface word forms, to increase their generalization power [12].

3. Two-step Machine Translation

The first method we propose to handle target-side rich morphology is a two-step MT procedure where translation and morphology generation are processed apart. The first step of the proposed scenario consists in translating from English into normalized Czech. For this purpose, the target side of the parallel data and the monolingual data have to be pre-processed.

3.1. Normalization of the Czech Data

Goldwater and McClosky [13] and Durgar El-Kahlout and Yvon [14] show the benefits of normalizing the morphologically rich language (respectively Czech and German) on the source side when translating into English. Such a normalization consists in grouping different word forms sharing the same lemma into a common class, by removing one or many attributes (e.g. gender, number, case) that are considered as redundant with respect to English. This pre-processing has the effect of reducing the source vocabulary, making both languages more symmetric, and has a positive impact on the translation quality.

When translating in the reverse direction, these ideas hold, but one needs in addition to make sure that the attribute that was removed at normalization step is recoverable from the monolingual context in the SMT output. Indeed, the models we propose for re-inflexion do not have access to source side information (see Section 3.2). Therefore, whenever an attribute is redundant with respect to English but is needed for the prediction of other attributes in surrounding words, it needs to be kept.

In our pre-processing, a word is represented as a lemma and a tag sequence, which we obtained using Morphodita [15]. Normalizing such a word simply means removing one or many tags from the sequence. We propose a deterministic schema for each part of speech. The following attributes are preserved:

- **Nouns**: *lemma, PoS, gender and number*. Number is an attribute that is common to English, and gender is an intrinsic part of Czech nouns, meaning that it may serve to disambiguate two identical lemmas that have a different lexical meaning. Moreover, as head of a noun phrase, the word propagates gender to its dependents. Case is systematically removed and we consider that it should be predictable from the monolingual context[5].
- **Adjectives**: *lemma, PoS, negation, degree of comparison*. Since the adjective is invariable in English, we remove gender and number, but keep both negation, which has a lexical value, and the degree of comparison, which is also marked in English.
- **Numerals**: *lemma, PoS*. Numbers have only one form in English.
- **Pronouns**: *lemma, PoS, subPoS, person, gender, number, number[psor], gender[psor]*. Only case is removed from pronouns. Gender and number of both possessor (*[psor]*) and possessed are hard to predict and are generally not well handled in SMT. We leave these attributes and are aware that their prediction would require a special attention that is beyond the scope of these experiments. Person is also kept and we expect it to be a useful predictor of nominative case when a pronoun agrees with a verb in the context.
- **Prepositions**: *word form, PoS, case*. Here, we keep the word form, since some prepositions have different forms depending on the right side context, e.g. ***s***

[5] Some contexts make case prediction hard and this attribute should probably sometimes be conveyed from the source, as in the normalized output *jím ruka+Plur* (*eat hand+Plur*). Instrumental case needs to be predicted for the noun, in order to obtain *jím rukama* (I eat with my hands). If the case tag is lost in this output, the classifier used for re-inflexion may ignore the semantic aspect of the clause and consider the noun as a direct object, generating the semantically less likely sentence with accusative case *jím ruce* (I eat hands).

tebou (with you) - ***se*** *mnou* (with me). The SMT system handles well this phenomenon. Case is kept, since some prepositions can be followed by different cases and we expect this attribute to propagate through the entire preposition phrase in the output. This choice implies that verb constructions are expected to be handled by the SMT system that is considered to be able to distinguish *jít v + Accusative* (go to) and *být v + Locative* (be in).

- **Verb**: The lemma and the whole tag sequence are kept. Verbs are not normalized, and we follow the same principle as Fraser et al. [16] that this PoS be considered an anchorage point of the output. The full tag sequence should mainly help distinguish the object from the subject with which it should agree in person, gender, and number.

- **Adverb, interjection, conjunction, particle**: Word forms are kept, since they have no morphological variation.

In this setup, only three attributes can be removed: gender, number and case. This constraint makes the tag prediction task easier, since only sequences of three tags need to be predicted (as opposed to sequences of fifteen tags according to the Morphodita tagset[6]). Finally, it allows us to train one different classifier for each attribute, as described in the next section.

3.2. Output Re-inflection

The machine translation system outputs a text in a normalized language that needs to be re-inflected. At this step, we have lemmas associated with a fixed sequence of attributes, some of them having missing values (gender, number and/or case). The task is therefore similar to any sequence labeling problem where the goal is to predict the right value for each empty attribute. When the full tag sequence has been predicted, a dictionary is used to recover the word form corresponding to the predictions.

The model we considered is a conditional random field [17] that predicts three morphological attributes using the Wapiti toolkit [18]. A joint prediction of all these attributes allows us to better account for the dependencies between them, but such a model can be challenging to train due to the potentially high number of attribute combinations to consider.

A total of 180 different combinations of attributes are observed in our corpus, which are reachable for a CRF model but would require more training data than available to obtain reasonable performance. To overcome this problem, we train a cascade of CRF models, in which the first three models predict a single morphological attribute. That output is used to feed the final joint classifier. The final joint model is therefore only responsible for discovering the dependencies between the attributes and for correcting the predictions made by the previous models.

All four models are trained using 1- to 3-gram word features in an 11-word window as well as 1- to 4-gram features concerning the known morphological information in the same window. Additionally, 1- to 4-gram features on the output of each previous models are used. The models are trained in a specific order: gender, number and case are successively trained, then the joint model is learnt. The same order is followed for decoding.

To extract the features based on previous models, a full decoding of the training data by these models is necessary. To get unbiased predictions, a 10-fold cross-validation is done for the training of the first three models.

The three morphological attributes should be predicted only in words for which they have been removed during the normalization process. Gender, for example, has to be predicted for adjectives but not for nouns. The models are trained to predict the attributes for every relevant words even if they are already known, but during inference the Viterbi decoder is forced to only consider paths that go through the already known attributes. This forced decoding allows the model to take account of this knowledge to make its predictions.

In order to train this CRF model, we used data from the Universal Dependencies Treebank project[7]. We used the Czech and Czech-CAC corpora covering general domain and transcripts of spoken language for a total of $2G$ words, from which $170k$ where held out as a development set.

3.3. Experimental Results

Experimental results for this two-step MT setup are shown in Table 1. Re-inflecting the one-best hypothesis from the MT system does not improve the baseline system on TED-2015 set, slightly improves it on TED-2016 and significantly deteriorates it on QED-2016. These mixed results become closer to each-other and provide a significant improvement in the nk-best re-inflection setup (except for the QED test set that shows no improvement with NCODE MT system). The latter results where obtained by taking the n-best hypothesis from the MT system ($n = 300$) and by keeping the k-best predictions of the CRF ($k = 5$). These nk-best hypothesis were then re-scored using `mira` over the official development data provided at the Workshop as test-2012 and test-2013. This optimization procedure considers the scores given by the MT system, as well as two additional scores: the score of the hypothesis given by the same language model used in the en2cs system[8] and the score returned by the CRF.

We can observe that the QED corpus gives the highest deterioration while re-inflecting one-best hypothesis and the

[6] `https://ufal.mff.cuni.cz/pdt2.0`

[7] `http://universaldependencies.org`

[8] Thus we end up with two distinct scores from two language models trained over the same data: the first one over normalized Czech (in en2cx system) and the second one over fully inflected Czech (in en2cx2cs system).

Table 1: BLEU scores for Moses and Ncode systems over direct translations (en2cs) and two-step translations (en2cx2cs) over the official IWSLT 2016 test sets.

Setup	TED-2015		TED-2016		QED-2016	
	MOSES	**NCODE**	**MOSES**	**NCODE**	**MOSES**	**NCODE**
en2cs	18.24	18.37	15.38	15.27	16.30	16.20
CRF	18.09 (-0.15)	18.35 (-0.02)	15.85 (+0.47)	15.86 (+0.59)	15.97 (-0.33)	15.83 (-0.37)
+ nk-best	18.84 (+0.60)	**19.65 (+1.28)**	16.32 (+0.94)	**16.63 (+1.36)**	**16.70 (+0.40)**	16.25 (+0.05)

worse improvement with nk-best re-inflection. We understand these scores as a result of the fact that the sentences from this test set are segmented, which drastically narrows the context that the CRF can use to make the right prediction.

We finally notice that, for the TED test sets, NCODE systems seem to make better use of the normalization of Czech data than MOSES systems in the nk-best re-inflection setup. Indeed, NCODE re-inflection outperforms MOSES by 0.81 (TED-2015) and 0.31 (TED-2016) in terms of BLEU. On the opposite, MOSES is 0.34 BLEU points higher than NCODE in the same setup over the QED test set.

4. Character-Based Neural Language Model

To address the difficulties of translating into a morphologically rich language, we choose to use an open-vocabulary character-level neural language model to re-score the n-best hypothesis of the MT system. We use a convolution layer followed by pooling to compute word representations from character n-grams. On top of this layer, we use a feedforward n-gram neural language model [19].

4.1. Character-level Word Embeddings

In word-based neural language models, word embeddings are parameters stored in a Look-up matrix $\mathbf{L}$. The embedding $\mathbf{e}_w^{word}$ of a word w is simply the column of $\mathbf{L}$ corresponding to its index in the vocabulary:

$$\mathbf{e}_w^{word} = [\mathbf{L}]_w$$

To infer a word embedding from its character embeddings, we use a *convolution layer* [20, 21]. As illustrated in figure 1, a word w is a character sequence $\{c_1, .., c_{|w|}\}$ represented by its embeddings $\{\mathbf{C}_{c_1}, .., \mathbf{C}_{c_{|w|}}\}$, where $\mathbf{C}_{c_i}$ denotes the vector associated to the character c_i. A convolution filter $\mathbf{W}^{conv} \in \mathbb{R}^{d_e} \times \mathbb{R}^{d_c * n_c}$ is applied over a sliding window of n_c characters, producing local features :

$$x_n = \mathbf{W}^{conv}(\mathbf{C}_{c_{n-n_c+1}} : .. : \mathbf{C}_{c_n})^T + \mathbf{b}^{conv}$$

where x_n is a vector of size d_e obtained for each position n in the word.[9] The i-th element of the embedding of w is the mean over the i-th elements of the feature vectors, passed by the activation function ϕ :

$$[\mathbf{e}^{char}]_i = \phi\left(\sum_{n=1}^{|w|-n_c+1} \frac{[\mathbf{x}_n]_i}{|w| - n_c + 1}\right) \quad (1)$$

Using a mean after a sliding convolution window ensures that the embedding combines local features from the whole word, and that the gradient is redistributed at scale for each character n-gram. The parameters of the layer are the matrices $\mathbf{C}$ and $\mathbf{W}^{conv}$ and the bias $\mathbf{b}^{conv}$.

4.2. Models

Our model follows the classic n-gram feedforward architecture [19]. The input of the network is a n-words context $H_i = (w_{i-1}, \ldots, w_{N-i+1})$, and its output the probability $P(w|H_i)$ for each word $w \in \mathcal{V}$. The embeddings of the word in the context are concatenated and fed into a hidden layer:

$$\mathbf{h}^{H_i} = \phi(\mathbf{W}^{hidden}(\mathbf{e}_{i-1} : \ldots : \mathbf{e}_{N-i+1}) + \mathbf{b}^{hidden})$$

A second hidden layer my be added. Finally, the output layer computes scores for each word:

$$\mathbf{s}^{H_i} = \exp\left(\mathbf{W}^{out}\mathbf{h}^{H_i} + \mathbf{b}^{out}\right)$$

$\mathbf{W}^{hidden}$, $\mathbf{b}^{hidden}$, $\mathbf{W}^{out}$ and $\mathbf{b}^{out}$ are the parameters of the model. As the input Lookup-matrix $\mathbf{L}$, the output weight matrix $\mathbf{W}^{out}$ contains word embeddings, that are output representations of the words in the vocabulary:

$$\mathbf{e}_w^{out} = [\mathbf{W}^{out}]_w$$

Then, the output scores are expressed as:

$$s^{H_i}(w) = \exp\left(\mathbf{e}^{out}\mathbf{h}^{H_i} + \mathbf{b}^{out}\right) \quad (2)$$

Later, we will use two different input layers to obtain word representations:

- A classic NLM using word-level embeddings only, that we will note **WE**.
- A NLM using embeddings constructed from character n-grams by convolution + pooling, concatenated with word embeddings, that we will note **CWE**.

[9] Two padding character tokens are used to deal with border effects. The first is added at the beginning and the second at the end of the word, as many times as it is necessary to obtain the same number of windows than the length of the word. Their embeddings are added to $\mathbf{C}$.

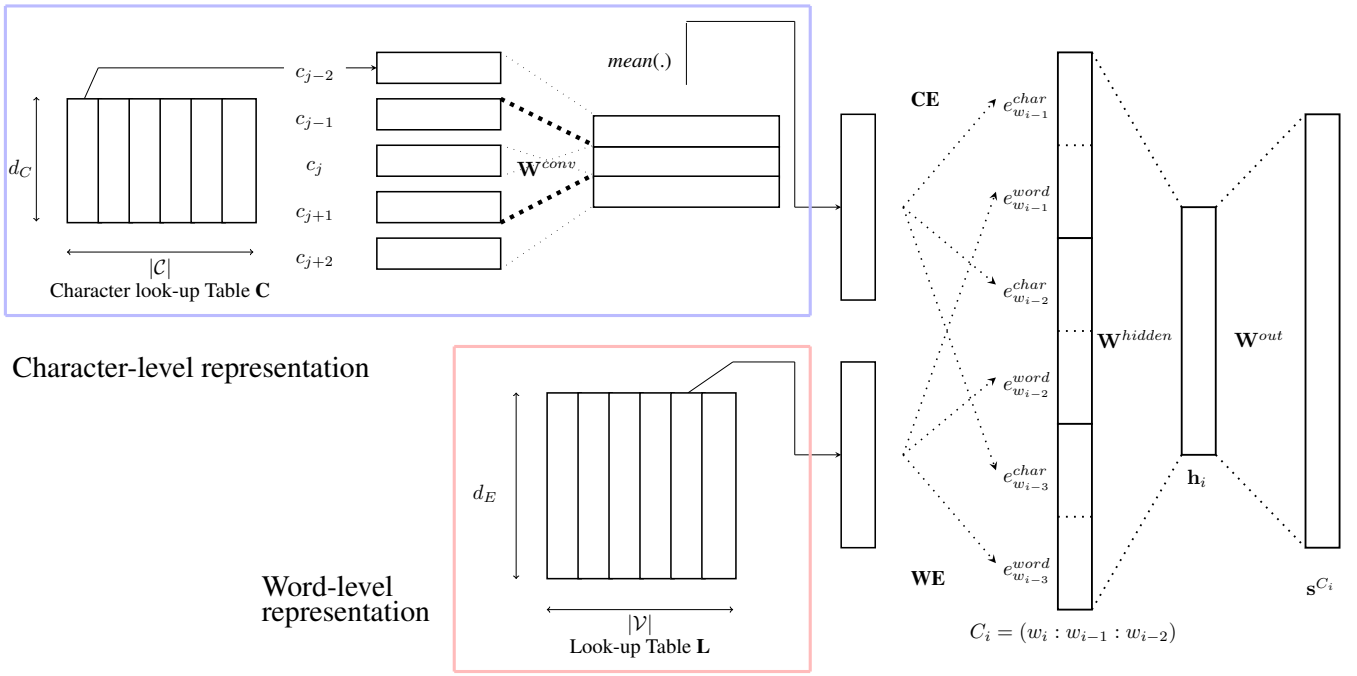

Figure 1: **CWE** Model architecture

4.3. Objective Function for Open Vocabulary Models

Usually, such a model is trained by maximizing the log-likelihood. This objective function raises two important issues. For conventional word models, it implies a very costly summation imposed by the softmax activation of the output layer. More importantly, this objective requires the definition of a finite vocabulary, while the proposed model may use character-based word embeddings, especially in the output, making the notion of vocabulary obsolete.

Therefore, the parameter estimation relies on Noise-Contrastive Estimation (NCE) introduced in [22, 23].

4.4. Character-based Output Weights with Noise-Contrastive Estimation

The output weights representing each word in the vocabulary $\mathbf{e}^{out}$ can also be replaced by embeddings computed by a convolution layer on character n-grams. In this case the model can efficiently represent and infer a score for any word observed during the training process or not, while with conventional word embeddings, out-of-vocabulary words only share the same representation and distribution. Instead of using a parameter matrix $\mathbf{W}^{out}$ to estimate the score like in equation 2, $\mathbf{e}^{out}$ can be replaced by $\mathbf{e}^{char-out}$ vector estimated on the fly based on its character sequence as described in equation 1. With this extension, the model does not rely on a vocabulary anymore, hence motivating our choice of the NCE : this criterion allows us to train both types of models based on conventional word embeddings, along with character-based embeddings. This unnormalized objective allows us to handle an open vocabulary, since we only need to compute $k+1$ word representations for each training examples. Models that use character-based embeddings both for input and output words are denoted by **CWE-CWE**.

4.5. Training

WE, **CWE**, and **CWE-CWE** models were trained using Adagrad [24] and using batches of 128 for various context sizes. The ReLu activation function is used, along with an embedding size of $d_e = 128$. When relevant, we used a character embedding size of $d_c = 32$ and a convolution on $n_c = 5$-grams of characters for all experiments[10]. We sampled 25 examples from the noise distribution for each example. The models were implemented using C++. Our Neural language models are trained on the target-side of the parallel data and the monolingual data used for the MT system, but training examples are sampled from corpora given weights that are computed to balance in-domain parallel data (**TED**), out-of domain parallel data, and additional monolingual data.

4.6. Re-scoring of en2cs Outputs

Experimental results are shown in Table 2. The procedure is similar to what is described in section 3.3: the 300-best hypothesis from the MT system are scored by our language models, and re-ranked using `mira` over the official development data provided at the Workshop as test-2012 and test-2013. We re-scored only outputs from NCODE systems, with **WE**, **CWE** and **CWE-CWE** systems. Models used here are

[10]Results do not improve significantly when increasing these embedding sizes, while a negative impact is observed on convergence speed and computation time.

Table 2: BLEU scores for re-ranked n-best direct translations (en2cs) Ncode outputs over the official IWSLT 2016 test sets.

Setup	TED-2015	TED-2016	QED-2016
en2cs baseline	18.37	15.27	16.20
WE	19.64 (+1.27)	16.40 (+1.13)	**17.54 (+1.34)**
CWE	**19.67 (+1.30)**	**16.48 (+1.21)**	17.05 (+0.85)
CWE-CWE	19.22 (+0.85)	15.83 (+0.56)	16.21 (+0.01)

Table 3: BLEU scores for re-ranked re-inflected nk-best translation hypothesis (en2cx2cs) over the official IWSLT 2016 test sets.

Setup	TED-2015	TED-2016	QED-2016
en2cs baseline	18.37	15.27	16.20
CRF	19.65 (+1.28)	16.63 (+1.36)	16.25 (+0.05)
WE	19.65 (+1.30)	16.66 (+1.39)	**16.26 (+0.06)**
CWE	**19.77 (+1.42)**	**16.80 (+1.53)**	15.96 (-0.24)
CWE-CWE	19.25 (+0.88)	16.31 (+1.04)	15.27 (-0.93)

trained with a context size of $n = 6$ words.

5. Re-inflection and Re-scoring

Our primary submission for the TED test sets consists in a combination of both methods that handle target-side morphology: the re-inflection procedure (introduced in Section 3.2) and the re-scoring of nk-best hypothesis from NCODE system (as shown in Section 3.3) using the neural language model with a character-based word representation (introduced in Section 4).

The results obtained in this manner are shown in Table 3. They bring out the fact that re-ranking nk-best hypothesis with the **CWE** model gives a slight improvement over the use of an n-gram word-based language model.

As for the QED set, the character-based word representations are not able to give any improvement over the baseline in the re-inflection setup. Therefore, our primary submission for this set is a direct English-to-Czech translation with an n-best re-ranking using a word-based neural language model (**WE** in Table 2). This setup gives the best improvement we could achieve over the baseline.

6. Conclusions

This paper describes LIMSI's system submission for IWSLT 2016. We report results on English-to-Czech systems as an attempt to address the difficulties of translating into a morphologically rich language.

We have introduced a representation of Czech words that does not take into account morphological information that is redundant with respect to English, such as case for nouns. This morphology normalization is expected to improve the translation step. In the next step, we showed that a CRF model could be used to transform a normalized Czech word into an inflected form. However, this re-inflection step does not give any improvement over the baseline when it is performed on the one-best hypothesis of the MT system. Running re-inflection over the n-best hypothesis and keeping the k-best hypothesis from the CRF model improves translation in terms of BLEU score when we proceed to a re-ranking using a language model. We have shown results with a classic n-gram language model, as well as an open vocabulary neural language model building word representations from characters.

The n-best hypothesis re-ranking using a neural language model was introduced as an alternative to the two-step MT setup, since the characted-based representation it uses is able to model rich morphology. Both neural language model n-best re-ranking and nk-best re-inflection turned out to bring a comparable improvement over the baseline. Finally, their combination only gives a slight improvement over the results obtained with each model separately, which tends to show that both models address the same issue: target-side morphological correctness.

7. Acknowledgments

This work has been partly funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

8. References

[1] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. ACL:Systems Demos*, Prague, Czech Republic, 2007.

[2] J. M. Crego, F. Yvon, and J. B. Mariño, "N-code: an open-source Bilingual N-gram SMT Toolkit," *Prague*

Bulletin of Mathematical Linguistics, vol. 96, pp. 49–58, 2011.

[3] P. Koehn, "A parallel corpus for statistical machine translation," in *Proc. MT-Summit*, Phuket, Thailand, 2005.

[4] O. Bojar, O. Dušek, T. Kocmi, J. Libovický, M. Novák, M. Popel, R. Sudarikov, and D. Variš, "CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered," in *Text, Speech and Dialogue: 19th International Conference, TSD*. Springer Verlag, September 12-16 2016, in press.

[5] D. Déchelotte, G. Adda, A. Allauzen, O. Galibert, J.-L. Gauvain, H. Maynard, and F. Yvon, "LIMSI's statistical translation systems for WMT'08," in *Proceedings of NAACL-HTL Statistical Machine Translation Workshop*, Columbus, Ohio, 2008.

[6] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proc. WMT*, Edinburgh, Scotland, 2011, pp. 187–197.

[7] F. Casacuberta and E. Vidal, "Machine translation with inferred stochastic finite-state transducers," *Computational Linguistics*, vol. 30, no. 3, pp. 205–225, 2004.

[8] J. M. Crego and J. B. Mariño, "Improving statistical mt by coupling reordering and decoding," *Machine translation*, vol. 20, no. 3, pp. 199–215, Jul 2006.

[9] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-jussà, "N-gram-based machine translation," *Comput. Linguist.*, vol. 32, no. 4, pp. 527–549, Dec. 2006.

[10] R. Zens, F. J. Och, and H. Ney, "Phrase-Based Statistical Machine Translation," in *25th German Conf. on Artificial Intelligence (KI2002)*. Aachen, Germany: Springer Verlag, Sept. 2002, pp. 18–32.

[11] C. Tillmann, "A unigram orientation model for statistical machine translation," in *Proceedings of HLT-NAACL 2004: Short Papers*, ser. HLT-NAACL-Short '04, Stroudsburg, PA, USA, 2004, pp. 101–104.

[12] J. M. Crego and J. B. Mariño, "Improving statistical MT by coupling reordering and decoding," *Machine Translation*, vol. 20, 2006.

[13] S. Goldwater and D. McClosky, "Improving statistical MT through morphological analysis," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, ser. HLT '05, 2005, pp. 676–683.

[14] I. Durgar El-Kahlout and F. Yvon, "The pay-offs of preprocessing for German-English Statistical Machine Translation," in *Proceedings of the International Workshop on Spoken Langugage Translation*, M. Federico, I. Lane, M. Paul, and F. Yvon, Eds., 2010, pp. 251–258.

[15] J. Straková, M. Straka, and J. Hajič, "Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition," in *Proc. ACL: System Demos*, Baltimore, Maryland, 2014, pp. 13–18.

[16] A. Fraser, M. Weller, A. Cahill, and F. Cap, "Modeling inflection and word-formation in SMT," in *Proc. EACL*, Avignon, France, 2012, pp. 664–674.

[17] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 282–289.

[18] T. Lavergne, O. Cappé, and F. Yvon, "Practical very large scale CRFs," in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, July 2010, pp. 504–513.

[19] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, p. 1137 1155, 2003.

[20] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, *Readings in Speech Recognition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, ch. Phoneme Recognition Using Time-delay Neural Networks, pp. 393–404.

[21] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Nov. 2011.

[22] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 307–361, Feb. 2012.

[23] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models." in *ICML*. icml.cc / Omnipress, 2012.

[24] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-24, Mar 2010.