MultiWord Expression Aware Neural Machine Translation

Andrea Zaninello[*†], **Alexandra Birch**[*]
[*]School of Informatics, University of Edinburgh, United Kingdom
[†]Zanichelli editore, Bologna, Italy
azaninello@zanichelli.it, a.birch@ed.ac.uk

Abstract

Multiword Expressions (MWEs) are a frequently occurring phenomenon found in all natural languages that is of great importance to linguistic theory, natural language processing applications, and machine translation systems. Neural Machine Translation (NMT) architectures do not handle these expression well and previous studies have not explicitly addressed MWEs in this framework. In this work, we show that using external linguistic resources and data augmentation we can improve both translations of MWEs that occur in the source, and the generation of MWEs on the target, and improve performance by up to 5.09 BLEU points on MWE test sets. We also devise a MWE score to specifically assess the quality of MWE translation which agrees with human evaluation. We make available the MWE score implementation – along with MWE-annotated training sets and corpus-based lists of MWEs – for reproduction and extension.

Keywords: multiword expressions, neural machine translation, evaluation

1. Introduction

Multiword Expressions (MWEs) are a pervasive phenomenon in all natural languages to the point that, according to some studies, they represent approximately half of a language's lexicon (Jackendoff, 1995). They also challenge NLP applications because of their often unpredictable morpho-syntactic and lexico-semantic behaviour (Villavicencio et al., 2005). We call a MWE an expression that is composed of two or more words working as a unit with respect to some levels of linguistic analysis (Calzolari et al., 2002); a MWE displays idiosyncratic properties that cannot be explained solely on the basis of regular syntactic and semantic rules (Everaert et al., 2014) and is generally characterised by some degree of conventionality (Baldwin and Kim, 2010; Constant et al., 2017).

In the last few years, Neural Machine Translation (NMT) has proved the best performing framework compared to previous methodologies, with neural architectures producing ever more natural-sounding target language. Even so, NMT output is sometimes a poor translation of the source sentence (Nguyen and Chiang, 2018) and it is therefore important to investigate specific linguistic phenomena and improve translation quality not only in terms of standard measurements.

Previously dominant phrase-based and syntax-based Statistical Machine Translation (SMT) techniques (Koehn et al., 2007; Junczys-Dowmunt et al., 2016) naturally take into account phrasal components, and there has been significant research on MWEs in these frameworks; however, for NMT, due to a lack of phrasal segmentation, it is less obvious how to address specific language phenomena such as MWEs. Moreover, while standard metrics are effective in terms of system comparison, their ability to account for more fine-grained improvements in MT is less straightforward (Callison-Burch et al., 2006), and their effectiveness has been questioned. Therefore, the performance of NMT architectures in translating MWEs remains an open challenge.

The aim of this study is to empirically verify whether integrating information on MWEs either through targeted training examples or through explicit annotation in the target language can help disambiguating between simple phrasal units and non-compositional expressions, and thus be beneficial to NMT. In our first approach, we try augmenting our training data with entries from a bilingual and a monolingual MWE dictionary, adding a relatively small number of instances (10% and 2% of the original data, respectively), both in isolation and in their sentence context from usage examples provided. The second approach takes a MWE annotation tool, and labels MWE on the source. We either concatenate MWE into one word or we use factors to indicate if they form part of a MWE.

We show that for a test set comprised of genuinely non-compositional MWEs the NMT output is of extremely low quality, indicating that these models struggle to handle these examples, especially in the small training data condition. We also show that all our methods improve translation in general and MWE translation in particular. The method of including MWE in context, with backtranslation to recreate the source side, does well in the low resource setting, but given the small number of genuine examples is not scalable. Our approach of labelling MWE does however extend to improving translation in a large resource experiment.

In order to further analyse our results, we propose a novel evaluation metric (the *Score_mwe*) that specifically evaluates how well MWEs on the source side are translated. It needs a test set with human annotated MWE on the source and their translation in the reference. It uses the Levenstein distance to find the closest matching word in the hypothesis and rewards partial matches at the character level. We compare our novel metric with manual evaluation and show that it agrees with human judgments.

In this paper we limit our study to one language pair (from English to Italian) and to one specific neural architecture, but our methods can easily be extended to other language combinations or different NMT frameworks. We also rely on human curated resources in order to prove their value to NMT, and in future work we plan to consider automatically extracted MWE lexicons and unsupervised taggers.

2. Related work

While most work focussing on MWEs in MT has been within the context of rule-based and statistical machine translation systems, there have however been recent papers addressing MWEs in an NMT framework. Some use general knowledge about phrasal structures to improve translation, such as Tang et al. (2016), for example, who improve translation by symbolically encoding phrasal candidates on the source side and allowing the decoder to output more than one word at a time for source phrases. While they focus on fixed expressions such as named entities, names, places, numbers etc., in our experiment 6.3.1. we address virtually all MWE categories.

Few recent studies also propose methods to integrate explicit information on MWEs in the context of NMT. For example, Rikters and Bojar (2017) use parallel MWE candidates on the source side, automatically extracted from preprocessed text, and employ them as additional training data (method 1), and extract parallel sentences featuring the MWEs selected in (1) as additional training examples (method 2). Our first method (Section 6.2.1.) is inspired by their first, best performing one, but while they use standard toolkits for MWE extraction, an approach that is affected by the accuracy of the selected tool and by noise of the data, we aim to minimize false positives by using a manually compiled bilingual dictionary to select the MWE candidates on the source side.

Other studies exploit monolingual data to learn a better language model and integrate it into the decoder (Gulcehre et al., 2015). Monolingual data in the target language are paired with synthetic back-translations on the source side (Edunov et al., 2018), and this new bi-text is used as additional training text. This method is effective in improving translation quality, as described in Sennrich et al. (2016a), outperforming state-of-the-art results for English-Turkish. Our second method (6.2.2.) is inspired by these approaches, which we extend by applying them specifically to MWEs.

Most studies focussing on MWE translation evaluation aim to evaluate how well the MWEs identified in the source text are translated into the target language. This involves (i) identifying the MWEs in the source text, (ii) identifying their translation in the target text, (iii) evaluating the translation quality based on criteria such as adequacy (full, partial, etc.) and fluency (fluent, non–native, disfluent, etc.) (Ramisch et al., 2013). The first two tasks are usually completed by MWE extraction tools and automatic alignment, the third is usually carried out via instance-based manual inspection of the output translation. When not evaluated in terms of accuracy over a manually compiled gold standard (Monti et al., 2015), translation quality is evaluated through standard measures such as word–based BLEU (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2014) or character–based chrF (Popović, 2015), but none of them specifically addresses the translation of MWEs (although character–based metrics have similarities to our proposed *Score_mwe* measure, as we will discuss in Section 4.2.).

3. Background

3.1. Neural machine translation

The neural machine translation toolkit used in all our experiments, Nematus (Sennrich et al., 2017), implements a bi-directional encoder-decoder architecture with attention, similar to the model described by Bahdanau et al. (2015). The Encoder consists of two single recurrent neural networks (RNNs), one encoding the input forward from left to right, the other backward from right to left, so that all the context from the input is available at each time step (not only the preceding words), and the hidden state of a word is represented by the concatenation of these hidden states.
In the implementation we employ in all our experiments, training is performed by cross-entropy minimization on the parallel training corpus with Adam (Kingma and Ba, 2014), a variant of the stochastic gradient descent algorithm. Periodic validation is performed on smoothed sentence-level BLEU (Chen and Cherry, 2014) and early stopping on this metric is applied for training stabilization.

3.2. Integrating input features: factored NMT

In order to allow to specify for arbitrary linguistic input features for each word, in the methods we describe in Sections 6.3.2. and 6.4. we represent the encoder input as a concatenation of input features, as originally proposed by Sennrich and Haddow (2016). The idea is that, for each feature, a separate embedding vector is created, and all feature vectors are then concatenated to form a factored representation of the input word, whose length is equal to the total embedding size. This is done for an arbitrary number of input features $|F|$ according to the equation 1

$$\vec{h_j} = tanh(\vec{W}(\|_{k=1}^{|F|} E_{k X_j k}) + \vec{U}\vec{h}_{j-1}) \tag{1}$$

where $\|$ is vector concatenation, E_k is a feature embedding matrix, K_k is the vocabulary size of input feature k, $\vec{W}$ and $\vec{U}$ are weight matrices.

4. Evaluation methods

4.1. BLEU score

For general translation evaluation we use detokenized, case-sensitive BLEU score (Papineni et al., 2002) as implemented in the `multi-bleu-detok.perl` script of the Moses toolkit (Koehn et al., 2007). The BLEU score measures the *n-gram* overlap between the translation hypothesis and the reference translation and somewhat correlates with human judgements, however, it has a host of known limitations (Callison-Burch et al., 2006), such as requiring exact (word-level) matches, and poorly managing word order (and reorder).

4.2. *Score_mwe*

Evaluating MWE translation usually means measuring how well the MWEs featuring the source side are translated into the target. To do this, we ideally need to identify (the exact extent of) MWEs on the source side and align them both with their reference and their hypothesis translations. In practice, this can be difficult especially in the NMT framework, where we cannot faithfully rely on the attention mechanism as a valid substitute for phrase alignment (Ghader and Monz, 2017). To overcome these limitations, we devise a metric which does not need phrase alignment, only requiring a reference translation of the source MWE, as it works on full target sentences[1]. We refer to this metric as *Score_mwe*.

Our test sets have identified MWEs on the source side (the scope of the evaluation), and their reference translations are also identified in the target. We can start by taking each of the words that comprise the reference translation of the MWE, and take its closest match in the translated hypothesis according to a character-based distance metric to be its *actual* translation. We use the standard definition of the Levenshtein distance between two strings a, b (of length $|a|$ and $|b|$ respectively) as $\operatorname{lev}_{a,b}(|a|, |b|)$ where

$$\operatorname{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \operatorname{lev}_{a,b}(i-1,j) + 1 \\ \operatorname{lev}_{a,b}(i,j-1) + 1 \\ \operatorname{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \tag{2}$$

where $1_{(a_i \neq b_j)}$ is 0 when $a_i = b_j$ and 1 otherwise, and $\operatorname{lev}_{a,b}(i,j)$ is the distance between the first i characters of a and the first j characters of b. The first element in the min corresponds to deletion (from a to b), the second to insertion and the third to match or substitution.

Specifically, for each reference sentence in the test set, we take a MWE reference translation of length n, comprised of the words $x_0, \ldots, x_n$, with word lengths of $|x_0|, \ldots, |x_n|$. For each word in the sequence $x_0, \ldots, x_n$, we find the word in the translation hypothesis y_{min} with the lowest Levenshtein distance then average this over all the MWEs in the sentence to get the Score_mwe for that sentence, as described in Equation 3:

$$Score\_mwe = 1 - \frac{\sum_{i=0}^{i=n} \left(\frac{lev_{x_n, y_{min}}}{|x_n|}\right)}{n} \tag{3}$$

We calculate the *Score_mwe* for all the sentences in the corpus, and divide by the number of sentences, to get the score for the entire test set. In our implementation, the distance is capped to be at most equal to $|x_n|$ (length of the reference component), and – being normalised by it – ranges from a minimum of 0 to a maximum of 1.

From a linguistic viewpoint, we are making a strong assumption that the least distant word y_{min} corresponds to the *actual* translation of the reference component *without* looking at its real alignment. While of course this does not hold for all sentences, we empirically find that this assumption matches up with manual inspection because it is very unlikely that the items featuring the MWE translation appear elsewhere in the sentence. For instance, using this measure on the `test_ted` and `test_mwe` *reference* translations (described in Section 5.2.) yields an almost perfect score of 99.4% and 99.8%, respectively.

The measure considers character–level matches rather than full words. This allows the metric to account for partial matches, which are particularly important when translating into highly inflected languages like Italian. In fact, a system may be able to correctly detect and translate a MWE, but in a different grammatical form or category than it appears in the reference (which many times may be acceptable), and this cannot be accounted for by word-level matches. In Table 1, we report a toy example to explain such intuition: the MWE *he woke up* is translated by "si è svegliato" (lit. "himself is woken") in the reference translation and "si sveglia" (lit. "himself wakes") in the hypothesis, which is correct as for the choice of the lexical item but incorrect in verb tense. A word-based match would only match the first element. Similarly, the MWE *I rang up* (reference: "ho telefonato" (lit. "(I) have phoned")) is translated as "ho fatto una telefonata" (lit. "(I) have made a phone call"), which is a correct translation and would be over penalized by a word-based metrics. On the contrary, the *Score_mwe* is able to smooth these effects by awarding (higher) partial credit for the character-based matching strings. Another advantage of this approach is that, although our metric requires specifically annotated test corpora, it is able to focus on MWEs while traditional word–based metrics such as the BLEU score or even character–based methods such as chrF (Popović, 2015) are far less sensitive to improvements in MWE translation specifically.

(Source) he woke up **(Hyp.)** *si sveglia* **(Reference)** *si è svegliato*	**character-based** si è svegliato $(2/2 + 0/1 + 7/9)/3 = 0.59$ **word-based** si è svegliato $(1 + 0 + 0)/3 = 0.33$
(Source) I rang up **(Hyp.)** *ho fatto una telefonata* **(Reference)** *ho telefonato*	**character-based** ho telefonato $(2/2 + 9/10)/2 = 0.95$ **word-based** ho telefonato $(1 + 0)/2 = 0.5$

Table 1: Difference between character-based and word-based matching. Green indicates correct matches of the hypothesis over the reference string (blue); scores are normalized by the length of reference string.

4.3. Human evaluation

Finally, we ask 4 expert human annotators to evaluate the quality of the translation of 100 MWEs identified in our `test_100` set (described in Section 5.2.), which we compare with the results given by our proposed *Score_mwe*. Annotators are provided with the source sentences (with one identified MWE for each sentence), a reference translation for each sentence (with the identified translation of the MWE), and the systems' output for each sentence.

We ask annotators to focus on the translation of the identified MWE only and evaluate it for each sentence on a scale from 1 to 5, considering a) idiomaticity (whether the idiomatic meaning of the source MWE has been correctly identified) b) grammaticality, and c) fluency. Reference translation is given as benchmark, but we ask annotators to also consider alternative correct translations as valid. Like each of our selected automatic measures, we do not distinguish between these linguistic aspects in a fine-grained manner, and ask annotators to express one single score for each sentence for

[1] However, if alignment is available, it can work directly – with even more precision – on the aligned items instead of the whole target sentence.

each system. Finally, we average each system's sentence scores to get system-level evaluation.

5. Data description

5.1. Training and development sets

As a baseline for our translation experiments in a low resource setting (described in Sections 6.2.1.-6.3.2.) we use the English-Italian parallel corpus released for the *IWSLT Evaluation Campaign 2014*, composed of 181,874 translated sentences from the TED conferences transcripts (`train_small`) and available from the WIT^3 Web Inventory (Cettolo et al., 2014). As a baseline for our last experiment (Section 6.4.), we combine the above corpus with the English-Italian section of the Europarl parallel corpus (2012 v7 release) (Koehn, 2005) which consists of 1,909,115 parallel sentences from the proceedings of the European parliament (`train_big`). We use the available *TEDx.dev2014* English–Italian parallel corpus for development (`dev`) (1,056 sentences).

5.2. Test sets

We evaluate our models on three separate test sets. The first (`test_ted`) is the annotated corpus made available by Monti et al. (2015), consisting of 1,529 parallel English-Italian sentences from TED talks transcripts (but not included in our training sets). This corpus is manually annotated for MWEs and – to the best of our knowledge – it represents the most comprehensive MWE-aware bilingual resource for the English-Italian pair. Each manually annotated MWE on the (English) source side (which can be of any grammatical type) is aligned with its reference translation into the target language (Italian) (which we use in our *Score_mwe* evaluation metric, Section 4.2.), and the reference translation may, or may not, be in turn a MWE. In total, the English side the corpus presents 880 different MWE types (over a total of 2066 MWE instances), the majority of which are continuous (91%). MWE length spans from 2 to 7 words, the majority featuring either 2 or 3 words. The criteria according to which the `test_ted` corpus has been annotated are such that phrases like *there are*, *expect to* or *and so* are annotated as valid MWEs, but a stricter definition of MWE-hood would classify them as collocations or frequent compositional combinations of words. Thus, in order to specifically target idiosyncratic MWEs in our experiments, we construct a second test set (`test_mwe`), using the example sentences of a manually compiled bilingual dictionary, the *Il Ragazzini* dictionary of English–Italian (Ragazzini, 2019). The example sentences are complete sentences that are presented to the reader in order to exemplify a lemma, an expression or a specific use of a word. We collect 3,494 parallel sentences, connected to expressions that in the dictionary have been marked as MWEs by professional lexicographers. Some of these example sentences, along with the MWE and their translation, are reported in Table 2.

Finaly, we randomly select a subset of these sentences (100 sentences) and annotate them for MWEs (similarly to how was done in the `test_ted` corpus by Monti et al. (2015). Annotation is partially done automatically, by matching the lexical items in the sentence with the lemmatized MWE and its translation as reported in the dictionary, then manually fine-tuned to correct errors. We call this subset `test_100` and use it to evaluate our results with the *Score_mwe* metric, which requires identification of the MWE reference translations.

Lemma	**MWE**	**Example sentence** ***Translation***

`short`	`(come)`
	`short of`
		the result has come short of our expectations
		il risultato ha deluso le nostre speranze

`sweet`	`sweet spot`	they're struggling to find the sweet spot in the market *stanno faticando a trovare la collocazione giusta sul mercato*

Table 2: Example sentences from the `test_mwe` test set, which has a more rigorous definition of MWE-hood, with the indication of the lemma and the MWE featuring in them (as originally reported in the *Il Ragazzini* dictionary).

5.3. Data preprocessing

We preprocess all datasets by applying language-specific tokenization and truecasing using the Moses toolkit (Koehn et al., 2007). In order to limit the size of the training vocabulary and to achieve open-vocabulary translation, we use byte-pair encoding (Sennrich et al., 2016b) by jointly learning and applying BPE on both source and target sides with 32K merge operations and apply it with vocabulary threshold = 1 (meaning that when re–applying BPE each subword must have been seen at least once at training time, and is otherwise replaced by the UKN symbol).

6. Experiments

In this section we describe four methods to integrate knowledge from existing lexicographic resources into an NMT framework. We do this in a low-resource setting by augmenting our training instances with artificial parallel data, specifically targeting MWEs in the source (6.2.1.) or in the target language (6.2.2.), by identifying existing MWEs on the source side and symbolically encode them as single tokens (6.3.1.) or through a factored representation, along with more linguistic information (6.3.2.). Finally, we scale the last, most flexible method to a high-resource setting (6.4.).

6.1. Baseline training and settings

We train a translation model from English into Italian using an encoder–decoder recurrent neural architecture with attention and gated recurrent units. Since our aim is not that of achieving state of the art results, but rather improve performance in NMT regardless of the particular experimental setting, we use a shallow architecture (one bidirectional layer for the encoder and one single layer for the decoder) with the following parameters (which we do not optimize), following Sennrich and Haddow (2016): embedding layer size = 512; hidden layer size = 1024; training minibatches size = 80; beam size = 12; maximum sentence length = 50; dropout with probability = 0.2. We train for about 1 day using cross-entropy as our objective loss function, we sort sentences by length and we shuffle them at each epoch. We calculate BLEU score every 5,000 iterations and apply early stop when we see no improvement on the development set for at least 10 checks.

6.2. Data augmentation

6.2.1. MWE dictionary

In our first method (which we refer to as `MWE_dictionary`) we use linguistic knowledge about MWEs from a traditional lexicographic resource and augment our training data with additional lemmatized bi-text from a manually compiled source. This is the English–Italian section of the *Il Ragazzini* bilingual dictionary (Ragazzini, 2019), which we use in its latest XML-encoded version. In the dictionary, each MWE receives several translations (often synonyms or near synonyms, separated by a semicolon), which we preprocess by splitting them into separate training examples.

In order to get the MWE candidates, we extract all the lemmas which contain a whitespace and have therefore been classified as (idiosyncratic) lexical items with a "word" status – because they have their own dictionary entry – but are composed of two or more words. The majority of these expressions are verbal, especially verb-particle constructions (such as *stand at*, *take off* etc.), secondarily prepositional (*according to*), nominal (*Bach flower remedies*) and adjectival (*au pair*) expressions. Notice that we extract *lemmatized forms* and pair them with their (base form) translations, which is how they appear in every standard dictionary, such as, for example, the English base verbal form *get into* and its Italian translation in the present infinitive (*entrare*).

In total, we cover 3560 different MWE types and obtain 18,048 (parallel) MWE tokens after processing, which includes splitting the several translations connected to a signle MWE into separate instances. We add the so obtained parallel entries to the baseline training set for a total of 199,922 sentences (i.e. we augment the baseline training data with synthetic data by approx. 10%). We learn and apply BPE jointly in both languages with 32K merge operations, and train the new model with the hyperparameters described in Section 6.1.

6.2.2. MWE in context with backtranslation

In this method (`MWE_backtrans`), we aim to improve translation by attending to MWEs *in context* on the target side. We do so by augmenting the training data with monolingual sentences on the target side that contain at least one MWE in context for each target instance. The hypothesis behind this is that, since RNNs are capable of modelling sequence context, they should be better at discriminating between MWEs and simple phrases — and therefore ensure a better translation quality — if exposed to their larger contexts. This approach may also result in overgeneration, as it can pair an idiomatic MWE with an incorrect literal translation on the English side; nonetheless, it should provide the decoder with both correct MWE instances and their contexts, which should help the system generate more correct, naturally sounding language.

Sentences containing MWEs can be extracted from raw text using standard identification techniques (Rikters and Skadiņa, 2016), but in order to ensure that our data contain genuine MWE candidates, we extract them from a monolingual dictionary of Italian *lo Zingarelli* (Cannella and Lazzarini (a cura di), 2019), encoded in XML with a similar mark-up schema as the *Il Ragazzini* used in `MWE_dictionary`. The encoding of the dictionary explicitly marks phrasal components with the `<phr>` tag, and for a selection of them (in particular, for the most idiosyncratic ones) it provides some *usage examples* (in the form of complete sentences). We extract these usage example sentences – which therefore contain at least one MWE – for a total of 3923 Italian sentences, and pair each with its approximated back-translation in English, obtained by re-training a model with the same hyperparameters, setting and data used as the baseline (6.1.), but in reversed direction (from Italian to English). We include the so obtained supplementary data to the original training instances, for a total of 185,797 (i.e. we augment the training data by 2%). We jointly learn and apply BPE with 32K merge operations, and train the new model with the same hyperparameters as in 6.1.

6.3. MWE labelling

In the following methods, we use the identification tool provided by the `annotate_mwe.py` module of the *mwetoolkit* (Ramisch et al., 2010) to identify MWEs on the source side. The method requires a list of MWE candidates to be specified in their lemmatized form, and uses it to look at each word's lemma in the text to tag it as part of a MWE[2].

As a source of lemmatized MWEs candidates, we exploit the English section of the *Il Ragazzini* dictionary of English-Italian, which we already employed for `MWE_dictionary` (Section 6.2.1.). There we adopted a strict definition of MWE-hood and only considered the `<lemma>` elements that contained a whitespace; however, in that method the number of MWE candidate type was little more that 3k, while here we aim to expand our candidate list in order to identify as many MWEs as possible. Thus, we take a more shallow approach and extend the English MWE list already used in `MWE_dictionary` by extracting *all* the elements in the dictionary annotated as phrasal units. These candidates span from a minimum length of 2 words to a maximum of 13, and can pertain to any grammatical category.

After constructing the MWE candidate list, we use the Natural Language Toolkit (NLTK) (Loper and Bird, 2002) to perform POS tagging over all our source training instances. POS annotation is then used by the `stem.lemmatize()` method in the NLTK module, combined with positional features, to extract the lemma for each word. Finally, we feed the lemma and POS information on each lexical item

[2] We are aware that more sophisticated methods for MWE identification exist but devising more complex methods of MWE identification would represent a task on its own, falling outside the scope of the present study.

to the `annotate_mwe.py` script to identify and annotate MWEs on the source text.

The method returns a factored representation of the text in Moses style (where each word is represented as a chain of features separated by the pipe symbol |), and words making up the identified MWEs (as well at its lemma) are joint with an underscore. For example, a sentence such as

(1) **the men were angry and walked off**

would receive the following annotation:

(2) **the**|the|DT **men**|man|NN **were**|be|VB
angry|angry|JJ **and**|and|CONG
walked_off|walk_off|VB

where the first factor indicates the surface form, the second the lemma (which can be a MWE lemma as in *walk_off*), and the third the part-of-speech. For the sake of our translation experiments, we use such information and encode it in two different ways, which we refer to as "word-with-spaces" and a "IOB encoding", respectively.

6.3.1. Word–with–spaces approach

In a first setting, which we use for `MWE_wordwithspaces`, we take a "word with spaces" approach and simply encode each MWE as a single word, by keeping the original underscore given by the annotation tool whenever a MWE has been identified, and discard all remaining linguistic information. In this setting, sentence (1) would thus be represented as

(3) the men were angry and **walked_off**

The intuition behind this is that, like subword units, named entities, semantic roles etc., MWEs represent a single unit "spread out" onto several symbols, and therefore encoding them as a single token may be beneficial to their interpretation.

Like in the other methods, we use BPE to achieve open vocabulary translation. When learning BPE, we do not consider the underscore as a word separator and instead we let the BPE algorithm treat it as a valid character. This has the nice side effect of allowing the algorithm to automatically learn regularities over MWEs too across our dataset. For example, if we examine the vocabulary file produced by the BPE learning script, we notice that, for instance, `out</w>`, `out@@` and `out_@@` are encoded as different tokens[3], meaning that, when applying BPE at test time, the system will know that they represent three different lexical items (and only the latter is part of a MWE), providing a simple but effective disambiguation method for these otherwise indistinguishable components. Moreover, the fact of encoding `out_` as a separate token can potentially allow the system to generalize on different MWEs not seen at training time, considering them MWEs if correctly annotated at test time. However, as a drawback, this simple method does not allow to encode discontinuous MWEs.

We apply the annotation on the `train_small` dataset, learn and apply BPE jointly in both languages with 32K merge operations and train the new model with the same hyperparameters as the baseline (6.1.).

6.3.2. IOB encoding

In a second setting (`MWE_iob_small`), we construct input features and annotate them in Moses factored format, as required by our chosen toolkit Nematus. We consider three features: the surface form of a word (factor 1), the lemma (factor 2) and whether the string in factor 1 is at the beginning, inside or outside a MWE. Firstly, we discard the POS information from the initial annotation. We keep factor 2 as returned by the annotation method, which then reports the word-level or the MWE-level lemma (with underscores). Secondly, we split the surface MWE forms into their single-word components, and apply BPE on the single-word corpus. For each resulting word or subword unit, we indicate whether it is at the beginning (B), inside (I) or outside (O) a MWE in field 3, and copy the word's feature value in factor 2 to all its smaller units. Assuming *walked* were a rare word, sentence (1) would receive the following annotation[4]:

(4) **the**|the|O **men**|man|O **were**|be|O
angry|angry|O **and**|and|O
walk@@|walk_off|B **ed**|walk_off|B
off|walk_off|I

Unlike the word-with-spaces approach in `MWE_wordwithspaces`, which only applies to continuous MWEs, this strategy may also apply to discontinuous configurations (which, however, we do not consider in the present study because of the lack of an appropriate identification technique, but represents a natural extension of our work, see Section 8.).

Following recommendations in Sennrich and Haddow (2016), we want to make sure that the improvements over the baseline are not due to an increase in the number of model parameters. Therefore, we make sure that the sum of the sizes of the embedding layers for each factor equals the size of the embedding layer as in the baseline (512). We set embedding size to 300, 202 and 10 for factors 1, 2 and 3, respectively (roughly balancing the sizes of the respective vocabulary, but without optimizing for this parameter). We adjust the training script as to specify the number of factors (3) and one vocabulary file for each of them (plus the target vocabulary). All other parameters are as in Section 6.1.

6.4. Factored NMT at scale

Finally, we aim to improve the translation of MWEs, and the general translation quality, in a high-resource setting by scaling the method used in `MWE_iob_small` (6.3.2.) to a much bigger training set. The reasons why we choose to scale this particular method and not the best scoring method described above (which would have been `MWE_backtrans`) are multiple. Firstly, the method used in `MWE_backtrans` is necessarily dependent on the training size, because data augmentation is likely to have an insignificant effect if it is only by a very small percentage; on

[3] where </w> and @@ indicate word and subword boundaries, respectively

[4] Notice that under this setting factor 3 is constrained in that B may not be followed by O, I may be followed by either I O or B (indicating consecutive MWEs); notice that more subsequent B's are possible as they may indicate multiple subwords units of the first word of a MWE.

the contrary, we aim to propose a method that is easily scalable to any size of the dataset, considering that NMT has proven to reach great improvement with increased training size. Secondly, the method used in `MWE_iob_small` is the most flexible of all other methods because it can potentially account for discontinuous MWEs. Thirdly, while we use a naive MWE identification technique, this method has a potential for improvement if more fine-grained identification methods are applied to the preprocessing of the data.

As a baseline for this method, which we refer to as *Baseline (big)*, we learn BPE and train a model with the same hyperparameters as in 6.1. on the much bigger, 2M sentence dataset `train_big`, described in detail in Section 5.1. We then apply the processing steps explained in Section 6.3.2. (factored MWE representation with IOB notation) to this dataset, learn and apply BPE and train the model again with the same parameters. We call the system `MWE_iob_big`.

7. Results and discussion

In Table 3, we report the results obtained by our systems according to detokenised case-sensitive BLEU score, *Score_mwe* (S_mwe) and human evaluation (Hum) on the different test sets. As explained in Section 4., Score_mwe and human evaluation on `test_mwe` are calculated on the annotated subset identified as `test_100`.

Dataset	`test_ted`		`test_mwe`		
	BLEU	**S_mwe**	**BLEU**	**S_mwe** (`t_100`)	**Hum** (`t_100`)
B.line small	21.34	6.42	3.53	1.3	0.1
Dictionary	22.37	6.48	6.64	2.83	0.62
Backtrans.	**22.61**	**6.5**	**8.62**	**3.88**	1.20
Wordwithsp.	22.25	6.44	**8.62**	3.8	**1.26**
IOB_small	22.58	6.43	5.46	3.31	0.76
B.line big	26.36	6.38	13.01	4.39	2.08
IOB_big	**26.78**	**6.88**	**14.44**	**5.89**	**3.24**

Table 3: Experiment results. The best results on each test set for each setting (high and low resource) are boldfaced.

All proposed methods outperform their baseline, with `MWE_backtrans` scoring best on every test set according to all metrics in a low resource setting. The improvements on BLEU score generally indicates that, under a low resource setting (small), our methods can make the overall quality of automatic translation better (by approx. 1 point in general language, and up to 5 points on the MWE targeted test set). In fact, while the improvement on the general language `test_ted` is not marked, we observe a substantial improvement in both BLEU and Score_mwe in translating the `test_mwe`. The extremely low figures for the baseline model on this dataset (which are hardly better than random translation) indicate that it is particularly hard for the NMT system to translate idiosyncratic, genuinely non-compositional MWEs like the ones included in `test_mwe`, especially if the system was not trained with in–domain text.

As for the higher resource setting (big), as expected, the baseline (big) model outperforms all models trained on the small dataset. `MWE_iob_big`, in turn, improves the baseline by 0.42 and 1.13 BLEU points on the `test_ted` and the `test_mwe` datasets, respectively, indicating that the proposed method is beneficial both on general and MWE translation. This is confirmed by the improvements in the Score_mwe measure by 0.05 and 0.15 respectively.

The human annotation confirms such improvements over the small and big baselines, which are even more marked than in autometic measures. The (Pearson) correlation between the average system scores of the individual human annotators was high (between 0.95 and 0.99) as is the correlation between the average of all human scores and Score_mwe on `test_100` (0.95), as Figure 1 demonstrates.

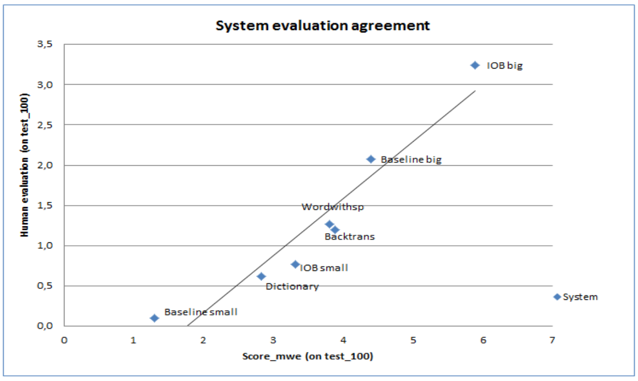

Figure 1: Human evaluation (y) and Score_mwe (x) results on the different translation systems.

Human evaluation and the Score_mwe generally agree in ranking the systems except for the best performing one in the low resource setting, where `MWE_wordwithspaces` scored best according to the annotators while automatic metrics favoured `MWE_backtranslation`. However, the difference between the two systems on all test sets are really not marked. We believe that the general agreement between the *Score_mwe* metric and the human evaluation in measuring the improvement of the systems over the baseline (and between each other) indicates that Score_mwe is good at capturing how well a system is translating MWEs specifically.

In order to further verify that the improvements achieved involve genuinely idiosyncratic expressions (and does not pertain to general translation improvement only), we carry out manual inspection of several instances of the `MWE_iob_big` output. In Table 4, we report some translation examples which we compare both with the baseline and with `MWE_backtrans`.

In some instances (Example 1), where `MWE_iob_big` produces a correct translation, `MWE_backtrans` may capture the overall meaning of the MWE but produce partially incorrect, literal translations ("costo di" instead of "costo della", missing the idiosyncratic combined article "la"), and also produces a wrong agreement between the subject and the past participle "cresciuta" (with feminine *-a* instead of masculine *-o*). On the other hand, the baseline mistranslates *scaled up* with "diminuito" (lit. *lowered*), producing almost the opposite meaning. As exemplified in Example 2, `MWE_iob_big` is generally good at identifying MWEs on the source side and produces natural sounding expressions

Example 1	**Text**
Source (test_mwe)	the cost of living has scaled up again
Backtrans	E il costo di vita è cresciuta qui
Baseline (big)	il costo della vita è diminuito di nuovo la mia
IOB_big	il costo della vita è diventato sempre più alto
Example 2	**Text**
Source (test_mwe)	I have to account for every penny I spend
Backtrans	devo discutere con tutti i miei piedi
Baseline (big)	devo tenere conto di ogni centesimo ...
IOB_big	devo rendere conto di ogni centesimo ...

Table 4: Manual inspection of example translations for MWE_iob_big, compared to Baseline (big) and MWE_backtrans. In blue, the original source MWE. Underlined is MWE translation. in green correct translations, in red incorrect translations.

on the target side, such as "rendere conto di", which is in turn a MWE in Italian.

8. Conclusions and further work

In this work, we argued that MWEs are a relevant phenomenon in natural languages and that neural machine translation systems do not generally handle them well. We showed that explicitly addressing MWEs can improve translation quality, and presented several methods to integrate such information and evaluate results, according to different resource availability (both in a low-resource and in a high-resource setting). The datasets we exploited are not freely available, neither for Italian nor other languages, and we take it as future work to extend this with automatically detected MWEs through open–source toolkits. We evaluated our findings with standard measures and through human judgments. We devised a specific score to assess the quality of MWE translation (Score_mwe) which, unlike previous metrics, does not require phrase alignment, and which correlates with human judgments. Along with the code used in all the stages of the work, we also make available our training sets annotated for MWEs, as well as corpus-based lists of MWEs, for reproduction and extension studies[5].

In terms of our findings, we have shown that, in a low-resource setting, augmenting the training data with MWE-targeted monolingual text, as in the MWE_backtrans method, by as little as 2% proved beneficial both in terms of general translation quality and MWE translation quality specifically. This is congruent with the the fact that NMT architectures perform better when exposed to a larger linguistic context – which they can model well – and MWEs make no exception in this respect. In our experiments we relied on a monolingual dictionary as a source of text: a natural extension of this method will be using identification techniques on the target language to extract more monolingual data from raw text, and augment the training data with that.

If monolingual data are not available, using synthetic lemmatized MWE source-target pairs, as in the MWE_dictionary method, proved less effective than identifying MWEs only on the source side, and in some cases hurt translation by producing ungrammatical forms. Thus, if only lemmatized lists are available, it is advisable to only exploit one side of the (lemmatized) bitext to train a simple identification method and annotate MWEs on the source. When doing so in a low-resource setting, annotating MWEs as single tokens, as in the MWE_wordwithspaces approach, ensures better results than the IOB tagging methods like the one leveraged in MWE_iob_small.

However, a word-with-spaces approach is limited to continuous MWEs, which do not cover the whole spectrum of the phenomenon, and is less likely to scale on a high-resource setting. In the MWE_iob_big method, we devised a flexible method for annotating MWEs on the source side, which consistently improved general and MWE translation quality and markedly outperformed its baseline according to all metrics. A natural extension of this method will be using a more sophisticated MWE identification method to also include discontinuous MWEs or MWEs appearing in marked configurations (such as topicalized or passive forms).

In conclusion, we believe that efforts should be made in the direction of improving the existing MWE identification systems. Recent approaches have experimented on developing MWE identification methods using neural networks (Gharbieh et al., 2017) (Klyueva et al., 2017), which however are supervised and thus heavily rely on the availability of annotated resources. In the future, it may be interesting to verify whether it is possible to automatically induce MWE identification with neural networks in an unsupervised fashion, for example modelling the internal compositionality of an expression by embedding it as a whole, and compare its vector representation with that of its single components. If such techniques prove successful, it may be possible to integrate them into a neural machine translation architecture, for instance by making MWE identification part of the training objective.

9. Acknowledgements

We would like to thank Zanichelli editore for providing the lexical resources used in this project. Alexandra Birch would like to acknowledge funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 825299 (GoURMET) and the EPSRC innovation fellowship MTStretch. We would like to thank the anonymous reviewers for their insightful comments, and the annotators for completing the human evaluation.

10. Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015.

Baldwin, T. and Kim, S. N. (2010). Multiword expressions. *Handbook of natural language processing*, 2:267–292.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the role of bleu in machine translation re-

[5] Available at github.com/azaninello/MWE_NMT

search. In 11th Conference of the European Chapter of the Association for Computational Linguistics.

Calzolari, N., Fillmore, C. J., Grishman, R., Ide, N., Lenci, R., Macleod, C., and Zampolli, A. (2002). Towards best practice for multiword expressions in computational lexicons. In In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Canary Islands, page 40.

Cannella, M. and Lazzarini (a cura di), B. (2019). Lo Zingarelli, vocabolario della lingua italiana. Zanichelli editore.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., and Federico, M. (2014). Report on the 11th iwslt evaluation campaign, iwslt 2014. In Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam, page 57.

Chen, B. and Cherry, C. (2014). A systematic comparison of smoothing techniques for sentence-level bleu. In Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 362–367.

Constant, M., Eryiğit, G., Monti, J., Van Der Plas, L., Ramisch, C., Rosner, M., and Todirascu, A. (2017). Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In Proceedings of the ninth workshop on statistical machine translation, pages 376–380.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Everaert, M., Van der Linden, E.-J., Schreuder, R., Schreuder, R., et al. (2014). Idioms: structural and psychological perspectives. Psychology Press.

Ghader, H. and Monz, C. (2017). What does attention in neural machine translation pay attention to? In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 30–39, Taipei, Taiwan, November. Asian Federation of Natural Language Processing.

Gharbieh, W., Bhavsar, V., and Cook, P. (2017). Deep learning models for multiword expression identification. In Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017), pages 54–64.

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H.-C., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Jackendoff, R. (1995). The boundaries of the lexicon. *Idioms: Structural and psychological perspectives*, pages 133–165.

Junczys-Dowmunt, M., Dwojak, T., and Hoang, H. (2016). Is neural machine translation ready for deployment. *A case study on*, 30.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klyueva, N., Doucet, A., and Straka, M. (2017). Neural networks for multi-word expression detection. In Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017), pages 60–65.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, pages 177–180.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In MT summit, volume 5, pages 79–86.

Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, pages 63–70.

Monti, J., Sangati, F., and Arcan, M. (2015). Ted-mwe: a bilingual parallel corpus with mwe annotation. In Proceedings of the Second Italian Conference on Computational Linguistics CLiC-it, pages 193–197.

Nguyen, T. and Chiang, D. (2018). Improving lexical choice in neural machine translation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 334–343, New Orleans, Louisiana, June. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistics.

Popović, M. (2015). chrf: character n-gram f-score for automatic mt evaluation. In Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 392–395.

Ragazzini, G. (2019). Il Ragazzini, dizionario Inglese-Italiano/Italiano-Inglese. Zanichelli editore.

Ramisch, C., Villavicencio, A., and Boitet, C. (2010). Multiword expressions in the wild?: the mwetoolkit comes in handy. In Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations, pages 57–60. Association for Computational Linguistics.

Ramisch, C., Besacier, L., and Kobzar, A. (2013). How hard is it to automatically translate phrasal verbs from english to french. In MT Summit 2013 Workshop on Multi-word Units in Machine Translation and Translation Technology, pages 53–61.

Rikters, M. and Bojar, O. (2017). Paying attention to multiword expressions in neural machine translation. *arXiv preprint arXiv:1710.06313*.

Rikters, M. and Skadiņa, I. (2016). Combining machine translated sentence chunks from multiple mt systems. In International Conference on Intelligent Text Processing and Computational Linguistics, pages 27–37. Springer.

Sennrich, R. and Haddow, B. (2016). Linguistic input features improve neural machine translation. In Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers, pages 83–91.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 86–96.

Sennrich, R., Haddow, B., and Birch, A. (2016b). 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.

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Miceli-Barone, A. V., Mokry, J., et al. (2017). Nematus: a toolkit for neural machine translation. In Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pages 65–68.

Tang, Y., Meng, F., Lu, Z., Li, H., and Yu, P. L. (2016). Neural machine translation with external phrase memory. *arXiv preprint arXiv:1606.01792*.

Villavicencio, A., Bond, F., Korhonen, A., and McCarthy, D. (2005). Introduction to the special issue on multiword expressions: Having a crack at a hard nut. *Computer Speech & Language*, 19(4):365–377.